

## **BLM0463 Veri Madenciliğine Giriş**

- **Kaynaklar**

***Introduction to Data Mining, Pang-Ning Tan, Michigan State University, Michael Steinbach, University of Minnesota, Vipin Kumar, University of Minnesota, Publisher: Addison-Wesley (Textbook & its slides)***

<https://www-users.cs.umn.edu/~kumar001/dmbook/firsted.php>

- **Başarı Notu**

- **Vize (%35)**
- **Final (%50)**
- **Lab. & Ödev & Proje (%15)**

- **Ders Asistanı**

- **Arş. Gör. Sena DİKİCİ**

# Data Mining: Introduction

---

---

## Lecture Notes for Chapter 1

Introduction to Data Mining, 2<sup>nd</sup> Edition

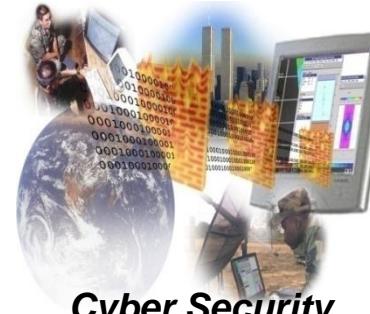
by

Tan, Steinbach, Karpatne, Kumar

Orijinal slaytların Türkçe çevirisidir.

# Large-scale Data is Everywhere!

- Veri oluşturma ve toplama teknolojilerindeki ilerlemeler nedeniyle hem ticari hem de bilimsel veri tabanlarında muazzam bir veri büyümesi olmuştur.
- New mantra (kutsal söz)
  - Mümkün olduğunda (**whenever**) ve mümkün olan her yerde (**wherever**) her türlü (**whatever**) veriyi toplayın.
- Beklentiler
  - Toplanan veriler ya toplanan amaç için ya da öngörülmeyen bir amaç için değerli olacaktır.



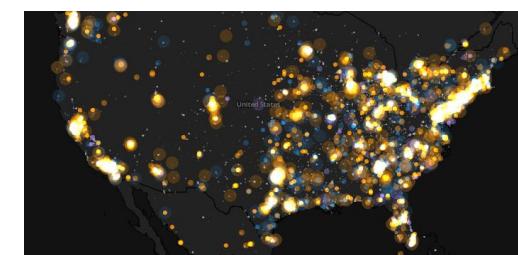
**Cyber Security**



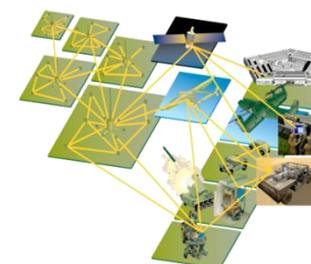
**E-Commerce**



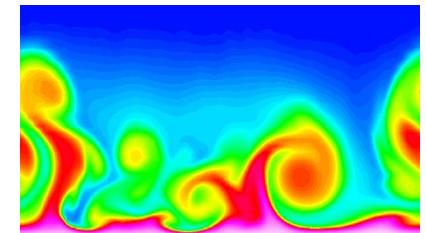
**Traffic Patterns**



**Social Networking: Twitter**



**Sensor Networks**



**Computational Simulations**

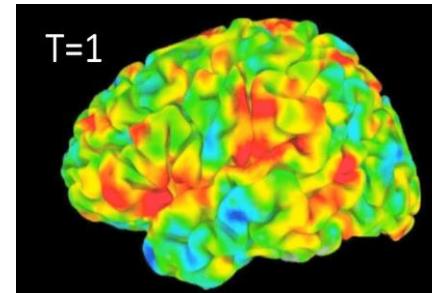
# Why Data Mining? Commercial Viewpoint

- Çok sayıda veri toplanıyor ve depolanıyor
  - Web data
    - ◆ Yahoo has Peta Bytes of web data
    - ◆ Facebook has billions of active users
  - mağaza / marketlerde alışveriş, e-ticaret
    - ◆ Amazon.com'u her gün milyonlarca kullanıcı ziyaret ediyor
  - Bank/Credit Card transactions
- Bilgisayarlar daha ucuz ve daha güçlü hale geldi
- Rekabetçi baskı güçlü hale geldi
  - Avantaj yakalamak için daha iyi, özelleştirilmiş hizmetler sunmak (örneğin, Müşteri İlişkileri Yönetimi'nde- **Customer Relationship Management**)

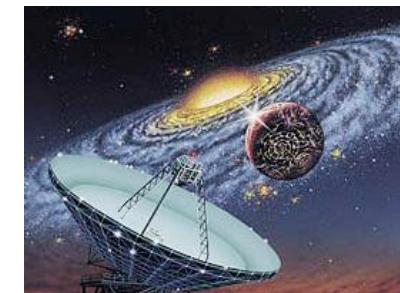


# Why Data Mining? Scientific Viewpoint

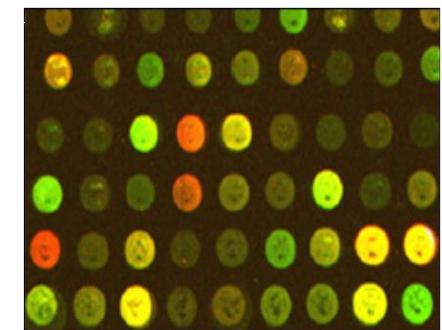
- Çok yüksek hızlarda toplanan ve depolanan veriler
  - Uydudaki sensörler (remote sensors on a satellite)
    - ◆ NASA EOSDIS yılda petabyte'ların üzerinde dünyaya ilişkin bilimsel veri arşivler
  - gökyüzünü tarayan teleskoplar
    - ◆ Sky survey data
  - Yüksek-hacimli biyolojik veriler (High-throughput biological data)
  - Bilimsel simülasyonlar
    - ◆ birkaç saat içinde üretilen terabaytlarca veri
- Veri madenciliği bilim insanlarına yardımcı olur
  - büyük veri kümelerinin otomatik analizinde
  - Hipotez oluşturmada



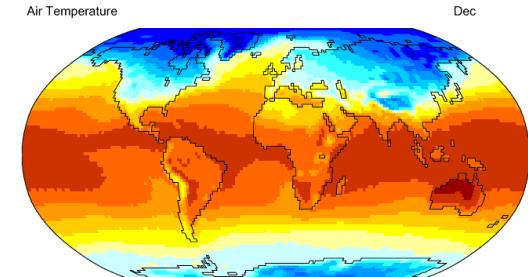
fMRI Data from Brain



Sky Survey Data



Gene Expression Data

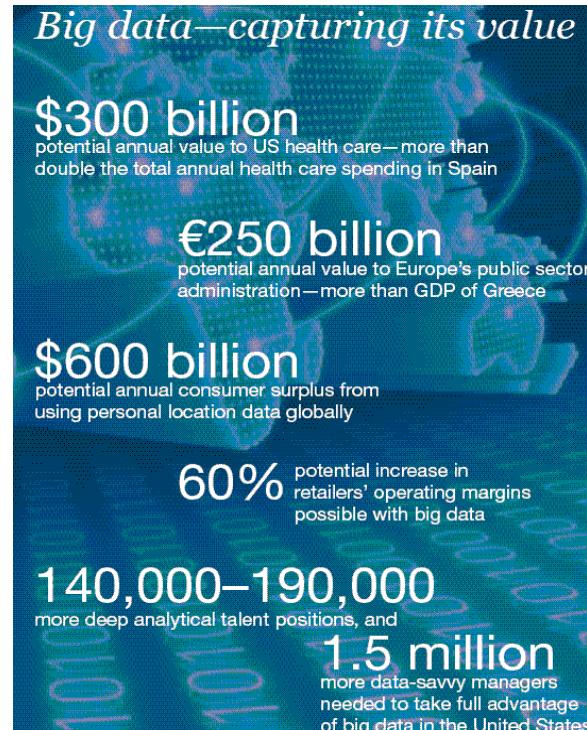
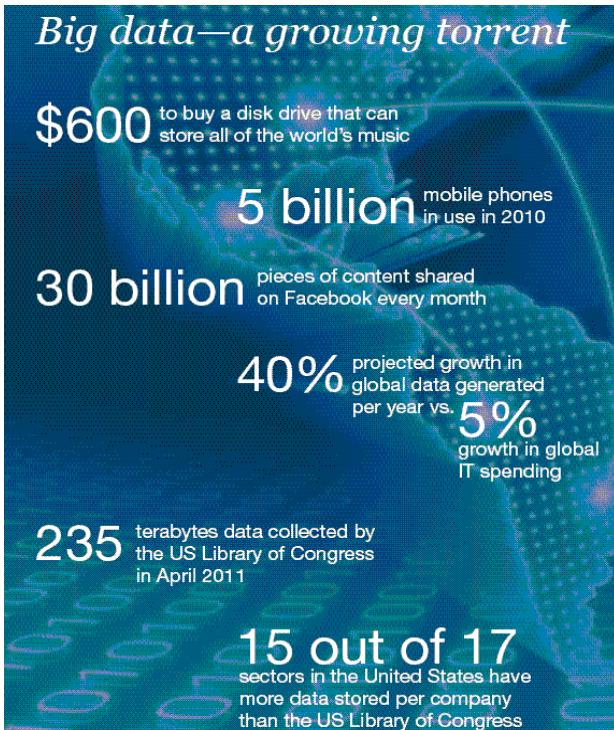


Surface Temperature of Earth

# Hayatın her alanında verimliliği artırmak için harika fırsatlar

McKinsey Global Institute

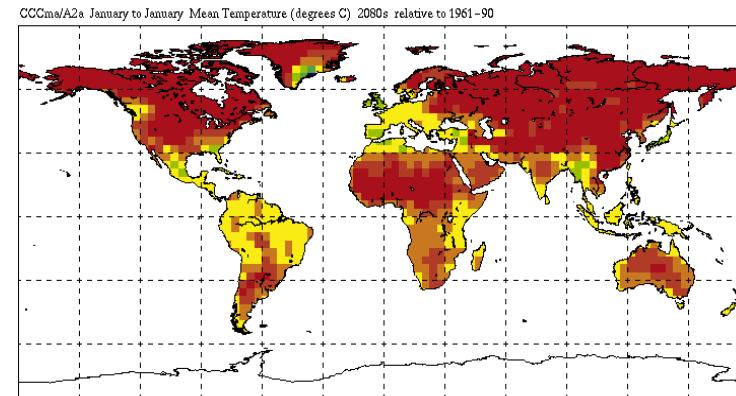
## Big data: The next frontier for innovation, competition, and productivity



# Toplumun Önemli Sorunlarını Çözmek İçin Büyük Fırsatlar



Sağlık hizmetlerini iyileştirmek ve maliyetleri düşürmek



İklim değişikliğinin etkilerini tahmin etmek



Alternatif / yeşil enerji kaynakları bulmak

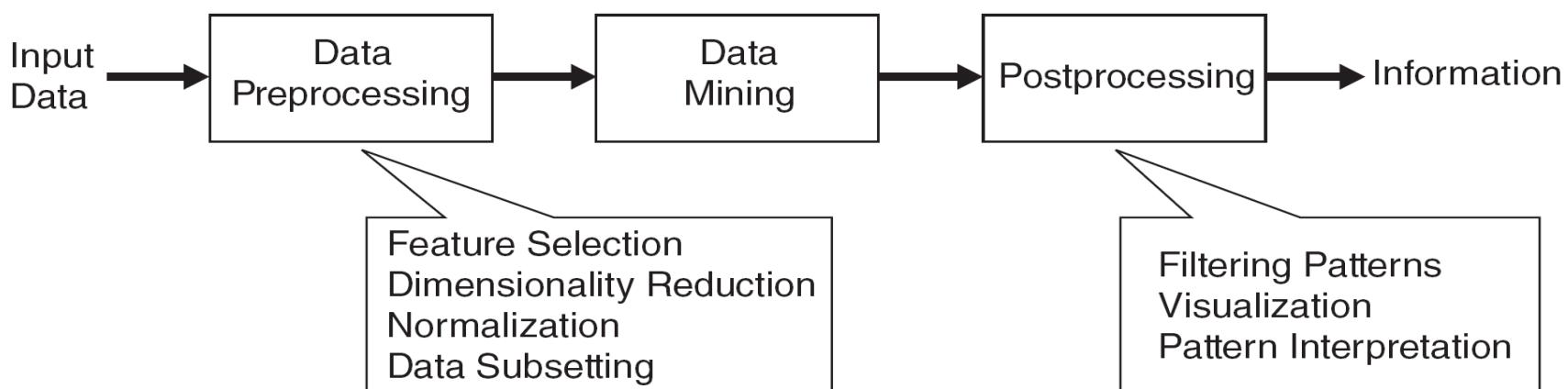


Tarımsal üretimi artırarak açlığı ve yoksulluğu azaltmak

# What is Data Mining?

- Pek çok tanımı vardır

- Verilerden örtük (**implicit**), önceden bilinmeyen ve potansiyel olarak yararlı (önem arz eden) bilgilerin çıkarılması
- Anlamlı örüntüleri keşfetmek için büyük miktarlarda verinin otomatik veya yarı otomatik olarak keşfi (**exploration**) ve analizi



# What is (not) Data Mining?

- What is not Data Mining?

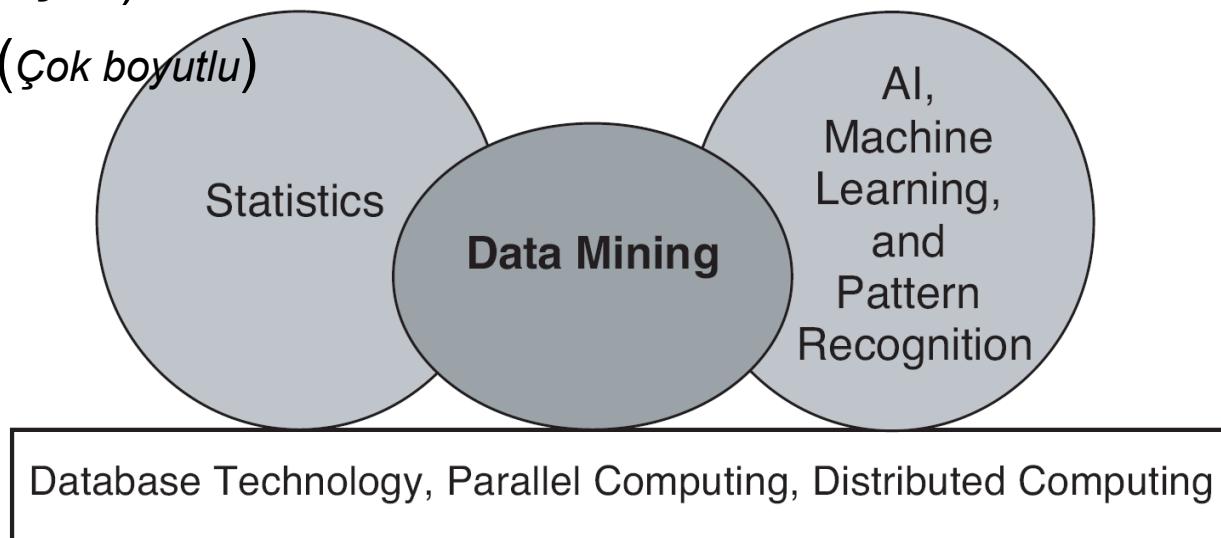
- Telefon rehberinde telefon numarasını aramak
- “Amazon” hakkında bilgi için bir Web arama motorunu sorgulamak

- What is Data Mining?

- Belirli isimler ABD'nin belirli bölgelerinde daha yaygındır (O'Brien, O'Rourke, O'Reilly... Boston bölgesinde)
- Arama motoru tarafından döndürülen benzer belgeleri içeriklerine göre gruplandırılın (örn. Amazon yağmur ormanları, Amazon.com)

# Origins of Data Mining

- Makine öğrenmesi / yapay zeka, örüntü tanıma, istatistik ve veritabanı sistemlerinden faydalıdır
- Geleneksel teknikler uygun olmayabilir, çünkü veri
  - Large-scale (*Büyük ölçekli*)
  - High dimensional (*Çok boyutlu*)
  - Heterogeneous
  - Complex
  - Distributed



- Yeni ortaya çıkan veri bilimi (**data science**) ve veri güdümlü keşif (**data-driven discovery**) alanının önemli bir bileşeni

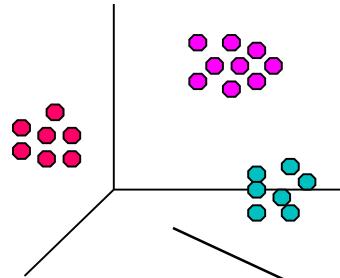
# Data Mining Tasks

---

- Tahmin/Öngörü Yöntemleri (**Prediction Methods**)
  - Diğer değişkenlerin bilinmeyen veya gelecekteki değerlerini tahmin etmek için bazı değişkenler kullanır.
- Tanımlama/Açıklama Yöntemleri (**Description Methods**)
  - Verileri tanımlayan, insan tarafından yorumlanabilen örüntüleri bulur.

From [Fayyad, et.al.] Advances in Knowledge Discovery and Data Mining, 1996

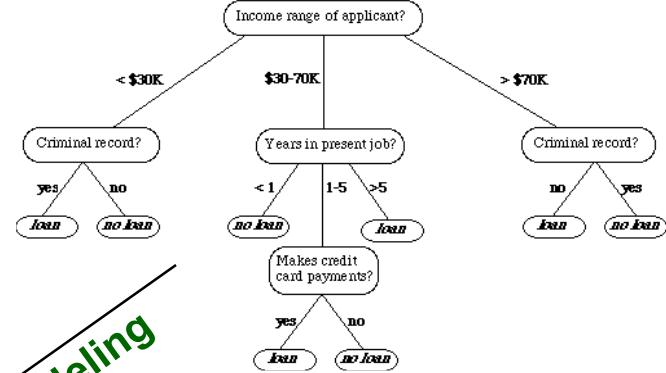
# Data Mining Tasks ...



*Clustering*

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes
11	No	Married	60K	No
12	Yes	Divorced	220K	No
13	No	Single	85K	Yes
14	No	Married	75K	No
15	No	Single	90K	Yes

*Association Rules*



*Predictive Modeling*

*Anomaly Detection*

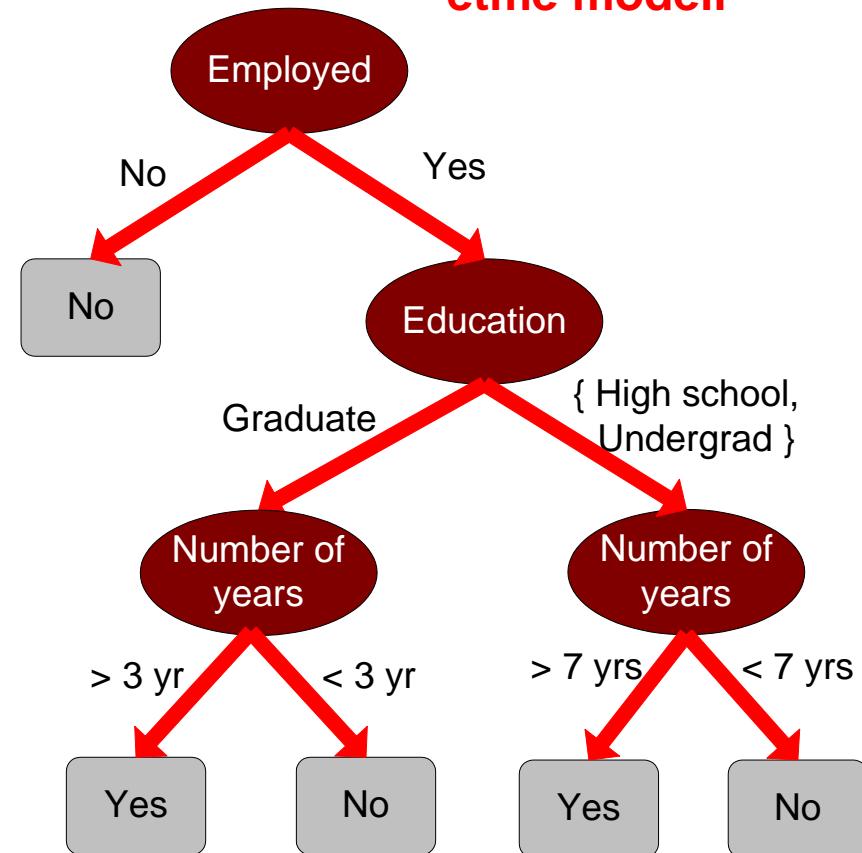


# Predictive Modeling: Classification

- Sınıf özniteliği (class attribute) için diğer özniteliklerin değerlerinin bir fonksiyonu olarak bir model bulma

Kredi liyakatını tahmin etme modeli

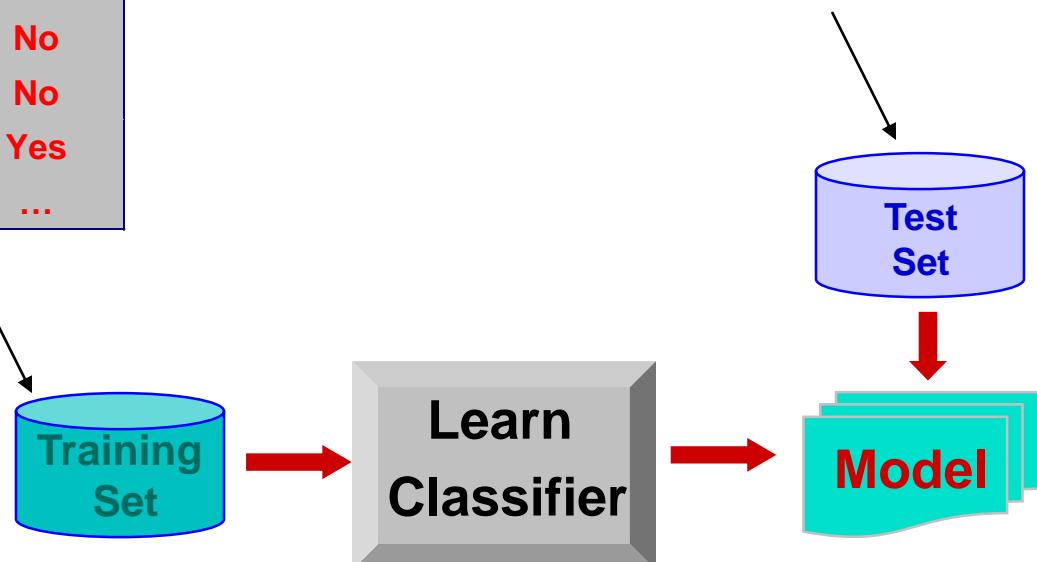
Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Graduate	5	Yes
2	Yes	High School	2	No
3	No	Undergrad	1	No
4	Yes	High School	10	Yes
...	...	...	...	...



# Classification Example

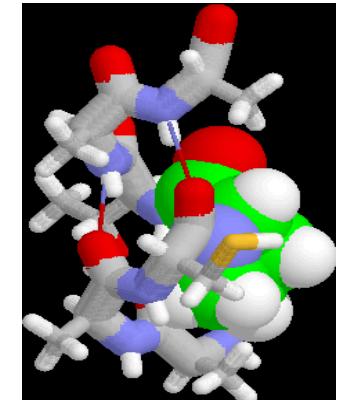
categorical categorical quantitative class				
Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Graduate	5	Yes
2	Yes	High School	2	No
3	No	Undergrad	1	No
4	Yes	High School	10	Yes
...	...	...	...	...

Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Undergrad	7	?
2	No	Graduate	3	?
3	Yes	High School	2	?
...	...	...	...	...



# Examples of Classification Task

- Kredi kartı işlemlerini yasal veya hileli olarak sınıflandırma
- Uydu verilerini kullanarak Arazi örtülerini (su havzaları, kentsel alanlar, ormanlar, vb.) sınıflandırma
- Haber sayfalarını finans, hava durumu, eğlence, spor vb. olarak kategorize etme
- Siber dünyada izinsiz giriş yapmaya çalışanları belirleme
- Tümör hücrelerini iyi huylu veya kötü huylu olarak tahmin etme
- Proteinin sekonder yapılarını alfa-sarmal, beta-yaprak veya rastgele spiral olarak sınıflandırmak



# Classification: Application 1

---

- Sahtekarlık Tespiti (Fraud Detection)
  - **Amaç:** Kredi kartı işlemlerindeki hileli vakaları tahmin etmek.
  - **Yaklaşım:**
    - ◆ Kredi kartı işlemlerini ve hesap sahibinin bilgileri öznitelik olarak kullanmak
      - müşteri ne zaman satın alır, ne satın alır, ne sıkılıkta zamanında ödeme yapar, vb.
    - ◆ Geçmiş işlemler sahtekarlık veya yasal işlem olarak etiketlenir. Bu, sınıf niteliğini oluşturur.
    - ◆ İşlemlerin sınıfı için bir model eğitilir/öğrenilir.
    - ◆ Bir hesaptaki kredi kartı işlemlerini gözlemleyerek sahtekarlığı tespit etmek için bu model kullanılır.

# Classification: Application 2

- Telefon operatörlerinin müşterileri için kayıp tahmini (Churn prediction)
  - **Amaç:** Bir müşterinin bir rakibe kaptırılıp kaptırılmayacağını tahmin etmek.
  - **Yaklaşım:**
    - ◆ Nitelikleri bulmak için geçmiş ve mevcut müşterilerin her biriyle ilgili işlemlerin ayrıntılı kaydını kullanılır.
      - Müşterinin ne sıklıkta aradığı, nereden aradığı, günün hangi saatinde en çok aradığı, finansal durumu, medeni durumu vb.
    - ◆ Müşteriler sadık veya sadık olmayan olarak etiketlenir.
    - ◆ Sadakat için bir model oluşturulur

From [Berry & Linoff] Data Mining Techniques, 1997

# Classification: Application 3

---

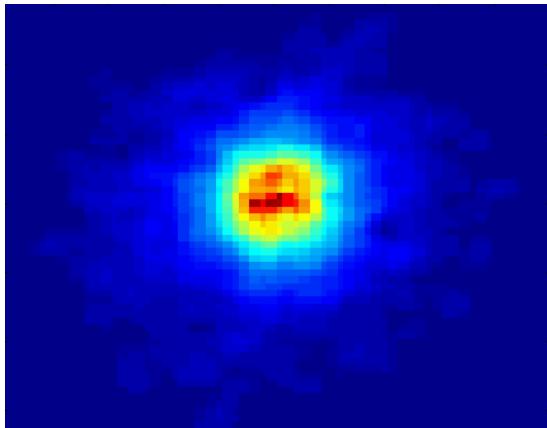
- Gök Haritası Kataloğu (Sky Survey Cataloging)
  - **Amaç:** Teleskopik inceleme görüntülerine (Palomar Gözlemevi'nden) dayalı olarak gökyüzü nesnelerinin, özellikle görsel olarak soluk olanların sınıfını (yıldız veya galaksi) tahmin etmek.
    - 3000 images with 23,040 x 23,040 pixels per image.
  - **Yaklaşım:**
    - ◆ Görüntüyü segmentlere ayırin.
    - ◆ Görüntü özniteliklerini (ozellikler) ölçün - nesne başına 40 tane.
    - ◆ Sınıfı bu özelliklere göre modelleyin.
    - ◆ Başarı Hikayesi: Bulması zor olan en uzak nesnelerden biri olan (galaksi dışında) 16 yeni kırmızı yıldızlı gökcism (red-shift quasars) bulunabildi!

From [Fayyad, et.al.] Advances in Knowledge Discovery and Data Mining, 1996

# Classifying Galaxies

Courtesy: <http://aps.umn.edu>

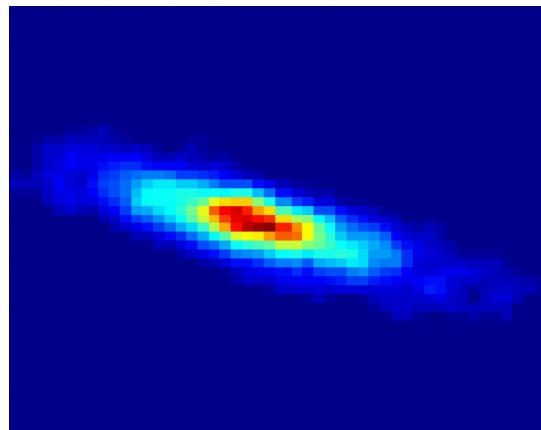
*Early*



**Class:**

- Stages of Formation

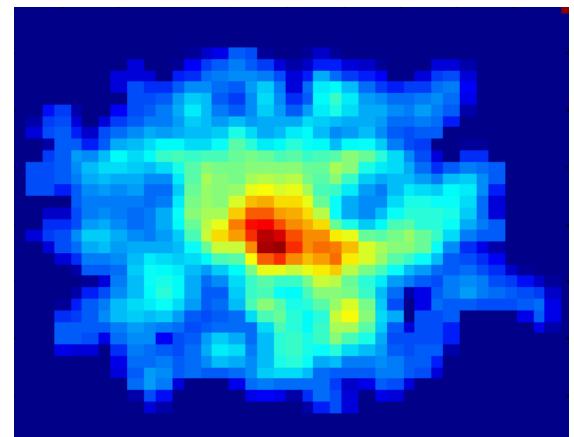
*Intermediate*



**Attributes:**

- Image features,
- Characteristics of light waves received, etc.

*Late*



**Data Size:**

- 72 million stars, 20 million galaxies
- Object Catalog: 9 GB
- Image Database: 150 GB

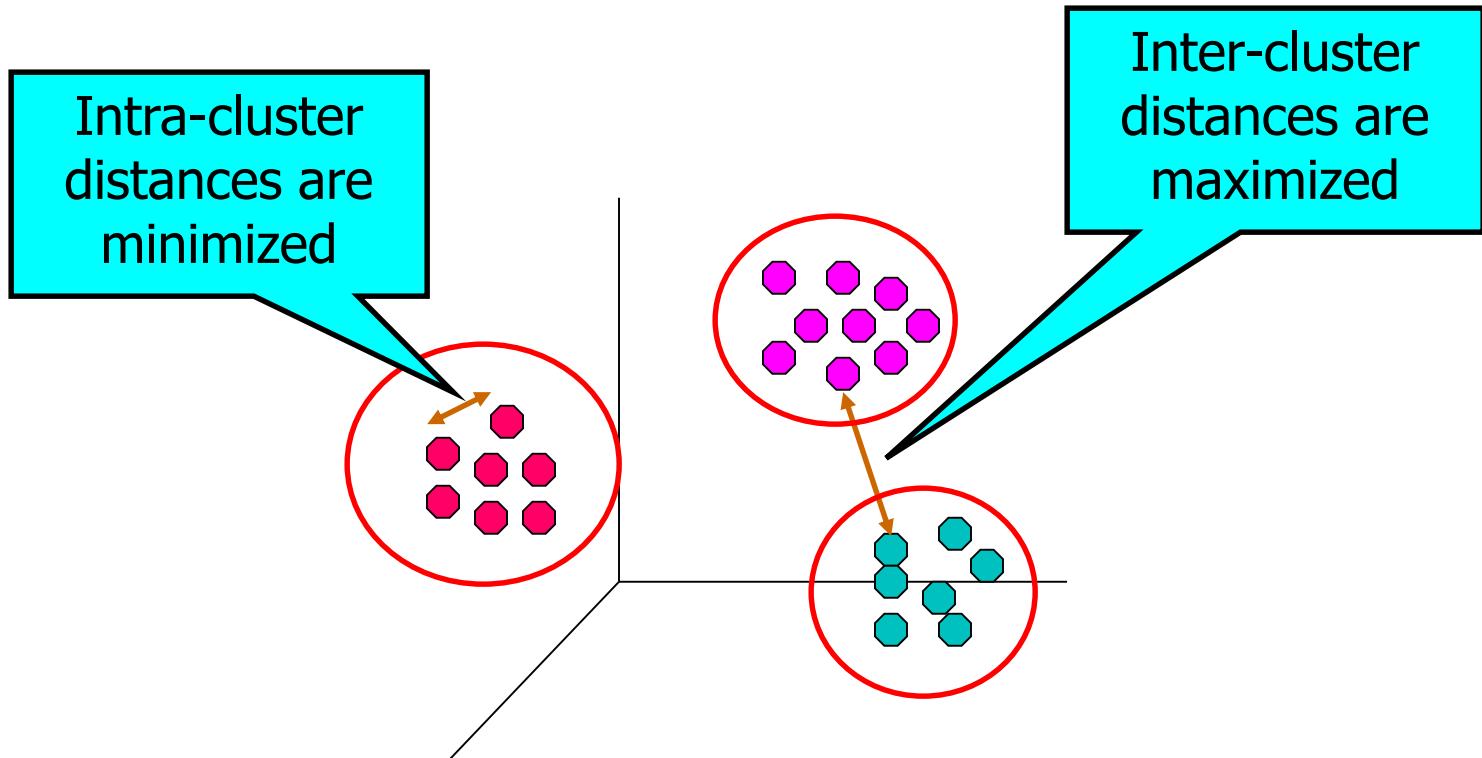
# Regression

---

- Doğrusal veya doğrusal olmayan bir bağımlılık modeli varsayıarak, belirli bir sürekli değerli değişkenin değerini diğer değişkenlerin değerlerine göre tahmin etmek
- İstatistik ve sinir ağı alanlarında üzerinde yoğun bir şekilde çalışılmıştır.
- Örnekler:
  - Reklam harcamalarına dayalı olarak yeni ürünün satış miktarlarını tahmin etme.
  - Sıcaklık, nem, hava basıncı vb. nin bir fonksiyonu olarak rüzgar hızlarını tahmin etme
  - Borsa endekslerinin zaman serisi tahmini.

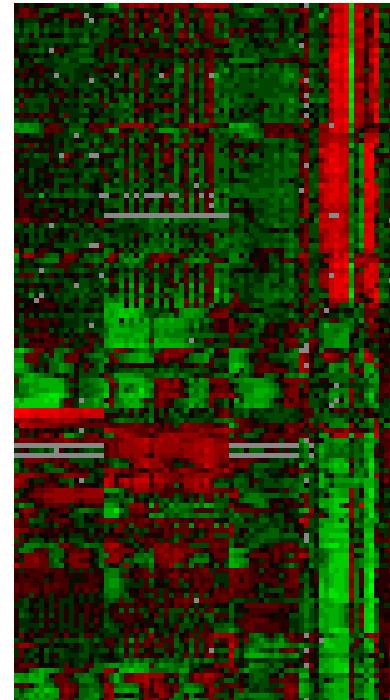
# Clustering (Kümeleme)

- Bir gruptaki nesnelerin birbirine benzeyeceği (veya ilişkilendirileceği) ve diğer grplardaki nesnelerden farklı (veya ilgisiz) olduğu nesne gruplarını bulma

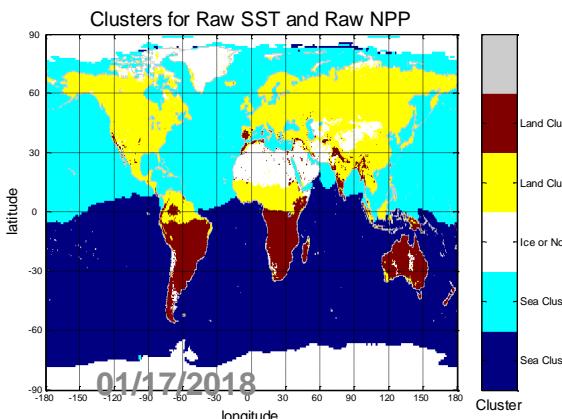
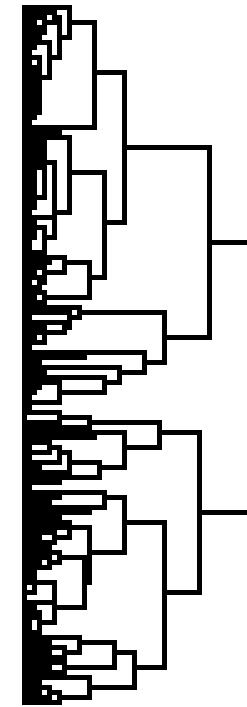


# Applications of Cluster Analysis

- **Anlama (Understanding)**
  - Hedeflenen pazarlar için özel profil oluşturma
  - «Browsing» için ilgili belgeleri gruplama
  - Benzer işlevsellîğe sahip genleri ve proteinleri gruplama
  - Benzer fiyat dalgalanmalarına sahip hisse senetlerini gruplama
- **Özetleme(Summarization)**
  - Büyük veri kümelerinin boyutunu küçültme

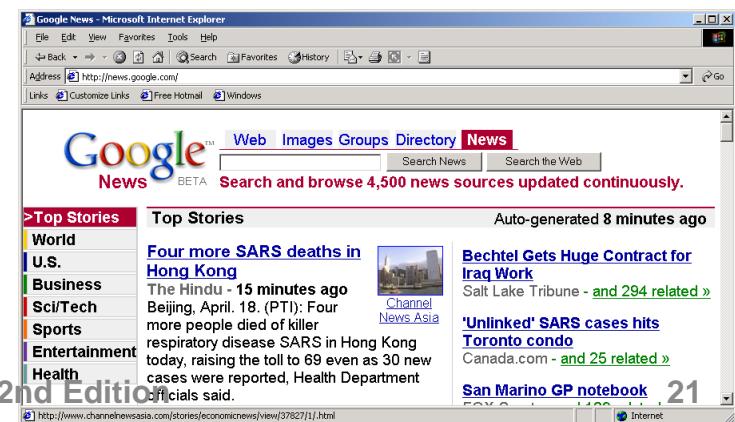


Courtesy: Michael Eisen



K-means yönteminin, Deniz Yüzeyi Sıcaklığı (SST) ve Net Birincil Üretimi (NPP) Kuzey ve Güney Yarımküre'yi yansitan kümelere ayırmak için kullanılması.

Introduction to Data Mining, 2nd Edition



# Clustering: Application 1

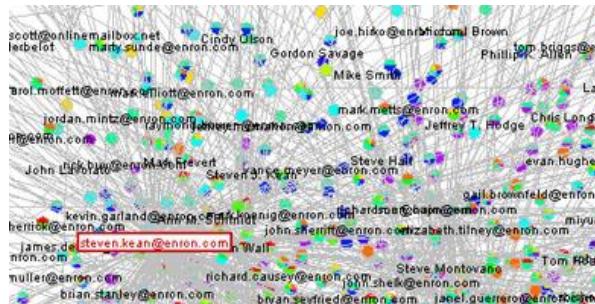
---

- Pazar Bölümlemesi (Market Segmentation):
  - Amaç: herhangi bir alt kümenin farklı bir pazarlama karmasıyla ulaşılacak bir pazar hedefi olarak seçilebileceği bir pazarın farklı müşteri alt kümelerine bölünmesi.
  - Yaklaşım:
    - ◆ Coğrafi ve yaşam tarzı ile ilgili bilgilere dayanarak müşterilerin farklı özelliklerini toplayın.
    - ◆ Benzer müşteri kümelerini bulun.
    - ◆ Farklı kümelerdekilerle aynı kümedeki müşterilerin satın alma örüntülerini gözlemleyerek kümeleme kalitesini ölçün.

# Clustering: Application 2

- Document Clustering:
  - **Amaç:** İçinde geçen önemli terimlere dayalı olarak birbirine benzeyen belge gruplarını bulmak
  - **Yaklaşım :** Her bir belgede sık görülen terimleri tanımlayıp farklı terimlerin frekanslarına dayalı bir benzerlik ölçüsü oluşturun ve bunları kümeleme için kullanın.

Enron email dataset



# Association Rule Discovery: Definition

## (Birliktelik Kuralı Keşfi)

- Her biri belirli bir koleksiyondan birkaç öğe içeren bir kayıt kümesi verildiğinde
  - Diğer öğelerin olma durumlarına dayalı olarak bir öğenin olmasını tahmin edecek bağımlılık kuralları üretmek

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Rules Discovered:

$\{\text{Milk}\} \rightarrow \{\text{Coke}\}$

$\{\text{Diaper}, \text{Milk}\} \rightarrow \{\text{Beer}\}$

# Association Analysis: Applications (Birliktelik analizi)

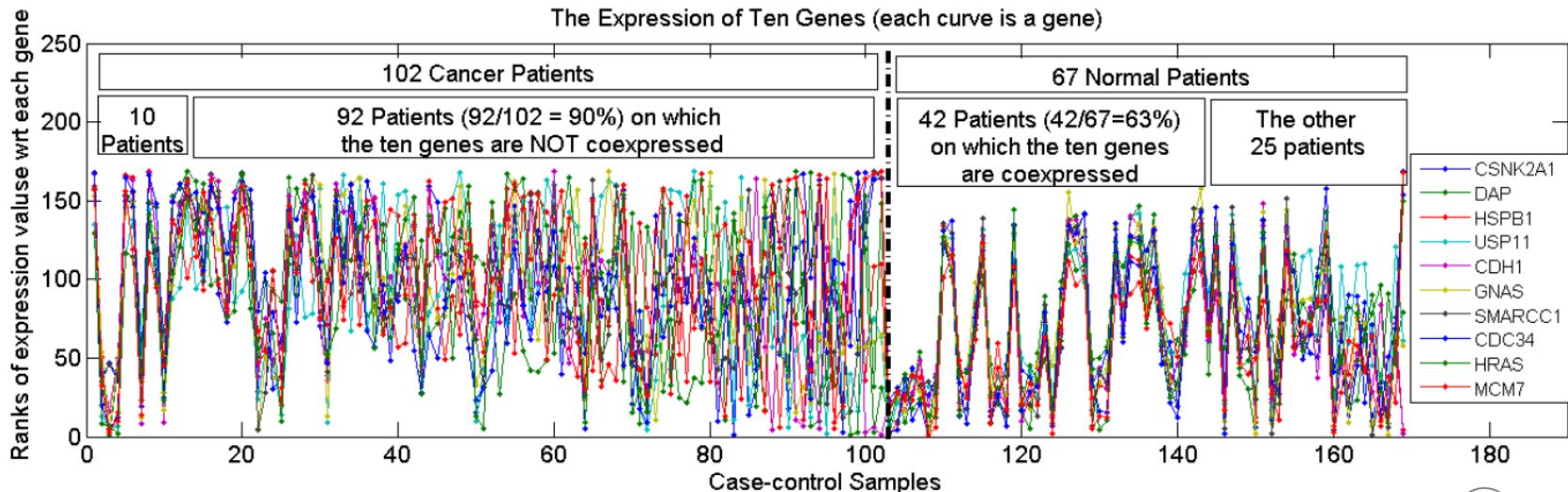
---

- Market sepeti analizi
  - Kurallar; satış promosyonu, raf yönetimi ve envanter yönetimi için kullanılır
- Telekomünikasyon alarm teşhisı
  - Kurallar, aynı zaman aralığında sık sık meydana gelen alarmların birleşimini bulmak için kullanılır
- Medical Informatics
  - Kurallar, hasta semptomları ve bazı hastalıklarla ilişkili test sonuçlarının kombinasyonunu bulmak için kullanılır

# Association Analysis: Applications

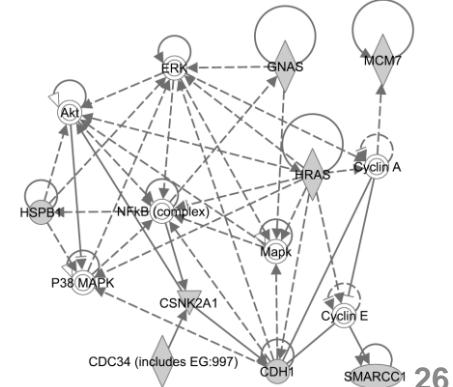
- An Example Subspace Differential Coexpression Pattern from lung cancer dataset

## Three lung cancer datasets [Bhattacharjee et al. 2001], [Stearman et al. 2005], [Su et al. 2007]



Enriched with the TNF/NFB signaling pathway which is well-known to be related to lung cancer P-value:  $1.4 \times 10^{-5}$  (6/10 overlap with the pathway)

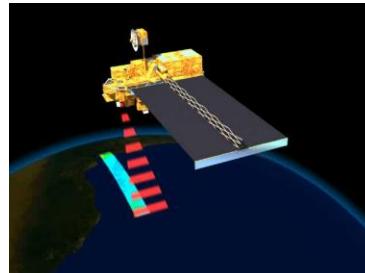
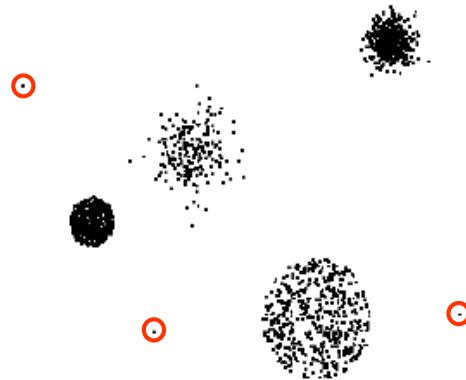
[Fang et al PSB 2010]



# Deviation/Anomaly/Change Detection

## (Sapma/Analomali/Değişim tespiti)

- Normal davranıştan önemli derecedeki sapmaları tespit etmek
- Applications:
  - Kredi Kartı Sahtekarlık Tespiti
  - İzinsiz Ağ (Network) Giriş Tespiti
  - İzleme ve gözetim için kullanılan sensör ağlarından gelen anormal davranışları belirlemek
  - Küresel orman örtüsündeki değişiklikleri tespit etmek



# Motivating Challenges

---

---

- Scalability
- High Dimensionality
- Heterogeneous and Complex Data
- Data Ownership and Distribution
- Non-traditional Analysis

# Data Mining: Data

---

---

## Lecture Notes for Chapter 2

Introduction to Data Mining

by

Tan, Steinbach, Kumar

Orijinal slaytların Türkçe çevirisidir.

# What is Data?

- **Veri nesneleri** ve onların **özniteliklerinin** koleksiyonu
- Öznitelik (**attribute**), bir nesnenin karakteristiği veya özelliği. Örnek: kişinin göz rengi, sıcaklık, vb.
  - Attribute is also known as variable, field, characteristic, or feature
- Bir öznitelik koleksiyonu bir nesneyi (**object**) tanımlar
  - Object is also known as record, point, case, sample, entity, or instance

**Attributes**

**Objects**

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

# Attribute Values (Öznitelik değerleri)

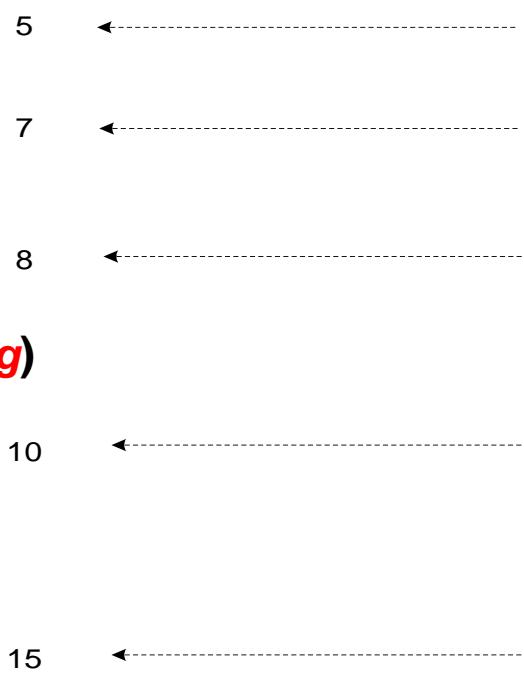
---

- Öznitelik değerleri, bir öznitelijke atanan sayılar veya sembollerdir
- Öznitelikler ve öznitelik değerleri arasındaki ayırım
  - Aynı öznitelik farklı öznitelik değerlerine izdüşürülebilir
    - ◆ Örnek: yükseklik metre veya feet olarak ölçülebilir
  - Farklı öznitelikler aynı değer kümesine eşlenebilir
    - ◆ Örnek: Kimlik numarası (ID) ve yaş (age) için öznitelik değerleri tamsayıdır
    - ◆ Fakat öznitelik değerlerinin özellikleri farklı olabilir
      - Kimlik numarasında sınırlama yoktur ancak yaşı maksimum ve minimum değeri vardır

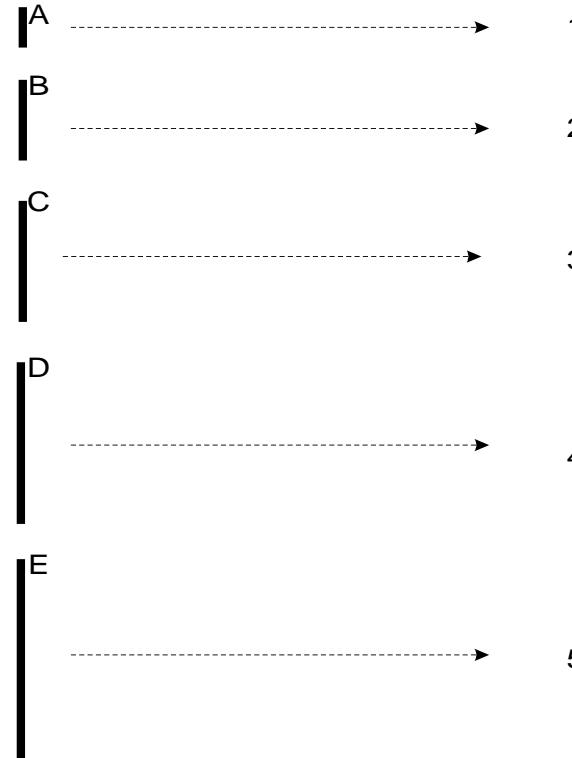
# Measurement of Length

- Bir özniteliği ölçme şekliniz, öznitelik özellikleriyle eşleşmeyebilir.

Bu ölçek yalnızca uzunluğun sıra (*ordering*) özelliğini korur.



A mapping to lengths to numbers that captures only the *order* properties of length



A mapping to lengths to numbers that captures both *order* and *additivity* properties of length

Bu ölçek uzunluğun sıra (*ordering*) ve toplanırlık (*additivity*) özelliğini korur.

*Thus, an attribute can be measured in a way that does not capture all the properties of the attribute.*

# Types of Attributes

---

- There are different types of attributes
  - Nominal
    - ◆ Examples: ID numbers, eye color, zip codes
  - Ordinal
    - ◆ Examples: rankings (e.g., taste of potato chips on a scale from 1-10), grades, height in {tall, medium, short}
  - Interval
    - ◆ Examples: calendar dates, temperatures in Celsius or Fahrenheit.
  - Ratio
    - ◆ Examples: temperature in Kelvin, length, time, counts

# Properties of Attribute Values

## (Öznitelik değerlerinin özellikleri)

- Bir (öz)niteliğin türü, aşağıdaki özelliklerden hangisine sahip olduğuna bağlıdır :
  - Distinctness:                    $= \neq$
  - Order:                           $< >$
  - Addition:                        $+ -$
  - Multiplication:                $* /$
  - Nominal attribute: distinctness
  - Ordinal attribute: distinctness & order
  - Interval attribute: distinctness, order & addition
  - Ratio attribute: all 4 properties

Attribute Type	Description	Examples	Operations
Nominal	Nominal bir niteliğin değerleri sadece farklı isimlerdir, yani nominal nitelikler sadece bir nesneyi diğerinden ayırt etmek için yeterli bilgi sağlar. ( $=, \neq$ )	zip codes, employee ID numbers, eye color, sex: $\{male, female\}$	mode, entropy, contingency correlation, $\chi^2$ test
Ordinal	Bir ordinal niteliğin değerleri, nesneleri sıralamak için yeterli bilgi sağlar. ( $<, >$ )	hardness of minerals, $\{good, better, best\}$ , grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
Interval	Aralık (Interval) nitelikleri için, değerler arasındaki farklar anlamlıdır, yani bir ölçü birimi mevcuttur. $(+, -)$	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation, Pearson's correlation, $t$ and $F$ tests
Ratio	Oran (Ratio) değişkenleri için, hem farklar hem de oranlar anlamlıdır. $(*, /)$	temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current	geometric mean, harmonic mean, percent variation

# This categorization of attributes is due to S. S. Stevens

**Categorical (or qualitative)**

**attribute**

**Numeric (Quantitative)**

**attributes**

Attribute Level	Transformation	Comments
Nominal	Her türlü permütasyon (Any permutation of values)	Tüm çalışan kimlik numaraları (ID) yeniden atansa, herhangi bir fark yaratır mı?
Ordinal	Değerlerin sırasını muhafaza eden bir değişiklik, yani $new\_value = f(old\_value)$ burada f monotonik bir fonksiyondur.	İyi, daha iyi en iyi kavramını kapsayan bir öznitelik, başka değerlerle de aynı şekilde temsil edilebilir {1, 2, 3} veya { 0.5, 1, 10} ile
Interval	$new\_value = a * old\_value + b$ burada $a$ ve $b$ sabitdir	Buradan hareketle, Fahrenheit ve Santigrat sıcaklık ölçekleri sıfır değerlerinin nerede olduğu ve bir birimin (derece) büyülüğu açısından farklılık gösterir.
Ratio	$new\_value = a * old\_value$	Uzunluk metre veya feet olarak ölçülebilir.

*Nitelik türleri, bir niteliğin anlamını değiştirmeyen dönüşümler (transformations) olarak da tanımlanabilir.*

# Discrete and Continuous Attributes

---

## ● Ayrık Nitelik (Discrete Attribute)

- Sonlu bir değer kümesine sahiptir
- Örnekler: posta kodu, sayılar veya bir belge koleksiyonundaki kelime kümesi
- Genellikle tamsayı değişkenleri olarak gösterilir.
- Not: ikili öznitelikler (**binary attributes**), ayrık özniteliklerin özel bir durumudur
  - ◆ Sadece iki değer alır, e.g., true/false, yes/no, male/female, or 0/1.

## ● Sürekli Nitelik (Continuous Attribute)

- Öznitelik değerleri olarak gerçek sayılar vardır
- Örnek: sıcaklık, yükseklik veya ağrılık.
- Pratikte, gerçek değerler sadece sınırlı sayıda basamak kullanılarak ölçülebilir ve temsil edilebilir.
- Sürekli öznitelikler genellikle kayan nokta değişkenleri olarak temsil edilir.

# Asymmetric Attributes

---

- Yalnızca varoluş/mevcudiyet (sıfır olmayan bir öznitelik değeri) önemli olarak kabul edilir
  - ◆ Dokumanlarda geçen kelimeler
  - ◆ Müşteri işlemlerinde mevcut olan kalemler
- Markette bir arkadaşla karşılaşsak şunu söyler miydik?  
*“Aynı şeylerin çoğunu almadığımız için alımlarımızın çok benzer olduğunu görüyorum.”*
- Sıfır olmayan değerlere odaklanmak daha anlamlı ve daha verimlidir.
- Sadece sıfır olmayan değerlerin önemli olduğu ikili niteliklere asimetrik ikili öznitelikler (**asymmetric binary attributes**) denir.
  - Birliktelik analizinde asimetrik öznitelikler kullanılır.

# Types of data sets

---

- **Record**

- Data Matrix
- Document Data
- Transaction Data

- **Graph-based**

- World Wide Web
- Molecular Structures

- **Ordered**

- Spatial Data
- Temporal Data
- Sequential Data
- Genetic Sequence Data

# Important Characteristics of Data

---

- **Dimensionality** (number of attributes)
  - ◆ Curse of Dimensionality
- **Sparsity**
  - ◆ Only presence counts
- **Resolution**
  - ◆ Patterns depend on the scale
- **Size**
  - ◆ Type of analysis may depend on size of data

# Record Data

- Her biri sabit bir öznitelik kümesinden (**fixed set of attributes**) oluşan kayıt koleksiyonu

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

# Data Matrix

- Veri nesneleri aynı sabit sayısal öznitelik kümese sahipse, veri nesneleri (**data objects**) çok boyutlu bir uzayda noktalar (**points in a multi-dimensional space**) olarak düşünülebilir; burada her boyut farklı bir özniteligi temsil eder
- Bu veri seti, her nesne için bir tane olmak üzere **m satır** ve her bir öznitelik için bir tane olmak üzere **n sütun** ile, yani bir **mxn** matrisi ile temsil edilebilir.

Projection of x Load	Projection of y load	Distance	Load	Thickness
10.23	5.27	15.22	2.7	1.2
12.65	6.25	16.22	2.2	1.1

# Document Data

---

- Her belge bir 'terim' vektörü olur
  - her terim, vektörün bir bileşenidir (özniteligidir)
  - her bileşenin değeri, karşılık gelen terimin belgede kaç kez geçtiğini gösterir.

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

# Transaction Data

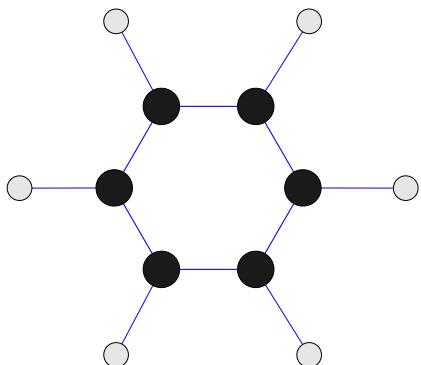
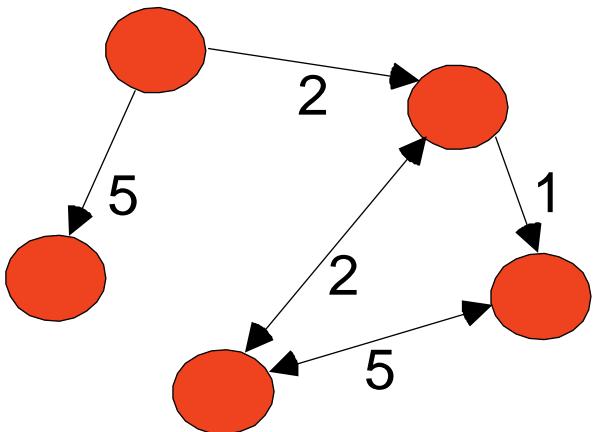
---

- Özel bir kayıt verisi türü, burada
  - Her kayıt (işlem/ transaction) bir dizi maddeyi içerir.
  - Örneğin, bir marketi düşünün. Bir müşterinin bir alışveriş gezisi sırasında satın aldığı ürün grubu bir işlem (**transaction**) oluştururken, satın alınan tekil ürünler öğelerdir (**items**).

<i>TID</i>	<i>Items</i>
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

# Graph Data

- Examples: Generic graph, a molecule, and webpages



Benzene Molecule: C<sub>6</sub>H<sub>6</sub>

## Useful Links:

- [Bibliography](#)
- Other Useful Web sites
  - [ACM SIGKDD](#)
  - [KDnuggets](#)
  - [The Data Mine](#)

## Knowledge Discovery and Data Mining Bibliography

(Gets updated frequently, so visit often!)

- [Books](#)
- [General Data Mining](#)

## Book References in Data Mining and Knowledge Discovery

Usama Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Ithurusamy, "Advances in Knowledge Discovery and Data Mining", AAAI Press/the MIT Press, 1996.

J. Ross Quinlan, "C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers, 1993. Michael Berry and Gordon Linoff, "Data Mining Techniques (For Marketing, Sales, and Customer Support)", John Wiley & Sons, 1997.

## General Data Mining

Usama Fayyad, "Mining Databases: Towards Algorithms for Knowledge Discovery", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, vol. 21, no. 1, March 1998.

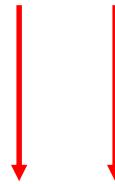
Christopher Matheus, Philip Chan, and Gregory Piatetsky-Shapiro, "Systems for knowledge Discovery in databases", IEEE Transactions on Knowledge and Data Engineering, 5(6):903-913, December 1993.

# Ordered Data

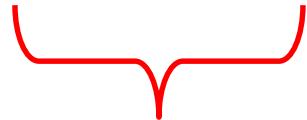
---

- Sequences of transactions

**Items/Events**



( A B)	(D)	(C E)
( B D)	(C)	(E)
( C D)	(B)	(A E)



**An element of  
the sequence**

# Ordered Data

---

- Genomic sequence data (Gen dizilim verisi)

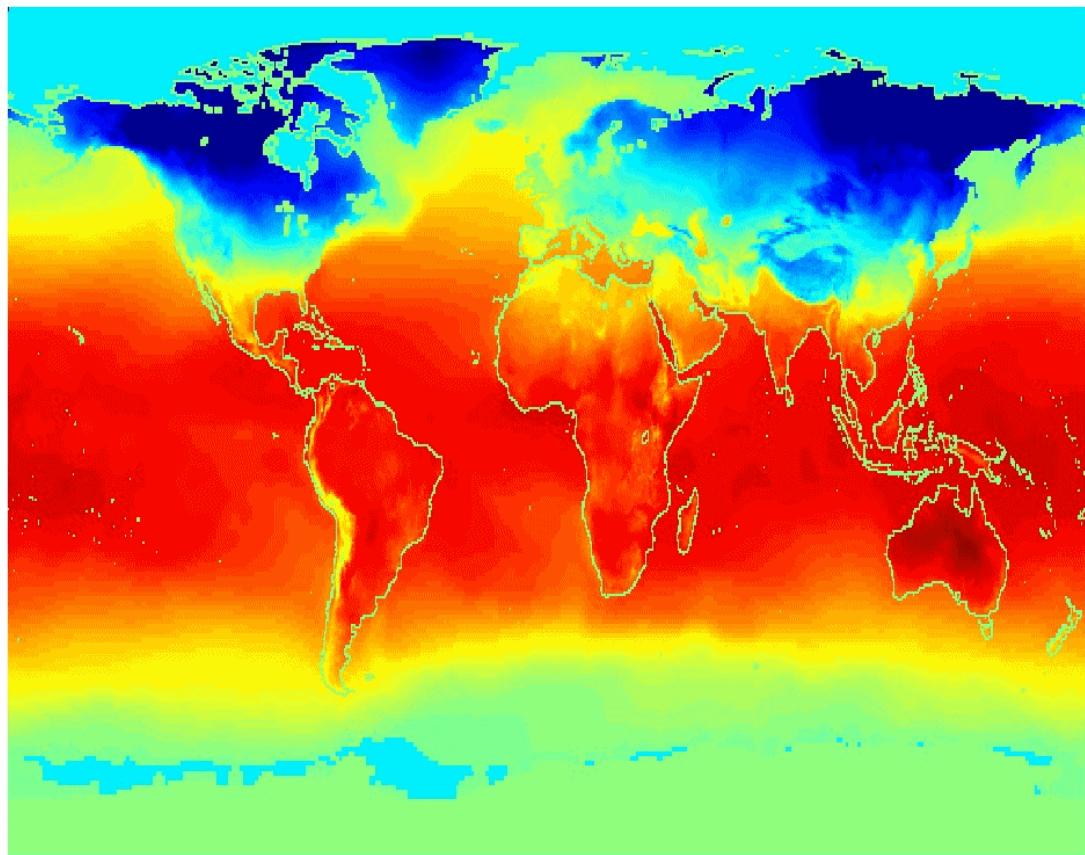
```
GGTTCCGCCTTCAGCCCCGCGCC  
CGCAGGGCCCAGCCCCGCCGCCGTC  
GAGAAGGGCCCGCCTGGCGGGCG  
GGGGGAGGCAGGGGCCGCCGAGC  
CCAACCGAGTCCGACCAAGGTGCC  
CCCTCTGCTCGGCCTAGACCTGA  
GCTCATTAGGCAGCAGCGGACAG  
GCCAAGTAGAACACCGCGAAGCGC  
TGGGCTGCCTGCTGCGACCAGGG
```

# Ordered Data

- Spatio-Temporal Data

Kara ve  
okyanusların  
Aylık Ortalama  
Sıcaklık verisi

Jan



# Data Quality

- Yetersiz veri kalitesi, birçok veri işleme çabasını olumsuz etkiler

“En önemli nokta, düşük veri kalitesinin gelişen bir felaket olmasıdır.

Düşük veri kalitesi, tipik bir şirketin gelirinin en az yüzde onuna (%10) mal olur; Yüzde yirmi (%20) muhtemelen daha iyi bir tahmin.”

Thomas C. Redman, DM Review, August 2004

- Veri madenciliği örneği: kredi riski olan kişileri tespit etmek için bir sınıflandırma modeli yetersiz/eksik veriler kullanılarak oluşturulmuştur
  - Bazı krediye değer adaylarının kredileri reddedildi
  - Temerrüde düşen kişilere daha fazla kredi verildi

# Data Quality

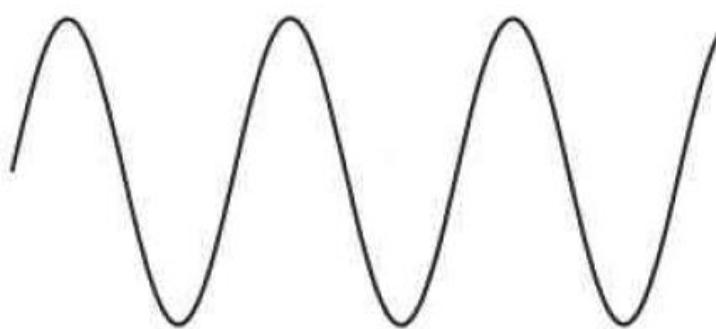
---

---

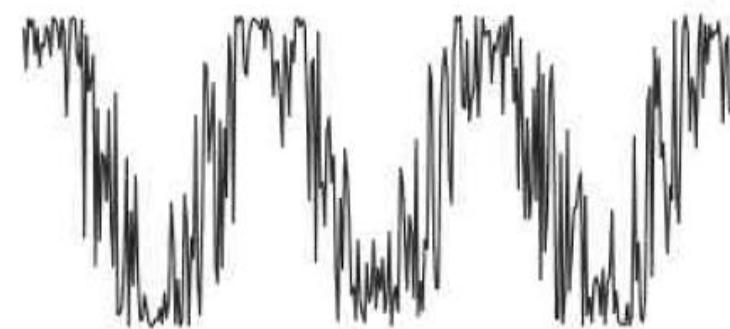
- Ne tür veri kalitesi sorunları?
  - Verilerle ilgili sorunları nasıl tespit edebiliriz?
  - Bu sorunlar hakkında neler yapabiliriz?
- 
- Examples of data quality problems:
    - Noise and outliers
    - missing values
    - duplicate data

# Noise

- Gürültü (***noise***), orijinal değerlerin değiştirilmesi anlamına gelir
  - Örnekler: kaklitesiz bir telefonda konuşurken kişinin sesinde bozulma ve televizyon ekranında "karlanma"



(a) Time series.

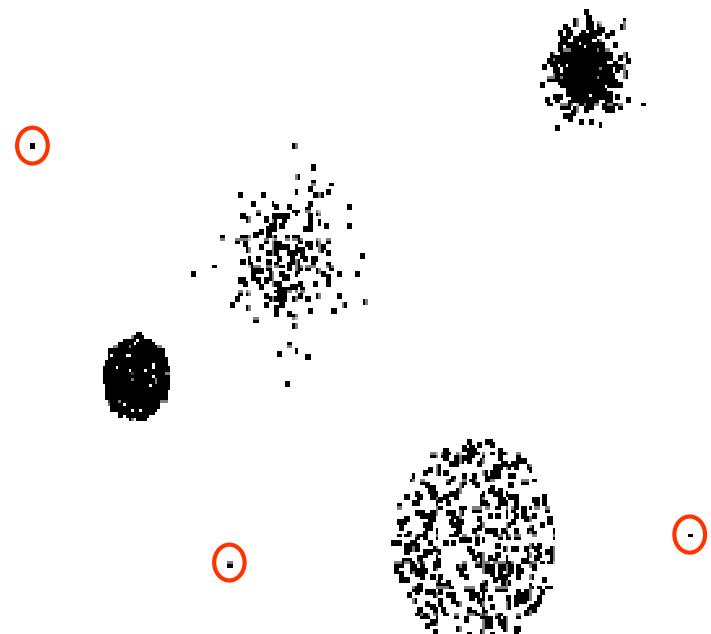


(b) Time series with noise.

**Figure 2.5.** Noise in a time series context.

# Outliers

- **Outliers** (uç/aykırı değerler) veri kümesindeki diğer veri nesnelerinin çoğundan önemli ölçüde farklı özelliklere sahip veri nesneleridir
  - **Case 1:** *Outliers*, veri analizine müdahale eden gürültüdür
  - **Case 2:** *Outliers* analizimizin hedefidir
    - ◆ Credit card fraud
    - ◆ Intrusion detection



# Missing Values

- Eksik değerlerin nedenleri
  - Bilginin toplanamadığı durumlar  
(ör. insanlar **yaşlarını** ve **kilolarını** vermeyi reddederler)
  - Nitelikler tüm durumlar için geçerli olmayabilir  
(ör. **yıllık gelir çocukların** için geçerli değildir)

- Eksik verilerle başa çıkma

- Eliminate Data Objects
- Estimate Missing Values
- Ignore the Missing Value During Analysis
- Replace with all possible values (weighted by their probabilities)

Age	Income	Team	Gender
23	24,200	Red Sox	M
39	?	Yankees	F
45	45,390	?	F

? : missing value

# Duplicate Data

---

- Veri kümesi, yinelenen (*duplicate*) veya neredeyse birbirinin kopyası olan veri nesnelerini içerebilir
  - Heterojen kaynaklardan gelen verileri birleştirirken önemli sorun
- Örnekler :
  - Birden çok e-posta adresine sahip aynı kişi:
- Data cleaning (Veri temizleme)
  - Tekrarlı veri sorunlarıyla ilgilenme süreci

# Data Preprocessing

---

---

- Aggregation
- Sampling
- Dimensionality Reduction
- Feature Subset Selection
- Feature Creation
- Discretization and Binarization
- Attribute Transformation

# Aggregation

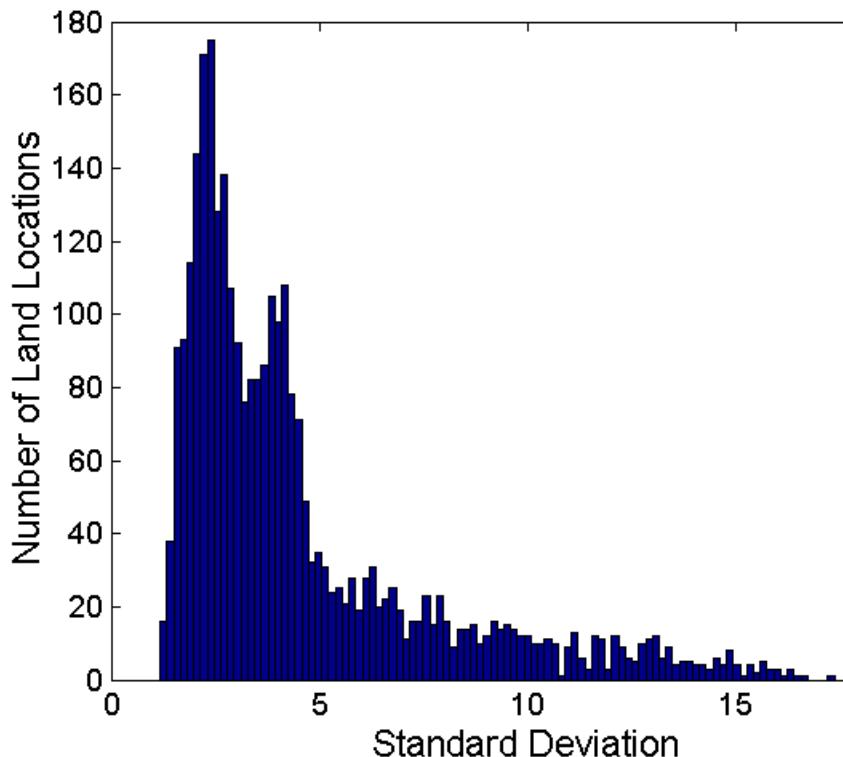
---

- İki veya daha fazla özniteliği (veya nesneyi) tek bir öznitelikte (veya nesnede) birleştirmek
- Amaç
  - Veri azaltma (*Data reduction*)
    - ◆ Özniteliklerin (attributes) veya nesnelerin (objects) sayısını azaltma
  - Ölçek değişikliği (*Change of scale*)
    - ◆ Bölgeler, eyaletler, ülkeler vb. şeklinde birleştirilmiş şehirler
  - Daha "kararlı" (*stable*) veriler
    - ◆ Birleştirilmiş veriler daha az değişkenliğe/oynaklığa sahip olma eğilimindedir

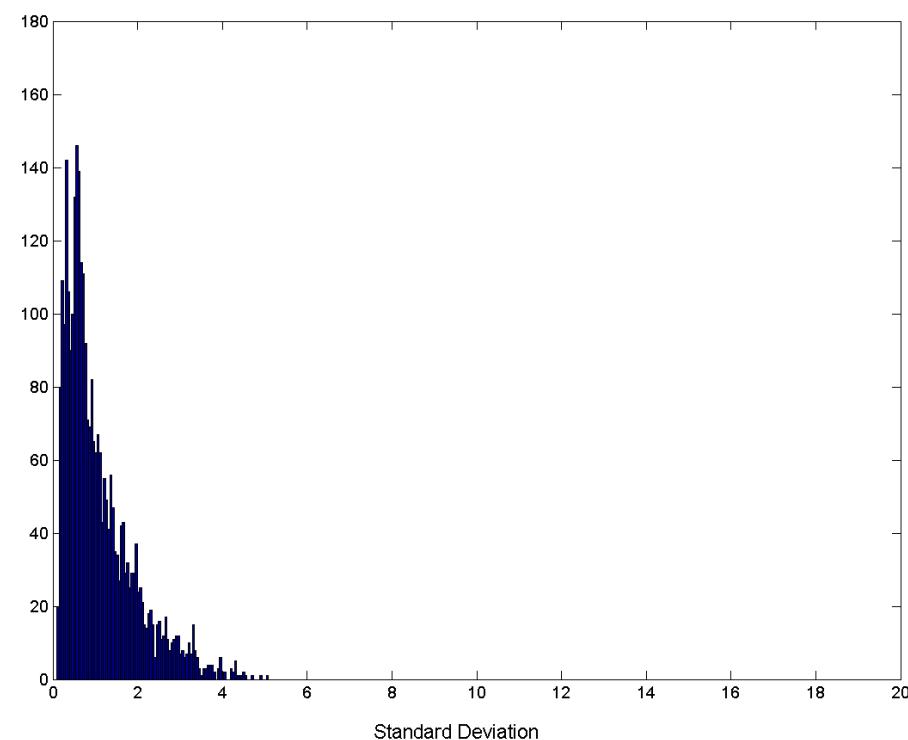
# Aggregation

## Avustralya'daki Yağış (Precipitation) Değişimi

*Aggregation sayesinde std. dev. miktarında belirgin azalma*



Ortalama Aylık Yağışların  
Standart Sapması



Ortalama Yıllık Yağışların  
Standart Sapması

# Sampling

---

- Veri seçimi (data selection) için kullanılan ana teknik **örneklemedir**.
  - Genellikle hem verilerin ön araştırması, hem de nihai veri analizi için kullanılır.
- İstatistikçiler örnekleme yapar çünkü ilgilenilen tüm veri setini **elde etmek** çok pahalı veya zaman alıcıdır.
- Örnekleme, veri madenciliğinde kullanılır çünkü ilgilenilen tüm veri kümесinin işlenmesi çok pahalı (**expensive**) veya zaman alıcıdır (**time consuming**).

# Sampling ...

---

- Etkili örnekleme için temel ilke şudur:
  - Eğer seçilen örneklem temsil gücü yüksek ise, **bir örneklem kullanmak neredeyse tüm veri setini kullanmak kadar** işe yarayacaktır.
  - Bir örneklem, orijinal veri kümesiyle yaklaşık olarak (ilgili) aynı özelliğe sahipse temsilcidir (representative).

# Types of Sampling

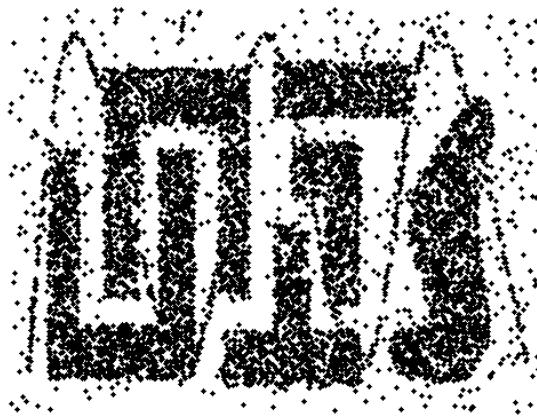
---

- Simple Random Sampling
  - Herhangi bir belirli öğeyi seçme konusunda eşit bir olasılık vardır
- Sampling without replacement
  - Her öğe seçildikçe popülasyondan çıkarılır.
- Sampling with replacement
  - Nesneler, örneklem için seçildikçe popülasyondan çıkarılmaz.
    - ◆ Aynı nesne birden fazla kez alınabilir.
- Stratified sampling
  - Verileri birkaç bölüme (*partition*) ayırin; sonra her bölümden rastgele örnekler alın

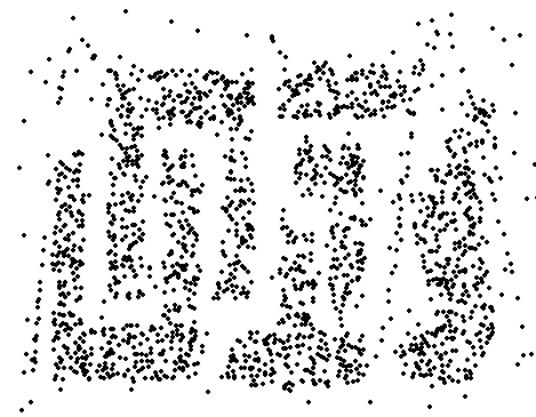
# Sample Size

---

---



**8000 points**



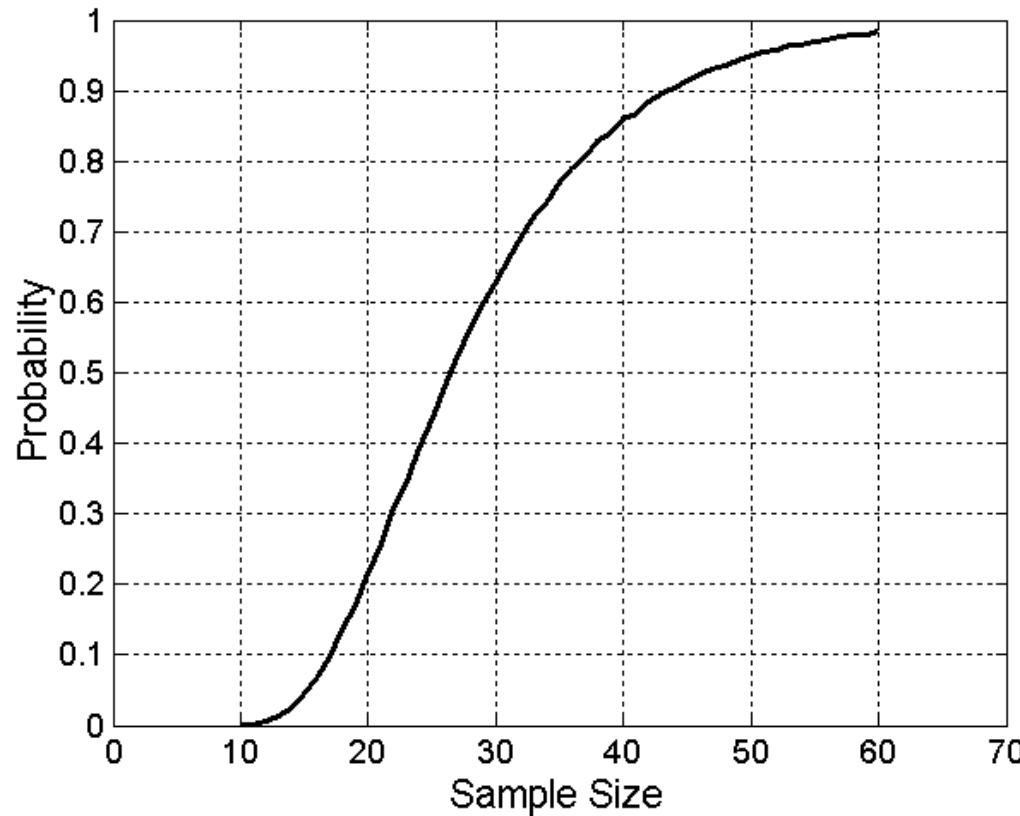
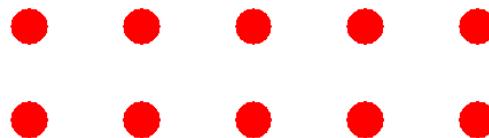
**2000 Points**



**500 Points**

# Sample Size

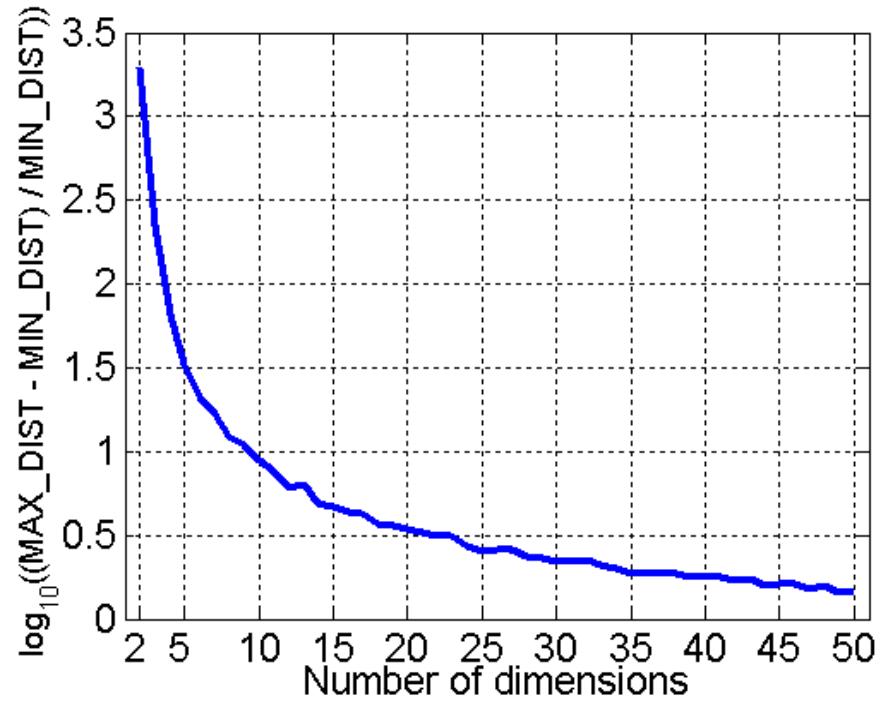
- 10 eşit büyüklükteki grubun her birinden en az bir nesne elde etmek için hangi örneklem boyutu gereklidir?



The figure showing an idealized set of clusters (groups) from which these points might be drawn

# Curse of Dimensionality

- Boyut arttığında (**dimensionality increases**), veri kapladığı alanda giderek daha seyrek (**sparse**) hale gelir
- Kümeyeleme (*clustering*) ve aykırı değer tespiti (*outlier detection*) için kritik olan yoğunluk (*density*) ve noktalar arasındaki mesafe tanımları **daha az anlamlı** hale gelir



- Rastgele 500 nokta oluşturun
- Herhangi bir nokta çifti arasındaki maksimum ve minimum mesafe arasındaki farkı hesaplayın

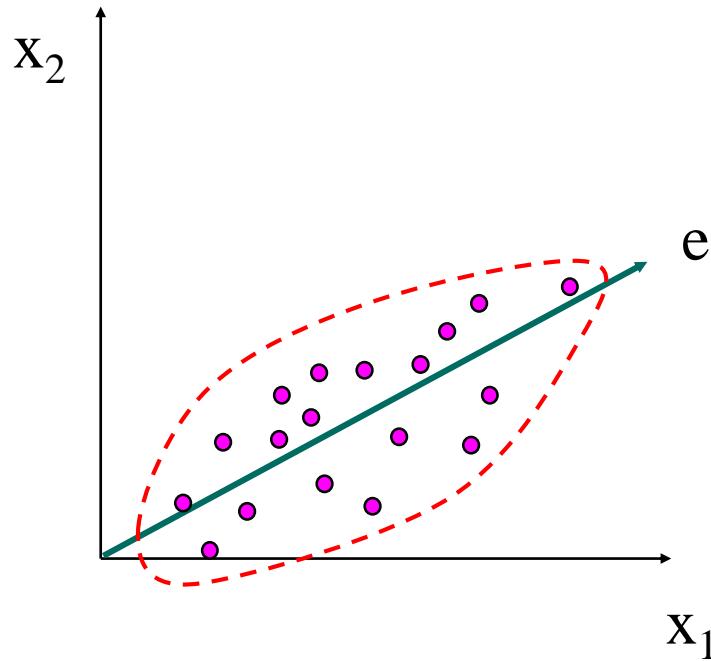
# Dimensionality Reduction

---

- Amaç:
  - Çok boyutluluğun getirdiği sıkıntıdan kurtulmak
  - Veri madenciliği algoritmalarının gerektirdiği süre (**time**) ve bellek (**memory**) miktarını azaltmak
  - Verilerin **daha kolay görselleştirilmesine** olanak tanır
  - Alakasız özellikler (**irrelevant features**) ortadan kaldırılmaya veya gürültüyü (**noise**) azaltmaya yardımcı olabilir
- Teknikler
  - Principle Component Analysis (PCA)
  - Singular Value Decomposition
  - Others: supervised and non-linear techniques

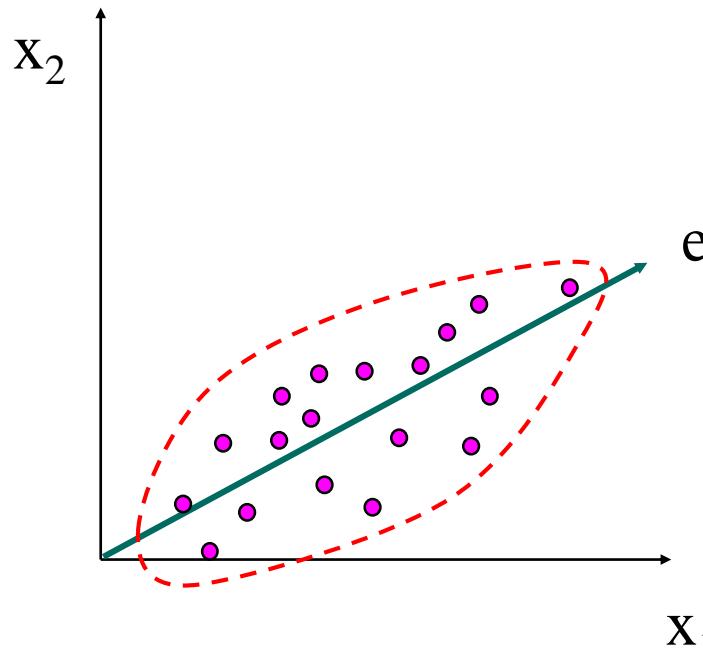
# Dimensionality Reduction: PCA

- Amaç, verilerdeki en büyük miktarda varyasyonu yakalayan bir projeksiyon bulmaktır.



# Dimensionality Reduction: PCA

- Kovaryans matrisinin özvektörlerini bulunur
- Özvektörler yeni uzayı tanımlar



# Dimensionality Reduction: PCA

---

- Temel Bileşenler Analizi (PCA) sürekli öznitelikler için yeni öznitelikler (**temel bileşenleri**) bulan bir lineer cebir tekniğidir ve bu bileşenler
  - (1) orijinal özelliklerin lineer kombinasyonlarıdır,
  - (2) birbirlerine dikdir (**orthogonal**)
  - (3) verilerdeki maksimum varyasyon mictarını yakalar

Örneğin, ilk iki temel bileşen, orijinal niteliklerin doğrusal kombinasyonları olan iki ortogonal nitelik ile mümkün olduğu kadar verideki varyasyonun çoğunu yakalar.

# Dimensionality Reduction: PCA

---

256



# Feature Subset Selection

---

- Verilerin boyutunu azaltmanın başka bir yolu
- **Redundant features** (yedekli özellikler)
  - bir veya daha fazla başka öznitelikte bulunan bilgilerin çoğunu veya tamamını tekrarlama (**duplicate**)
  - Örnek: bir ürünün satın alma fiyatı ve ödenen satış vergisi tutarı
- **Irrelevant features** (alakasız özellikler)
  - eldeki veri madenciliği görevi için yararlı hiçbir bilgi içermez
  - Örnek: öğrencilerin kimliği (ID) genellikle öğrencilerin not ortalamasını (GPA) tahmin etme görevi ile ilgisizdir

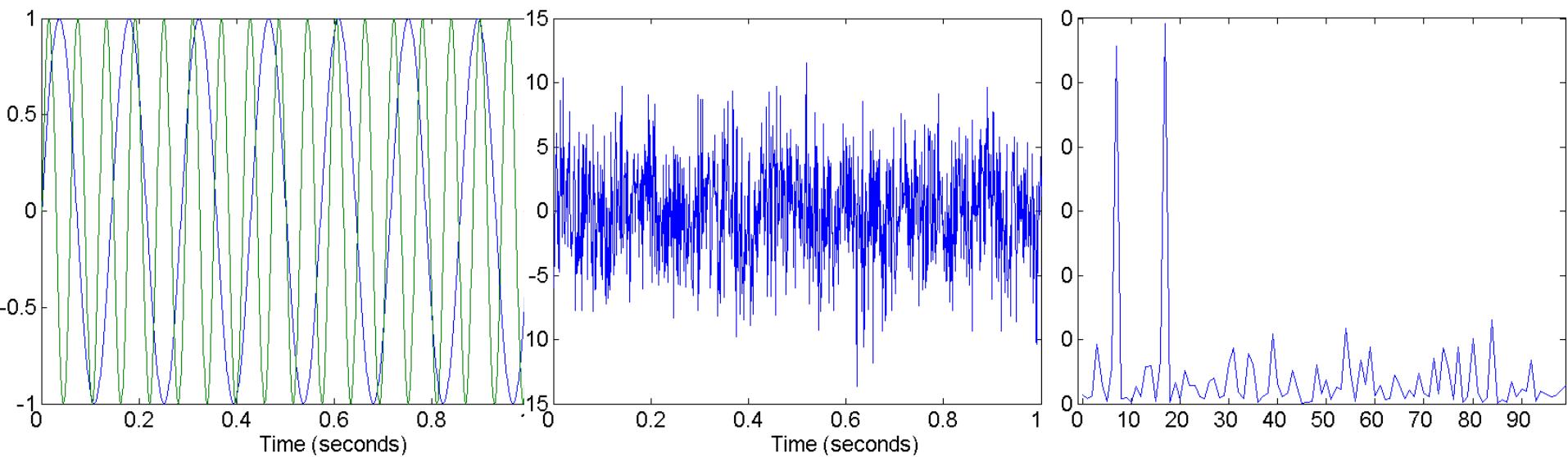
# Feature Creation

---

- Bir veri kümesindeki önemli bilgileri orijinal özniteliklerden çok daha verimli bir şekilde yakalayabilen yeni öznitelikler oluşturma
- Üç genel metodoloji :
  - Feature extraction (*öznitelik çıkarımı*)
    - ◆ Örnek: görüntülerden kenarları çıkarma
  - Feature construction (*öznitelik oluşturma*)
    - ◆ Örnek: yoğunluğu elde etmek için kütleyi hacme bölmeye
  - Mapping data to new space (*Verileri yeni uzaya izdüşürme*)
    - ◆ Örnek: Fourier and wavelet analizi

# Mapping Data to a New Space

- Fourier transform
- Wavelet transform



Two Sine Waves

Two Sine Waves + Noise

Frequency

# Discretization

---

- Ayrıklaştırma (**Discretization**), sürekli (**continuous**) bir özniteligi sırasal (**ordinal**) öznitelige dönüştürme sürecidir.
  - Potansiyel olarak sonsuz sayıda değer, az sayıda kategoriye eşlenir
  - Ayrıklaştırma genellikle sınıflandırmada kullanılır
  - Birçok sınıflandırma algoritması, hem **bağımsız** hem de **bağımlı değişkenleri** yalnızca birkaç değere sahipse **en iyi şekilde çalışır**
  - Iris veri setini kullanarak ayrıklaştımanın yararlılığına dair bir örnek...

# Iris Sample Data Set

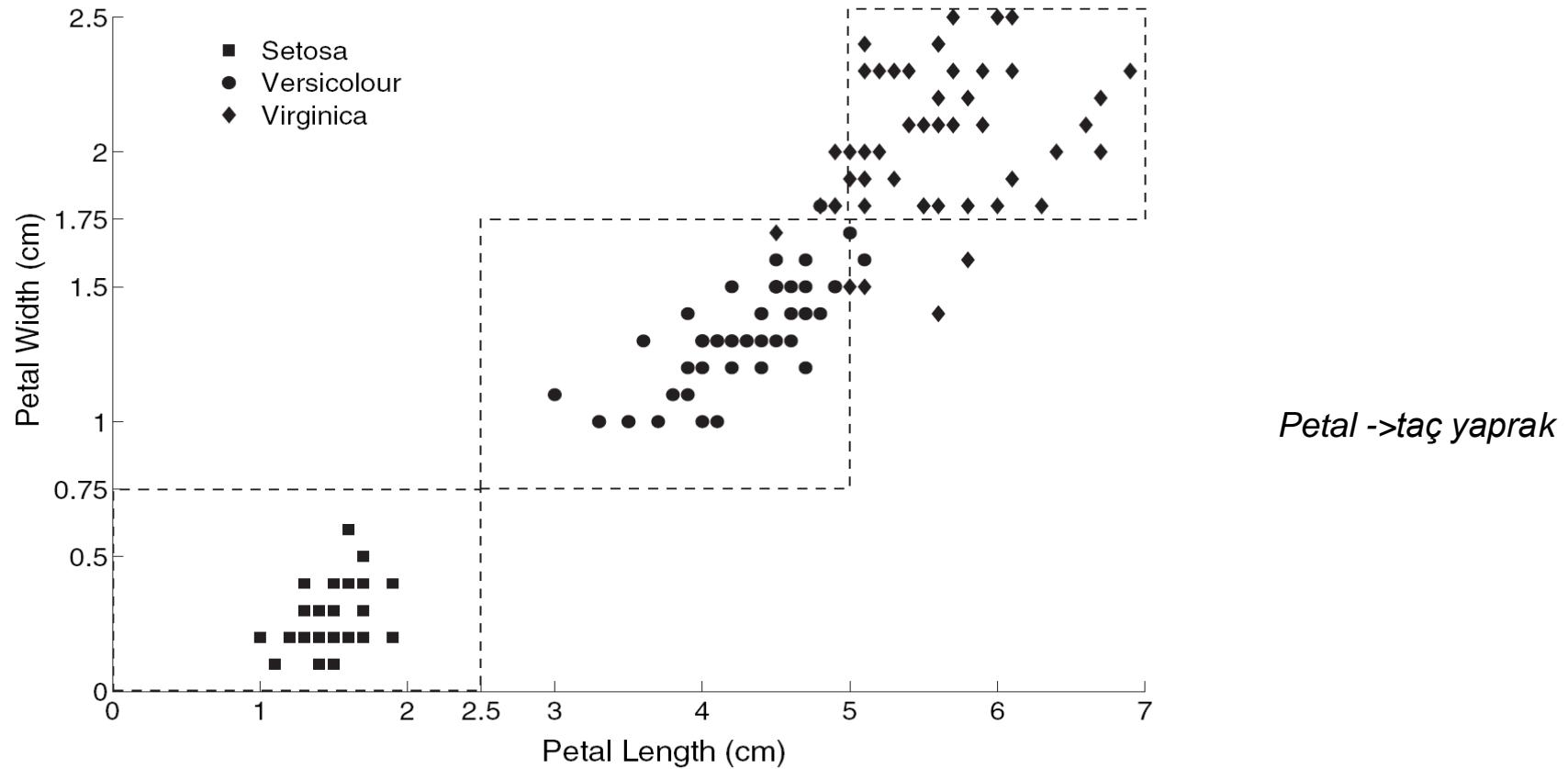
---

- Iris Plant data set.
  - Can be obtained from the UCI Machine Learning Repository  
<http://www.ics.uci.edu/~mlearn/MLRepository.html>
  - From the statistician Douglas Fisher
  - Three flower types (classes):
    - ◆ Setosa
    - ◆ Versicolour
    - ◆ Virginica
  - Four (non-class) attributes
    - ◆ Sepal width and length
    - ◆ Petal width and length



Virginica. Robert H. Mohlenbrock. USDA NRCS. 1995. Northeast wetland flora: Field office guide to plant species. Northeast National Technical Center, Chester, PA. Courtesy of USDA NRCS Wetland Science Institute.

# Discretization: Iris Example

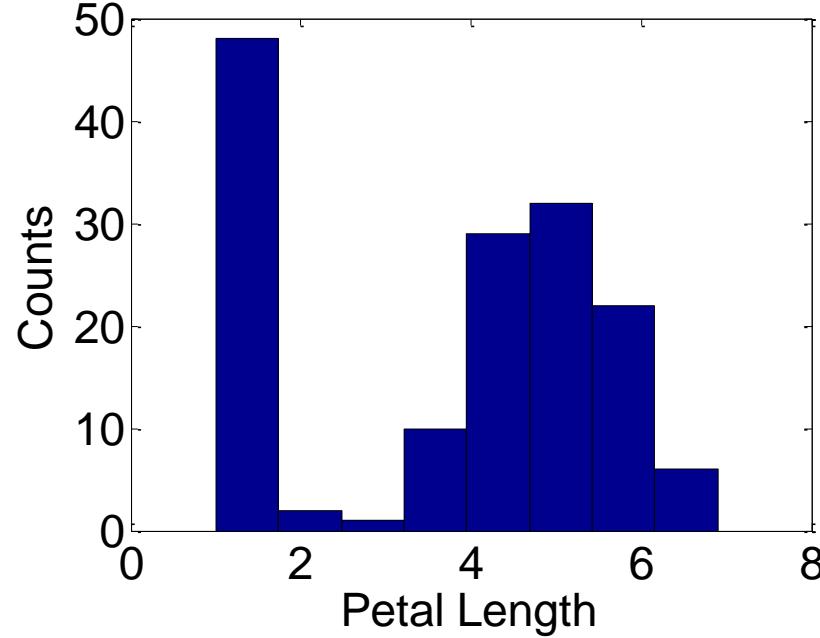


Petal genişliği düşük veya petal uzunluğu düşük, **Setosa** anlamına gelir.  
Petal genişliği orta veya petal uzunluğu orta, **Versicolour** anlamına gelir.  
Petal genişliği yüksek veya petal uzunluğu yüksek, **Virginica** anlamına gelir.

# Discretization: Iris Example ...

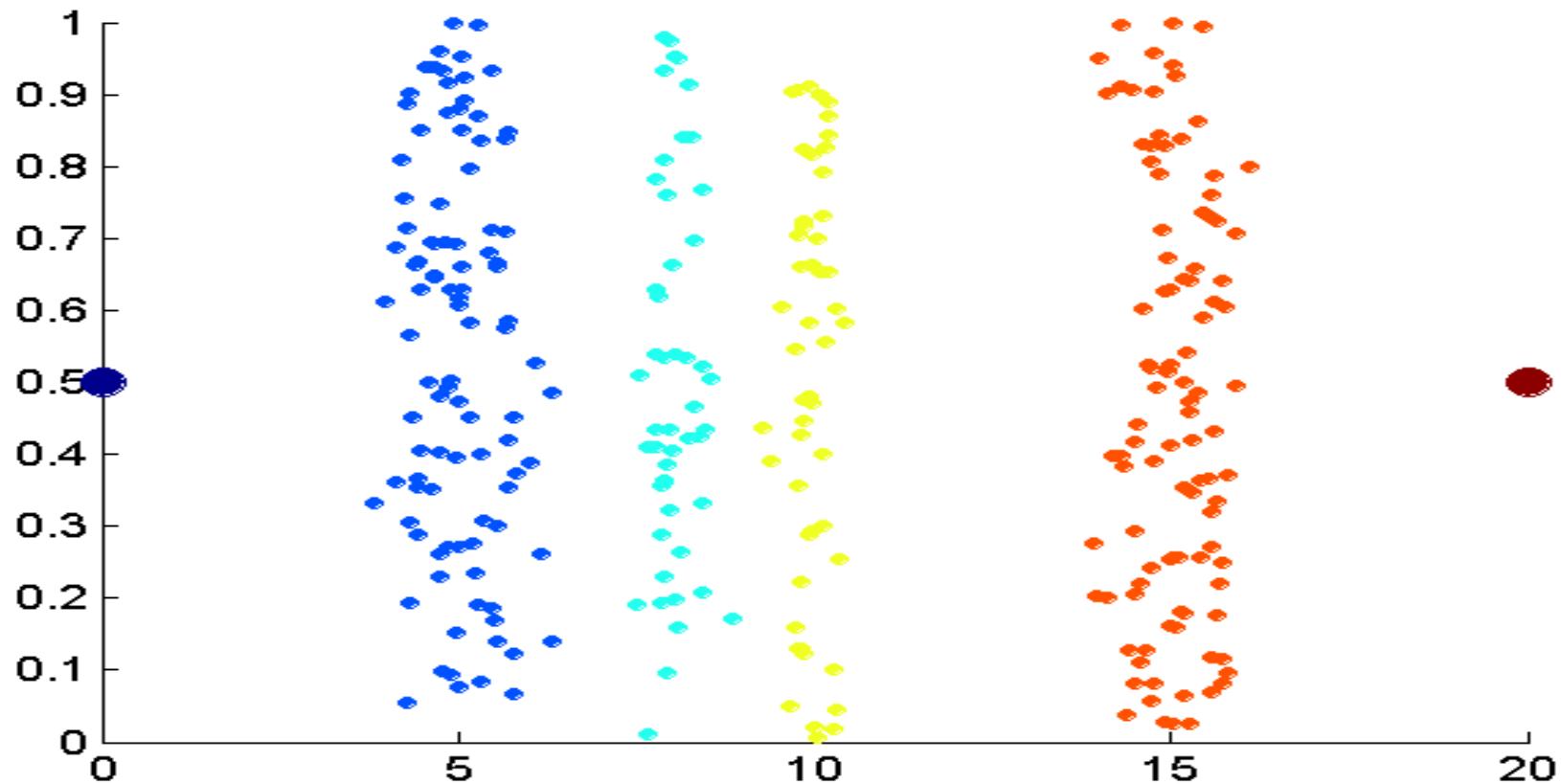
En iyi ayıklaştırmının (best discretization) ne olduğunu nasıl anlayabiliriz?

- **Unsupervised discretization:** veri değerindeki kırılmaları (breaks) bulmak
  - ◆ Example:  
Petal Length



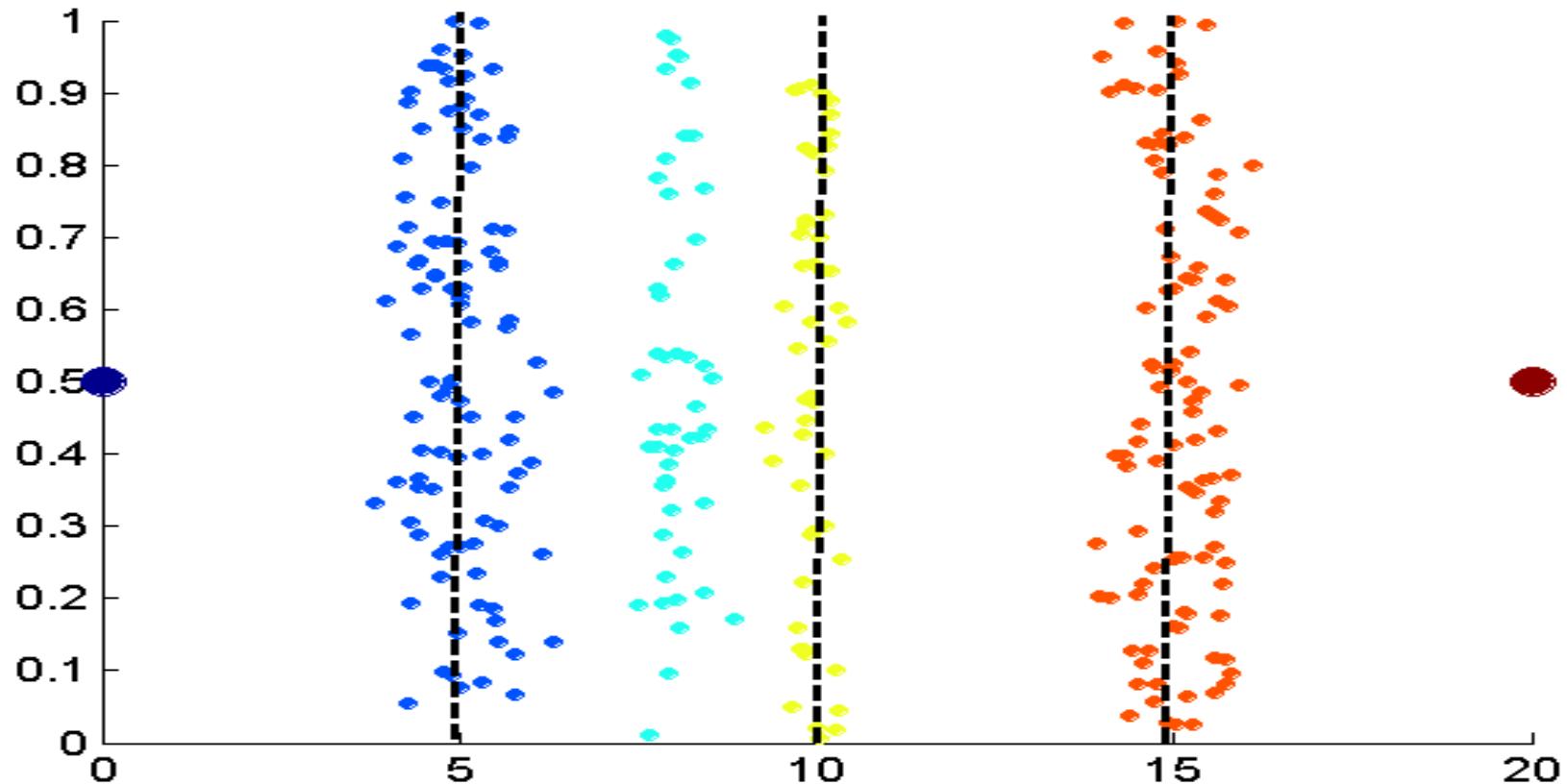
- **Supervised discretization:** Kırılmaları bulmak için sınıf etiketleri kullanmak

# Discretization Without Using Class Labels



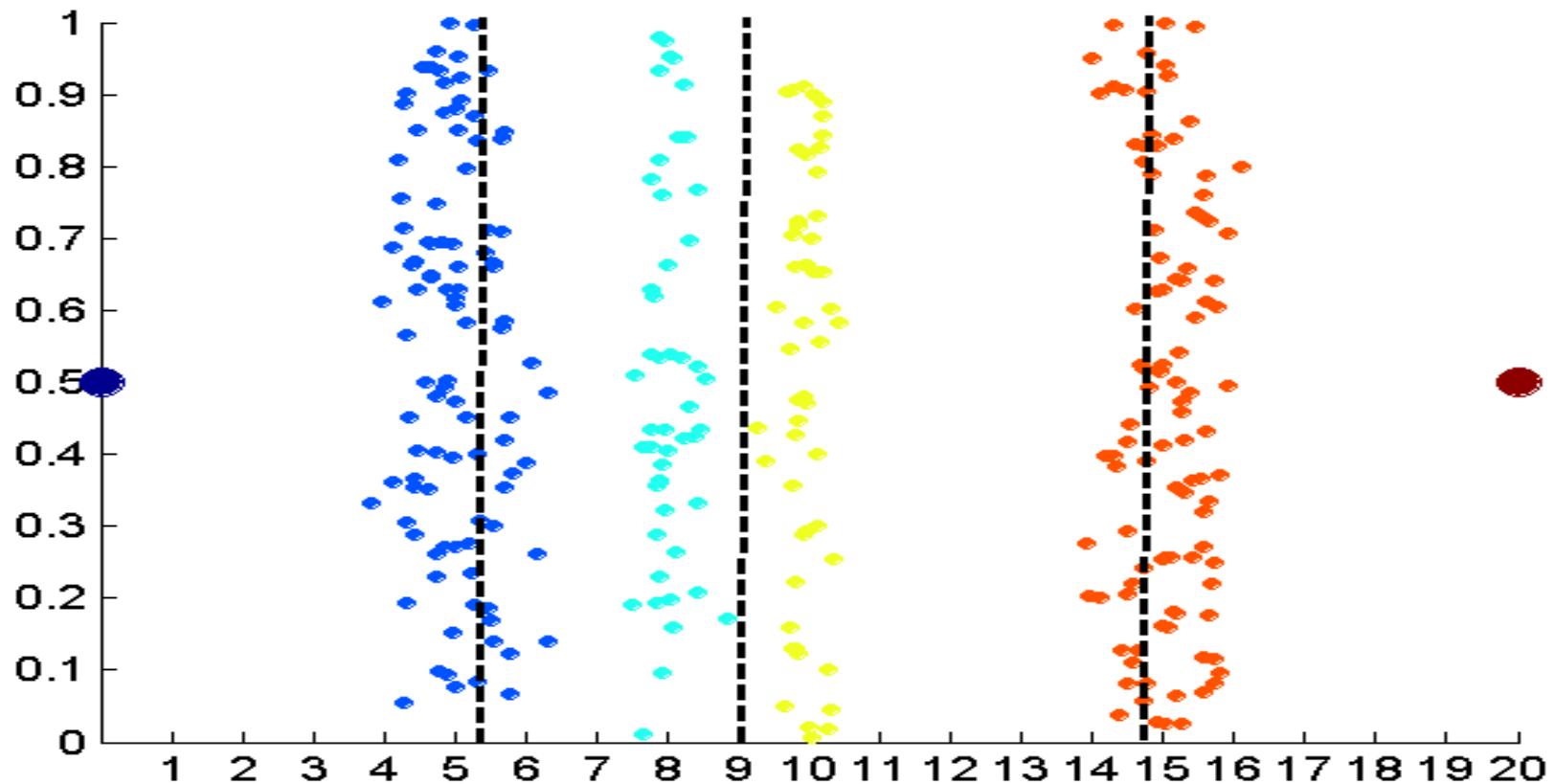
Veriler dört grup noktadan ve iki uç değerden oluşur. Veriler tek boyutludur, ancak çakışmayı azaltmak için rastgele bir y bileşeni eklenir

# Discretization Without Using Class Labels



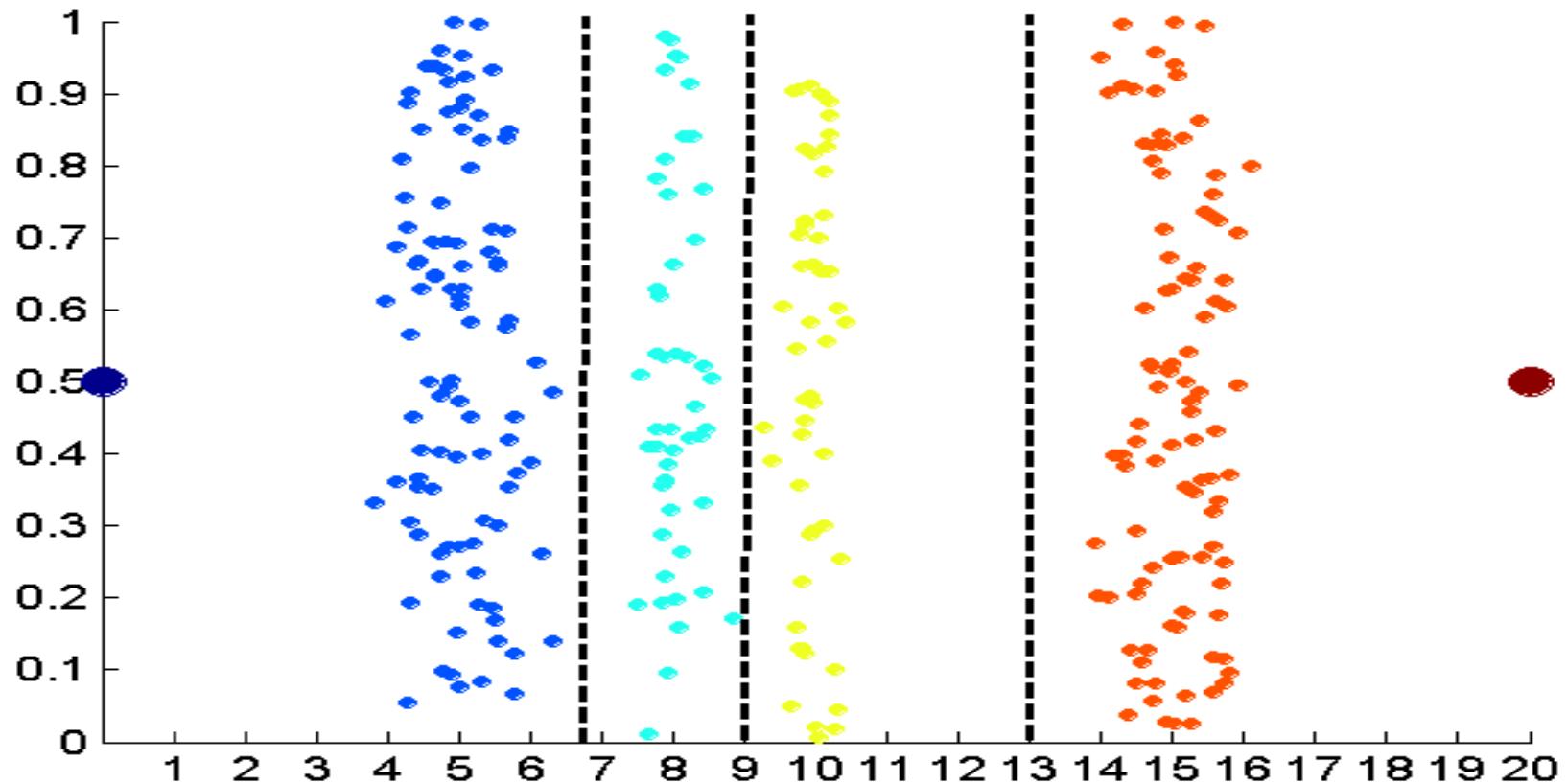
4 değer elde etmek için kullanılan eşit aralık genişliği  
(**Equal interval width**) yaklaşımı.

# Discretization Without Using Class Labels



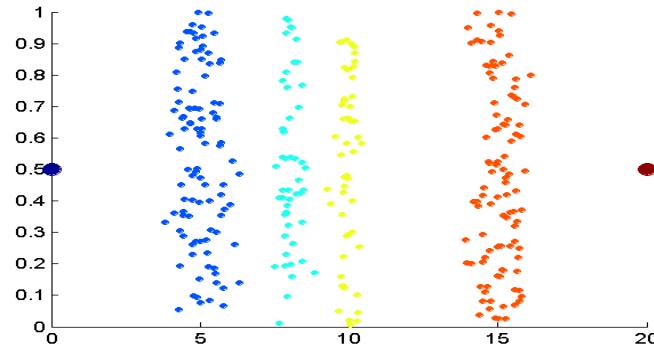
4 değer elde etmek için kullanılan eşit frekans (**Equal frequency**) yaklaşımı

# Discretization Without Using Class Labels

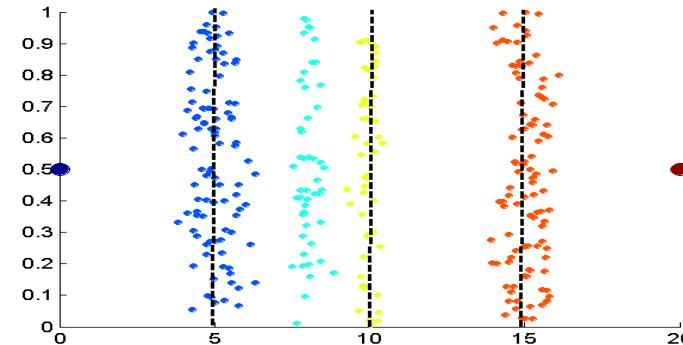


4 değer elde etmek için K-means yaklaşımı

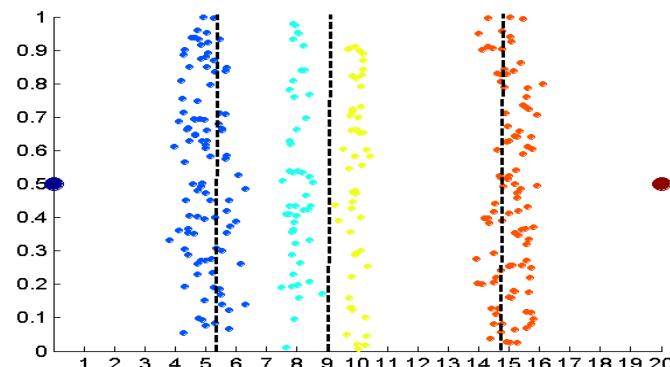
# Discretization Without Using Class Labels



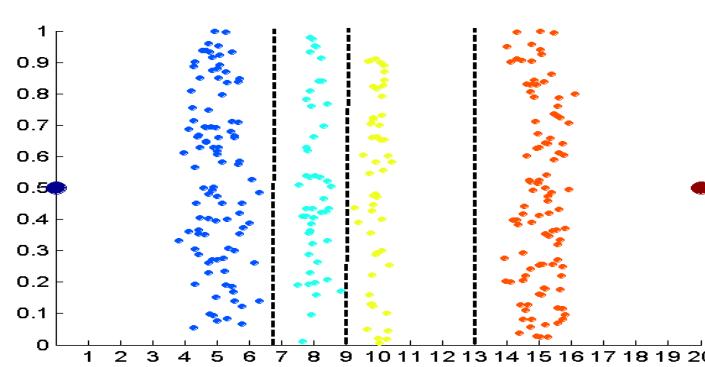
Data



Equal interval width



Equal frequency



K-means

# Binarization

---

- İkilileştirme, sürekli veya kategorik bir özniteligi bir veya daha fazla ikili degiskenle esler.
- Tipik olarak birlikTELik (association) analizi icin kullanilir.
- Genellikle sürekli bir öznitelik once kategorik öznitelige donusturuler ve ardiginden kategorik öznitelik bir dizi ikili öznitelige donusturuler
  - BirlikTELik analizi asimetrik ikili özniteliklere (**asymmetric binary attribute**) ihtiyac duyar
  - Örnekler: göz rengi ve {low, medium, high} şeklinde ölçülen boy özniteligi

# Binarization

**Table 2.5.** Conversion of a categorical attribute to three binary attributes.

Categorical Value	Integer Value	$x_1$	$x_2$	$x_3$
<i>awful</i>	0	0	0	0
<i>poor</i>	1	0	0	1
<i>OK</i>	2	0	1	0
<i>good</i>	3	0	1	1
<i>great</i>	4	1	0	0

*Kategorik  
özniteliğin ikili  
özniteliğe  
dönüştürülmesi*

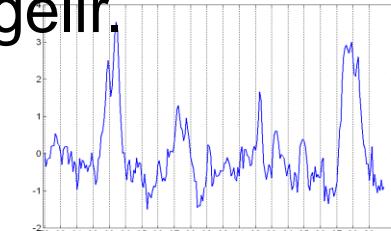
**Table 2.6.** Conversion of a categorical attribute to five asymmetric binary attributes.

Categorical Value	Integer Value	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
<i>awful</i>	0	1	0	0	0	0
<i>poor</i>	1	0	1	0	0	0
<i>OK</i>	2	0	0	1	0	0
<i>good</i>	3	0	0	0	1	0
<i>great</i>	4	0	0	0	0	1

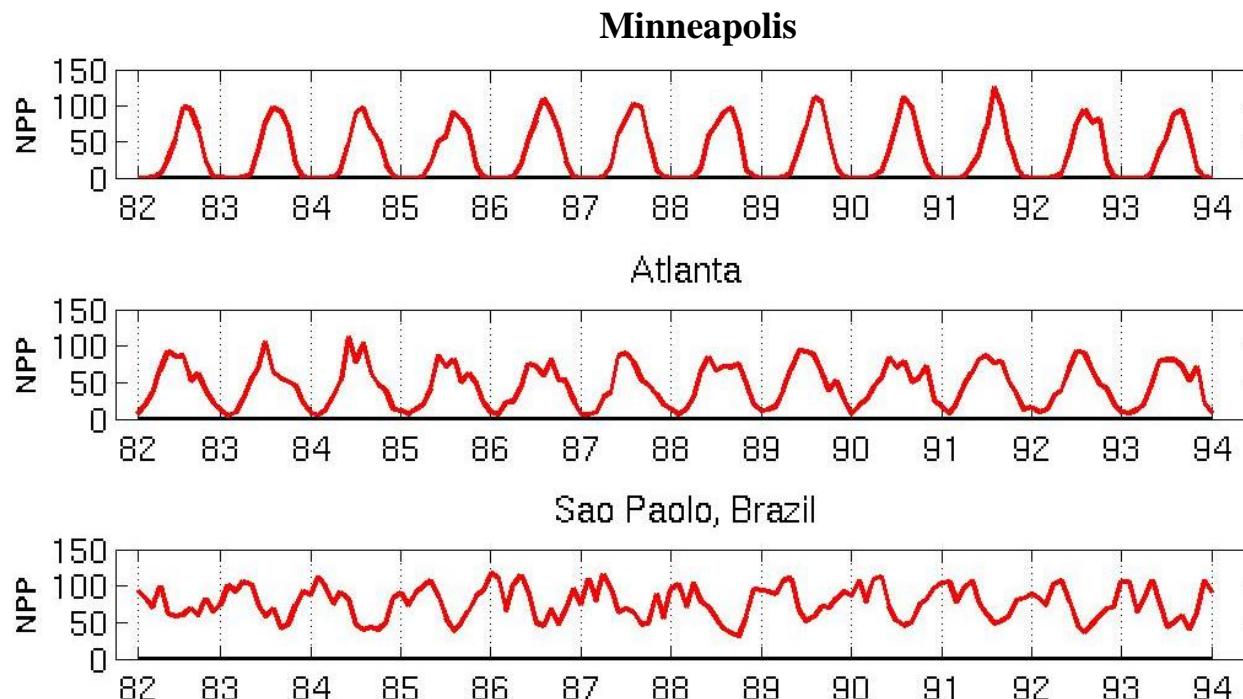
*Kategorik  
özniteliğin 5 tane  
asimetrik ikili  
özniteliğe  
dönüştürülmesi*

# Attribute Transformation

- **Attribute transform:** Belirli bir özniteliğin tüm değer kümesini yeni bir ikame değerler kümesiyle eşleştiren bir fonksiyon, böylece her eski değer yeni değerlerden biriyle tanımlanabilir
  - Basit fonksiyonlar:  $x^k$ ,  $\log(x)$ ,  $e^x$ ,  $|x|$
  - **Normalization**
    - ◆ Ortalama (*mean*), varyans (*variance*), aralık (*range*) açısından özellikler arasındaki farklılıklara uyum sağlamak için çeşitli teknikleri ifade eder.
    - ◆ İstenmeyen, ortak sinyali çıkarın, örn. mevsimsellik
  - **Standardization**, istatistikte ortalamaların çıkarılması ve standart sapmaya bölünmesi anlamına gelir.



# Example: Sample Time Series of Plant Growth

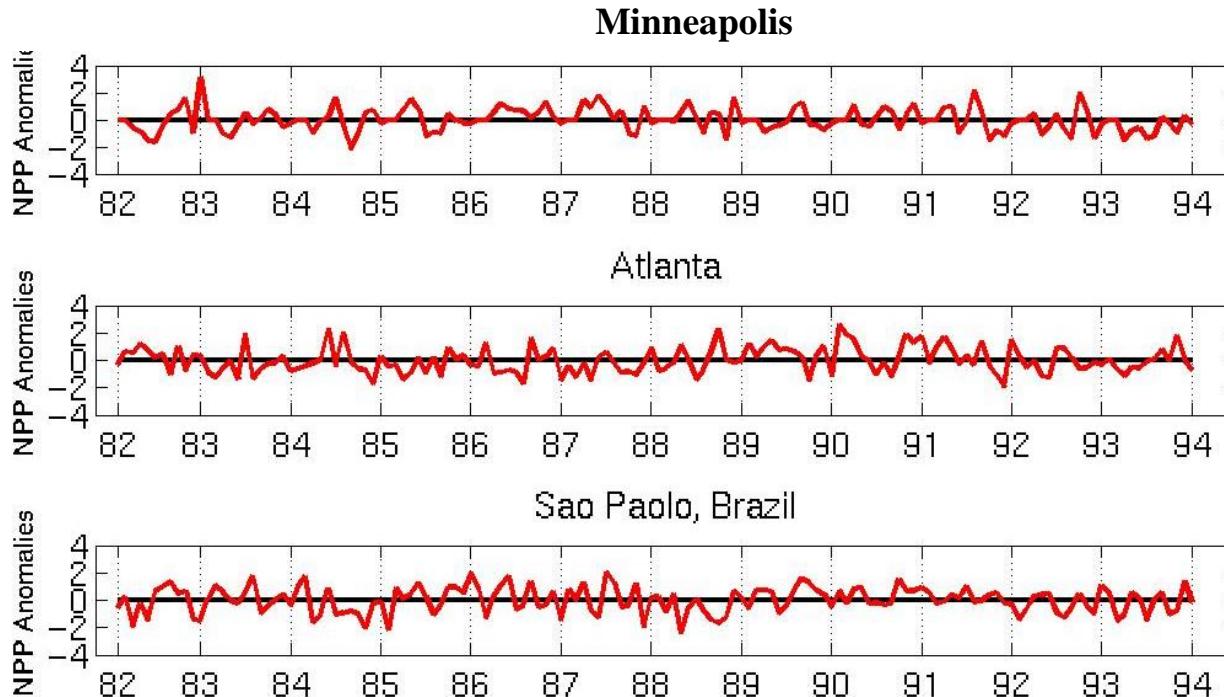


**Net Birincil Üretim  
(Net Primary  
Production -NPP),  
ekosistem  
bilimcileri  
tarafından  
kullanılan bitki  
büyümesinin bir  
ölçüsüdür.**

## Zaman serileri arasındaki korelasyon

	Minneapolis	Atlanta	Sao Paolo
Minneapolis	1.0000	0.7591	-0.7581
Atlanta	0.7591	1.0000	-0.5739
Sao Paolo	-0.7581	-0.5739	1.0000

# Seasonality Accounts for Much Correlation



Korelasyonun  
büyük bölümü  
mevsimsellik  
sebebiyedir

Aylık Z Score  
kullanılarak  
normalize edildi

Aylık ortalamayı  
çıkarın ve aylık  
standart sapmaya  
bölün

## Correlations between time series

	Minneapolis	Atlanta	Sao Paolo
Minneapolis	1.0000	0.0492	0.0906
Atlanta	0.0492	1.0000	-0.0154
Sao Paolo	0.0906	-0.0154	1.0000

# Similarity and Dissimilarity

---

- Similarity (*Benzerlik*)

- İki veri nesnesinin ne kadar benzer olduğunu sayısal ölçüsü.
- Nesneler birbirine daha çok benzediğinde daha yüksektir.
- Genellikle [0,1] aralığına düşer

- Dissimilarity (*Farklılık*)

- İki veri nesnesinin ne kadar farklı olduğunu sayısal ölçüsü
- Nesneler birbirine daha çok benzediğinde daha düşük
- Minimum farklılık genellikle 0'dır
- Üst limit değişebilir

- Yakınlık (*Proximity*), benzerlik veya farklılığı ifade eder

# Similarity/Dissimilarity for Simple Attributes

$p$  and  $q$  are the attribute values for two data objects.

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$	$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$
Ordinal	$d = \frac{ p-q }{n-1}$ (values mapped to integers 0 to $n-1$ , where $n$ is the number of values)	$s = 1 - \frac{ p-q }{n-1}$
Interval or Ratio	$d =  p - q $	$s = -d, s = \frac{1}{1+d}$ or $s = 1 - \frac{d - \min_d}{\max_d - \min_d}$

**Table 5.1.** Similarity and dissimilarity for simple attributes

# Similarity/Dissimilarity transformation examples

---

For the dissimilarity values of 0, 1, 10, 100;

$s = \frac{1}{1+d}$  transformation equation results in similarity values of 1, 0.5, 0.09, 0.01, respectively.

$s = 1 - \frac{d - \min_d}{\max_d - \min_d}$  transformation equation results in similarity values of 1.00, 0.99, 0.00, 0.00, respectively.

$s = e^{-d}$  transformation equation results in similarity values of 1.00, 0.37, 0.00, 0.00, respectively.

# Euclidean Distance

---

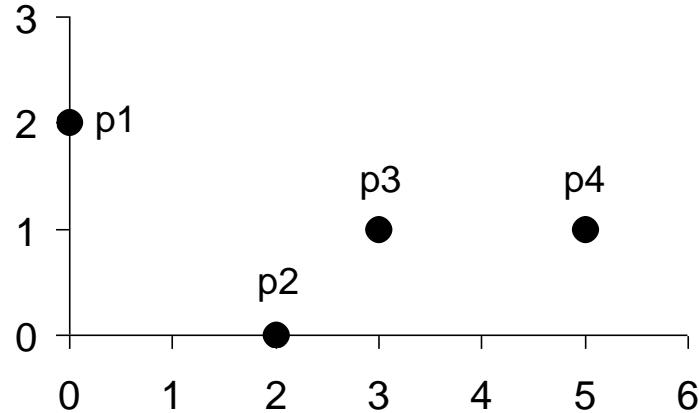
- Euclidean Distance (Öklit Mesfesi)

$$dist = \sqrt{\sum_{k=1}^n (p_k - q_k)^2}$$

Burada n boyut sayısı (öznitelikler) ve  $p_k$  ve  $q_k$  sırasıyla k'inci öznitelikler (bileşenler) veya p ve q veri nesneleridir.

- Ölçekler farklıysa standardizasyon gereklidir.

# Euclidean Distance



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

## Distance Matrix

# Minkowski Distance

---

- Minkowski Distance is a generalization of Euclidean Distance

$$dist = \left( \sum_{k=1}^n |p_k - q_k|^r \right)^{\frac{1}{r}}$$

Where  $r$  is a parameter,  $n$  is the number of dimensions (attributes) and  $p_k$  and  $q_k$  are, respectively, the  $k^{\text{th}}$  attributes (components) or data objects  $p$  and  $q$ .

# Minkowski Distance: Examples

---

- $r = 1$ . City block (Manhattan, taxicab,  $L_1$  norm) distance.
  - Bunun yaygın bir örneği, iki binary vektör arasında farklı olan bitlerin sayısı, Hamming mesafesidir (**Hamming distance**).
- $r = 2$ . Euclidean distance ( $L_2$  norm)
- $r \rightarrow \infty$ . “supremum” ( $L_{\max}$  norm,  $L_{\infty}$  norm) distance.
  - Bu, vektörlerin herhangi bir bileşeni arasındaki maksimum farktır
- Do not confuse  $r$  with  $n$ , i.e., all these distances are defined for all numbers of dimensions.

# Minkowski Distance

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

L $\infty$	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

## Distance Matrix

# Common Properties of a Similarity

---

- Similarities, also have some well known properties.
  1.  $s(p, q) = 1$  (or maximum similarity) only if  $p = q$ .
  2.  $s(p, q) = s(q, p)$  for all  $p$  and  $q$ . (Symmetry)

where  $s(p, q)$  is the similarity between points (data objects),  $p$  and  $q$ .

# Similarity Between Binary Vectors

---

- Common situation is that objects,  $p$  and  $q$ , have only binary attributes
- Compute similarities using the following quantities

$M_{01}$  = the number of attributes where  $p$  was 0 and  $q$  was 1

$M_{10}$  = the number of attributes where  $p$  was 1 and  $q$  was 0

$M_{00}$  = the number of attributes where  $p$  was 0 and  $q$  was 0

$M_{11}$  = the number of attributes where  $p$  was 1 and  $q$  was 1

- Simple Matching and Jaccard Coefficients

SMC = number of matches / number of attributes

$$= (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$$

J = number of 11 matches / number of not-both-zero attributes values

$$= (M_{11}) / (M_{01} + M_{10} + M_{11})$$

# SMC versus Jaccard: Example

---

$p = 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$

$q = 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1$

$M_{01} = 2$  (the number of attributes where p was 0 and q was 1)

$M_{10} = 1$  (the number of attributes where p was 1 and q was 0)

$M_{00} = 7$  (the number of attributes where p was 0 and q was 0)

$M_{11} = 0$  (the number of attributes where p was 1 and q was 1)

$$SMC = (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00}) = (0+7) / (2+1+0+7) = 0.7$$

$$J = (M_{11}) / (M_{01} + M_{10} + M_{11}) = 0 / (2 + 1 + 0) = 0$$

# Cosine Similarity

---

- If  $\mathbf{d}_1$  and  $\mathbf{d}_2$  are two document vectors, then

$$\cos(\mathbf{d}_1, \mathbf{d}_2) = \langle \mathbf{d}_1, \mathbf{d}_2 \rangle / \|\mathbf{d}_1\| \|\mathbf{d}_2\|,$$

where  $\langle \mathbf{d}_1, \mathbf{d}_2 \rangle$  indicates inner product or vector dot product of vectors,  $\mathbf{d}_1$  and  $\mathbf{d}_2$ , and  $\|\mathbf{d}\|$  is the length of vector  $\mathbf{d}$ .

- Example:

$$\mathbf{d}_1 = 3 \ 2 \ 0 \ 5 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0$$

$$\mathbf{d}_2 = 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 2$$

$$\langle \mathbf{d}_1, \mathbf{d}_2 \rangle = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$$

$$\|\mathbf{d}_1\| = (3^2 + 2^2 + 0^2 + 5^2 + 0^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2)^{0.5} = (42)^{0.5} = 6.481$$

$$\|\mathbf{d}_2\| = (1^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 1^2 + 0^2 + 2^2)^{0.5} = (6)^{0.5} = 2.449$$

$$\cos(\mathbf{d}_1, \mathbf{d}_2) = 0.3150$$

# Extended Jaccard Coefficient (Tanimoto)

---

- Variation of Jaccard for continuous or count attributes
  - Reduces to Jaccard for binary attributes

$$EJ(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - \mathbf{x} \cdot \mathbf{y}}$$

# Correlation

---

- Korelasyon, nesneler arasındaki doğrusal ilişkiye ölçer
- Korelasyonu hesaplamak için, veri nesnelerini, p ve q'yu standartlaştıryoruz ve sonra «dot product» alıyoruz

$$p'_k = (p_k - \text{mean}(p)) / \text{std}(p)$$

$$q'_k = (q_k - \text{mean}(q)) / \text{std}(q)$$

$$\text{correlation}(p, q) = p' \bullet q'$$

# Correlation measures the linear relationship between objects

$$\text{corr}(\mathbf{x}, \mathbf{y}) = \frac{\text{covariance}(\mathbf{x}, \mathbf{y})}{\text{standard\_deviation}(\mathbf{x}) * \text{standard\_deviation}(\mathbf{y})} = \frac{s_{xy}}{s_x s_y}, \quad (2.11)$$

where we are using the following standard statistical notation and definitions

$$\text{covariance}(\mathbf{x}, \mathbf{y}) = s_{xy} = \frac{1}{n-1} \sum_{k=1}^n (x_k - \bar{x})(y_k - \bar{y}) \quad (2.12)$$

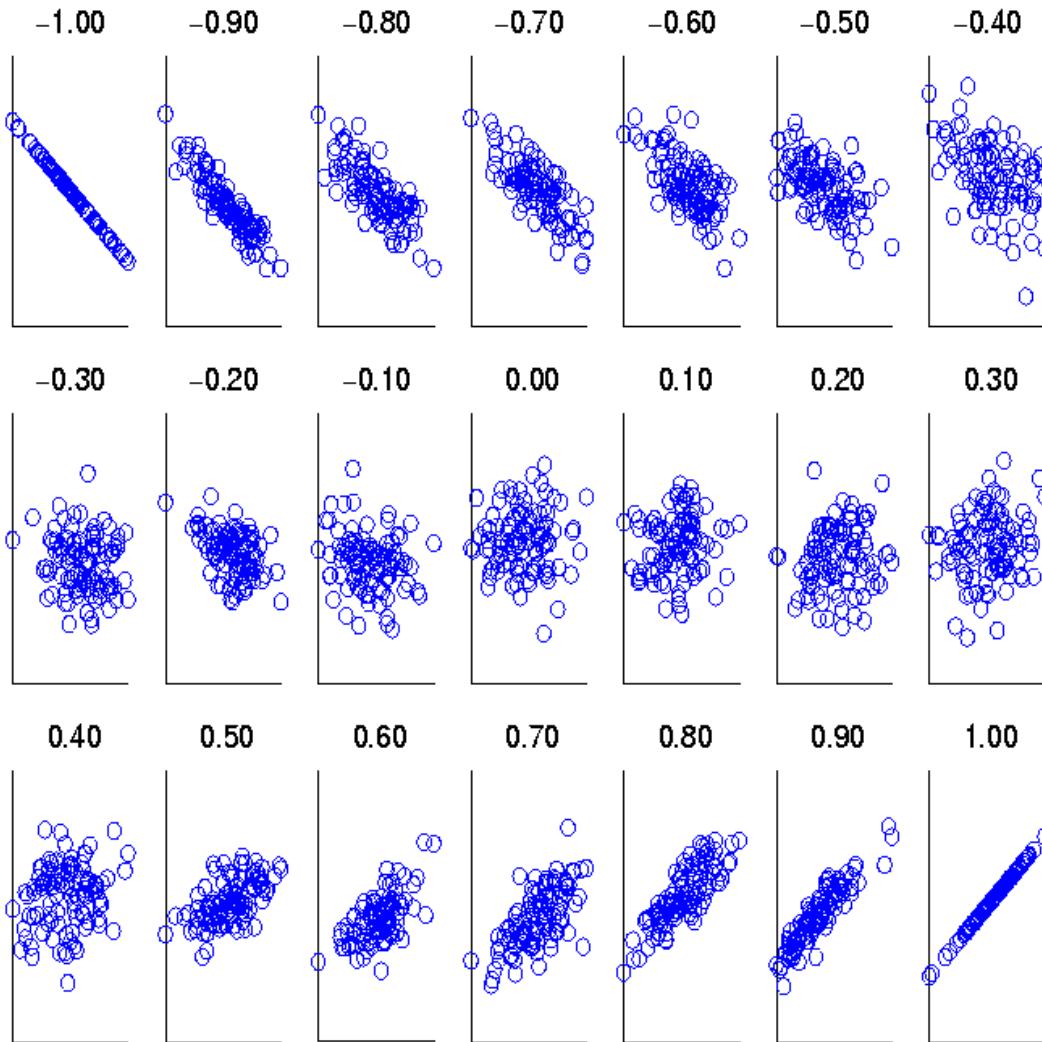
$$\text{standard\_deviation}(\mathbf{x}) = s_x = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (x_k - \bar{x})^2}$$

$$\text{standard\_deviation}(\mathbf{y}) = s_y = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (y_k - \bar{y})^2}$$

$$\bar{x} = \frac{1}{n} \sum_{k=1}^n x_k \text{ is the mean of } \mathbf{x}$$

$$\bar{y} = \frac{1}{n} \sum_{k=1}^n y_k \text{ is the mean of } \mathbf{y}$$

# Visually Evaluating Correlation



**Scatter plots  
showing the  
similarity from  
-1 to 1.**

# Drawback of Correlation

---

- $\mathbf{x} = (-3, -2, -1, 0, 1, 2, 3)$
- $\mathbf{y} = (9, 4, 1, 0, 1, 4, 9)$

$$y_i = x_i^2$$


- $\text{mean}(\mathbf{x}) = 0, \text{mean}(\mathbf{y}) = 4$
- $\text{std}(\mathbf{x}) = 2.16, \text{std}(\mathbf{y}) = 3.74$
- $\text{corr} = (-3)(5)+(-2)(0)+(-1)(-3)+(0)(-4)+(1)(-3)+(2)(0)+3(5) / ( 6 * 2.16 * 3.74 )$   
= 0      If the **correlation** is 0, then there is **no linear relationship** between the attributes of the two data objects. However, **non-linear relationships** may still exist as in this example.

# General Approach for Combining Similarities

---

- Sometimes attributes are of many different types, but an overall similarity is needed.

1. For the  $k^{th}$  attribute, compute a similarity,  $s_k$ , in the range  $[0, 1]$ .
2. Define an indicator variable,  $\delta_k$ , for the  $k_{th}$  attribute as follows:

$$\delta_k = \begin{cases} 0 & \text{if the } k^{th} \text{ attribute is a binary asymmetric attribute and both objects have} \\ & \text{a value of 0, or if one of the objects has a missing values for the } k^{th} \text{ attribute} \\ 1 & \text{otherwise} \end{cases}$$

3. Compute the overall similarity between the two objects using the following formula:

$$similarity(p, q) = \frac{\sum_{k=1}^n \delta_k s_k}{\sum_{k=1}^n \delta_k}$$

# Using Weights to Combine Similarities

---

- May not want to treat all attributes the same.
  - Use weights  $w_k$  which are between 0 and 1 and sum to 1.

$$\text{similarity}(p, q) = \frac{\sum_{k=1}^n w_k \delta_k s_k}{\sum_{k=1}^n \delta_k}$$

$$\text{distance}(p, q) = \left( \sum_{k=1}^n w_k |p_k - q_k|^r \right)^{1/r}$$

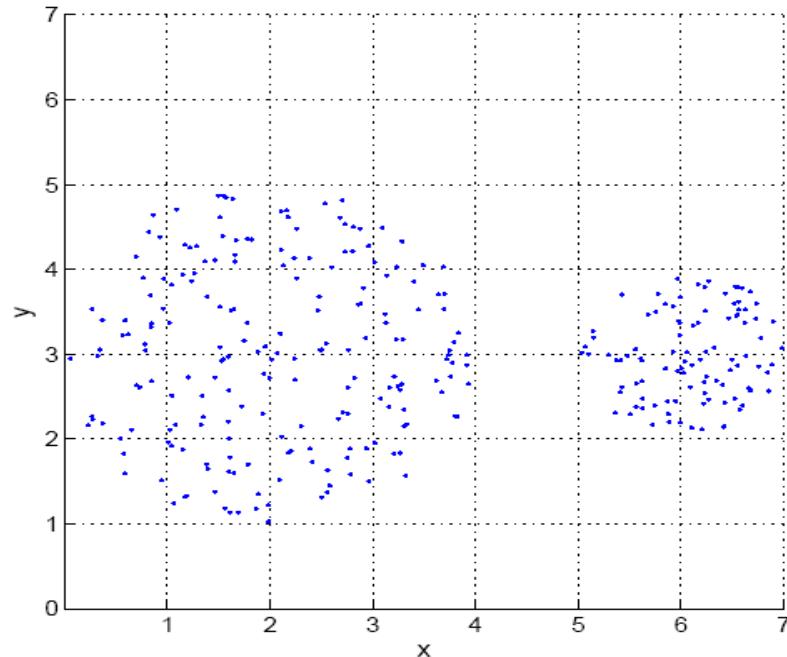
# Density

---

- Belirli bir alanda veri nesnelerinin birbirine yakın olma derecesini ölçer
- Yoğunluk (**density**) kavramı yakınlık kavramı ile yakından ilgilidir.
- Yoğunluk kavramı tipik olarak kümeleme ve anormallik tespiti için kullanılır
- Examples:
  - Euclidean density
    - ◆ Euclidean density = number of points per unit volume
  - Probability density
    - ◆ Estimate what the distribution of the data looks like
  - Graph-based density
    - ◆ Connectivity

# Euclidean Density: Grid-based Approach

- En basit yaklaşım, bölgeyi belirli sayıda eşit hacimli dikdörtgen hücrelere bölmek ve yoğunluğu hücrenin içерdiği nokta sayısı olarak tanımlamaktır.



**Grid-based density.**

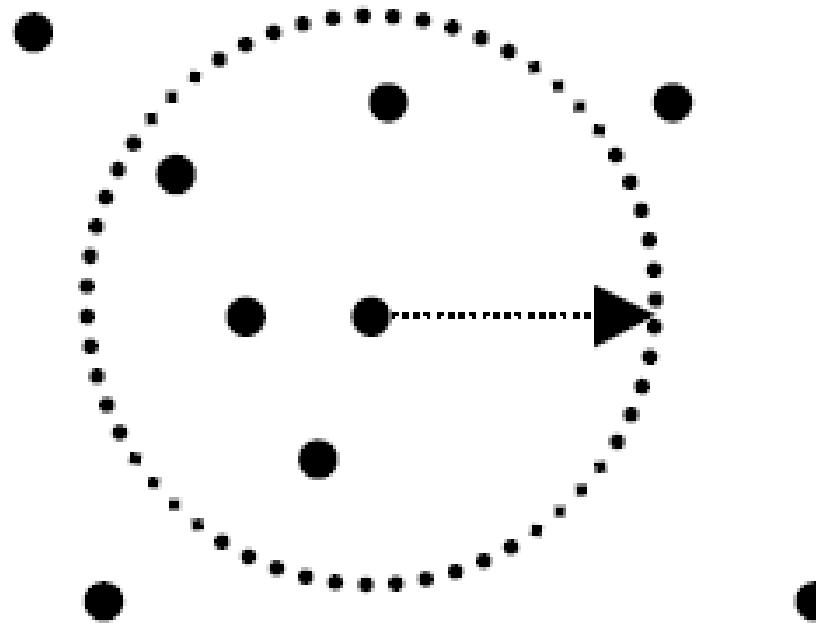
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
4	17	18	6	0	0	0	0
14	14	13	13	0	18	27	
11	18	10	21	0	24	31	
3	20	14	4	0	0	0	0
0	0	0	0	0	0	0	0

**Counts for each cell.**

# Euclidean Density: Center-Based

---

- Öklid yoğunluğu, belirli bir yarıçapı içindeki noktaların sayısıdır.



**Illustration of center-based density.**

# Data Mining: Exploring Data

---

---

## Lecture Notes for Chapter 3

Introduction to Data Mining

by

Tan, Steinbach, Kumar

# What is data exploration?

---

Özelliklerini daha iyi anlamak için veriler  
üzerinde ön araştırma yapma işi

- «Data exploration» temel motivasyonlar :
  - Ön işleme veya analiz için doğru aracı seçmeye yardımcı olma
  - İnsanların kalıpları/örüntüleri tanıma yeteneklerinden yararlanma
    - ◆ İnsanlar veri analizi araçları tarafından yakalanmayan kalıpları tanıyalabilir
- Related to the area of Exploratory Data Analysis (EDA)
  - Created by statistician John Tukey
  - Seminal book is Exploratory Data Analysis by Tukey
  - A nice online introduction can be found in Chapter 1 of the NIST Engineering Statistics Handbook

<http://www.itl.nist.gov/div898/handbook/index.htm>

# Techniques Used In Data Exploration

---

- EDA'da, orijinal olarak Tukey tarafından tanımlandığı gibi
  - Odak noktası görselleştirme (*visualization*) idi
  - Kümeleme ve anormallik tespiti keşif teknikleri (*explatory techniques*) olarak görüldü
  - Veri madenciliğinde, kümeleme ve anormallik tespiti başlıca ilgi alanlarıdır ve sadece keşif amaçlı olarak düşünülmez
- In our discussion of data exploration, we focus on
  - Summary statistics
  - Visualization
  - Online Analytical Processing (OLAP)

# Iris Sample Data Set

- Many of the exploratory data techniques are illustrated with the Iris Plant data set.
  - Can be obtained from the UCI Machine Learning Repository  
<http://www.ics.uci.edu/~mlearn/MLRepository.html>
  - From the statistician Douglas Fisher
  - Three flower types (classes):
    - Setosa
    - Virginica
    - Versicolour
  - Four (non-class) attributes
    - Sepal width and length
    - Petal width and length



Virginica. Robert H. Mohlenbrock. USDA NRCS. 1995. Northeast wetland flora: Field office guide to plant species. Northeast National Technical Center, Chester, PA. Courtesy of USDA NRCS Wetland Science Institute.

# Summary Statistics

---

- **Özet istatistikler**, verilerin özelliklerini özetleyen sayılardır
  - Özet özellikler arasında sıklık (**frequency**), konum (**location**) ve yayılma (**spread**) bulunur
    - ◆ Examples: location - **mean**  
spread - **standard deviation**
  - Özet istatistiklerin çoğu, veriler üzerinden **tek bir geçişte** (**in a single pass through the data**) hesaplanabilir.

# Frequency and Mode

---

- Bir öznitelik değerinin sıklığı, değerin veri kümesinde var olma yüzdesidir.
  - Örneğin, "cinsiyet" özniteliği ve temsili bir insan popülasyonu verildiğinde, cinsiyet "kadın" yaklaşık %50 oranında ortaya çıkar.
- Bir özniteliğin modu (mode of an attribute ) **en sık görülen öznitelik değeridir**
- Sıklık (frequency) ve mod kavramları tipik olarak kategorik verilerle kullanılır

# Percentiles

---

- Sürekli veriler (***continuous data***) için, yüzdelik (***percentile***) kavramı daha kullanışlıdır.

Sıralı veya sürekli bir  $x$  özniteliği ve 0 ile 100 arasında bir  $p$  sayısı verildiğinde, ***p.*** yüzdelik dilim  $x_p$ ,  $x$ 'in gözlemlenen değerlerinin  $\%p$  'inden küçük olacak şekilde bir  $x$  değeridir.

- Örneğin, 50. yüzdelik dilim  $x_{50\%}$ ,  $x$ 'in tüm değerlerinin %50'sinin ondan daha küçük olacağı değerdir.

# Measures of Location: Mean and Median

---

- Ortalama (*mean*), bir nokta kümelerinin konumunun en yaygın ölçüsüdür.
- Bununla birlikte, ortalama (*mean*), **uç değerlere (*outliers*) karşı çok hassastır**.
- Bu nedenle, *medyan (median)* veya kırpılmış ortalama da yaygın olarak kullanılır.

$$\text{mean}(x) = \bar{x} = \frac{1}{m} \sum_{i=1}^m x_i$$

$$\text{median}(x) = \begin{cases} x_{(r+1)} & \text{if } m \text{ is odd, i.e., } m = 2r + 1 \\ \frac{1}{2}(x_{(r)} + x_{(r+1)}) & \text{if } m \text{ is even, i.e., } m = 2r \end{cases}$$

# Measures of Location: Mean and Median

---

- **Trimmed mean** (kırılpılmış ortalama):
  - 0 ile 100 arasında bir yüzde  $p$  belirlenir, verilerin **üst** ve **alt  $\%(p / 2)$**  'si atılır ve daha sonra ortalama normal şekilde hesaplanır.
  - Örnek
    - $\{1,2,3,4,5,90\}$  değerler kümesini düşününüz.
    - What is the **mean**, **median** and the **trimmed mean** with  $p=40\%$ ?
  - Answer
    - **mean**=17.5
    - **median**=3.5
    - **trimmed mean**(40%)=3.5

# Measures of Spread: Range and Variance

- Range, maksimum ve minimum arasındaki farktır.
- Varyans veya standart sapma, bir nokta kümelerinin yayılmasının (spread) en yaygın ölçüsüdür.

$$\text{variance}(x) = s_x^2 = \frac{1}{m-1} \sum_{i=1}^m (x_i - \bar{x})^2$$

- Fakat, bu da üç değerlere duyarlıdır, bu nedenle sıkılıkla başka ölçüler kullanılır.

absolute average deviation (AAD)

$$AAD(x) = \frac{1}{m} \sum_{i=1}^m |x_i - \bar{x}|$$

median absolute deviation (MAD)

$$MAD(x) = \text{median}\left(\{|x_1 - \bar{x}|, \dots, |x_m - \bar{x}|\}\right)$$

interquartile range (IQR)

$$\text{interquartile range}(x) = x_{75\%} - x_{25\%}$$

# Visualization

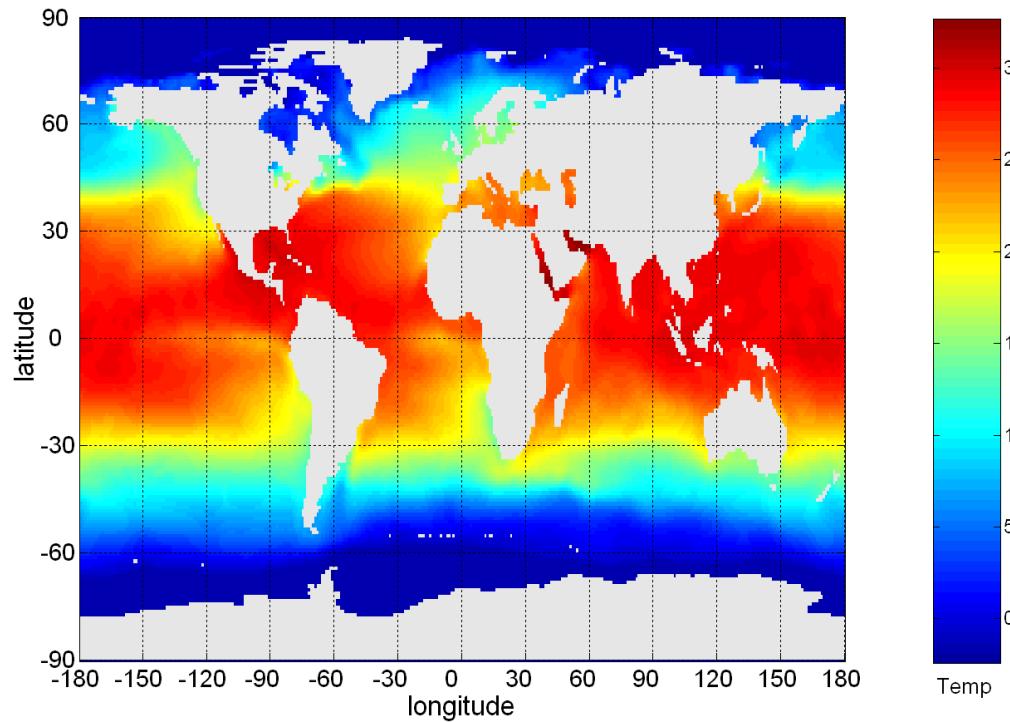
---

Görselleştirme (*Visualization*) verilerin karakteristiklerinin ve veri öğeleri veya öznitelikler arasındaki ilişkilerin analiz edilebilmesi veya raporlanabilmesi için verilerin **görsel veya tablo biçiminde bir biçimde dönüştürülmüş**dir.

- Verilerin görselleştirilmesi, veri keşfi(*data exploration*) için **en güçlü ve çekici tekniklerden** biridir.
  - **İnsanlar** görsel olarak sunulan büyük miktarda bilgiyi analiz etme konusunda **gelişmiş bir beceriye** sahiptir.
  - Genel kalıpları ve eğilimleri tespit edebilir
  - Uç değerleri ve alışılmadık kalıpları tespit edebilir

# Example: Sea Surface Temperature

- Aşağıda, Temmuz 1982 için Deniz Yüzeyi Sıcaklığı (SST) gösterilmektedir.
  - On binlerce veri noktası tek bir şekilde özetlenmiştir



# Representation

---

- Bilginin görsel bir formatla eşleştirilmesi
- Veri nesneleri, öznitelikleri ve veri nesneleri arasındaki ilişkiler, noktalar, çizgiler, şekiller ve renkler gibi grafiksel öğelere çevrilir.
- Örnek:
  - Nesneler genellikle **noktalar** olarak temsil edilir
  - Öznitelik değerleri, noktaların **konumu** veya noktaların **özellikleri**, örn. **renk**, **boyut** ve **şekil** olarak gösterilebilir.
  - **Konum** bilgisi kullanılırsa, noktaların **ilişkileri**, yani gruplar oluşturup oluşturmadıkları veya bir noktanın üç değer olup olmadığı **kolayca algılanır**.

# Arrangement

- Görsel öğelerin bir ekran içinde yerleşimidir
- Verileri anlamanın ne kadar kolay olduğu konusunda **büyük bir fark yaratabilir**
- Örnek:

	1	2	3	4	5	6
1	0	1	0	1	1	0
2	1	0	1	0	0	1
3	0	1	0	1	1	0
4	1	0	1	0	0	1
5	0	1	0	1	1	0
6	1	0	1	0	0	1
7	0	1	0	1	1	0
8	1	0	1	0	0	1
9	0	1	0	1	1	0

	6	1	3	2	5	4
4	1	1	1	0	0	0
2	1	1	1	0	0	0
6	1	1	1	0	0	0
8	1	1	1	0	0	0
5	0	0	0	1	1	1
3	0	0	0	1	1	1
9	0	0	0	1	1	1
1	0	0	0	1	1	1
7	0	0	0	1	1	1

Satırların ve sütunların ilişkilerinin belirgin hale getirildiği altı tane ikili niteliğe (sütun) sahip dokuz nesneden (satır) oluşan bir tablo.

# Selection

---

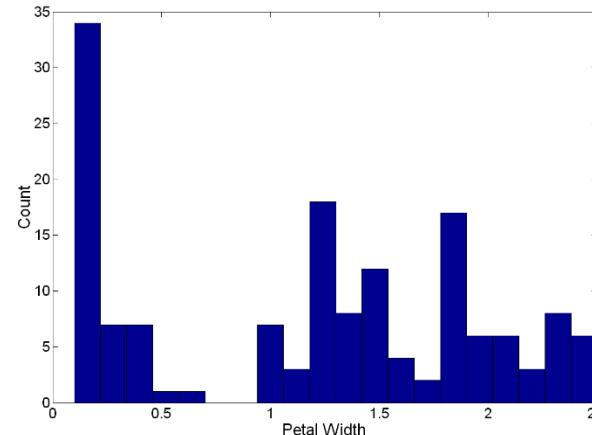
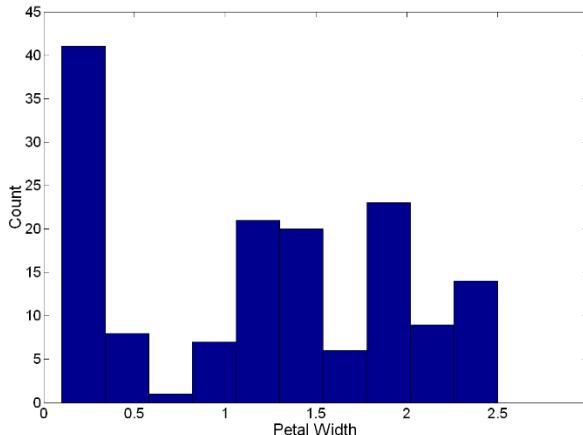
- Belirli nesnelerin ve niteliklerin ortadan kaldırılması veya vurgulanmaması
- Seçim (*selection*), özniteliklerin bir alt kümесinin seçilmesini içerebilir
  - Boyut azaltma (*Dimensionality reduction*), genellikle boyutların sayısını iki veya üçe düşürmek için kullanılır
  - Alternatif olarak, öznitelik çiftleri (*pairs of attributes*) düşünülebilir
- Seçim, ayrıca nesnelerin bir alt kümесini (*a subset of objects*) seçmeyi de içerebilir
  - Ekranın bir bölgesi yalnızca belirli sayıda nokta gösterebilir
  - Örnekleme yapılabilir, ancak seyrek alanlardaki noktalar korunmak istenir

# Visualization Techniques: Histograms

- Histogram

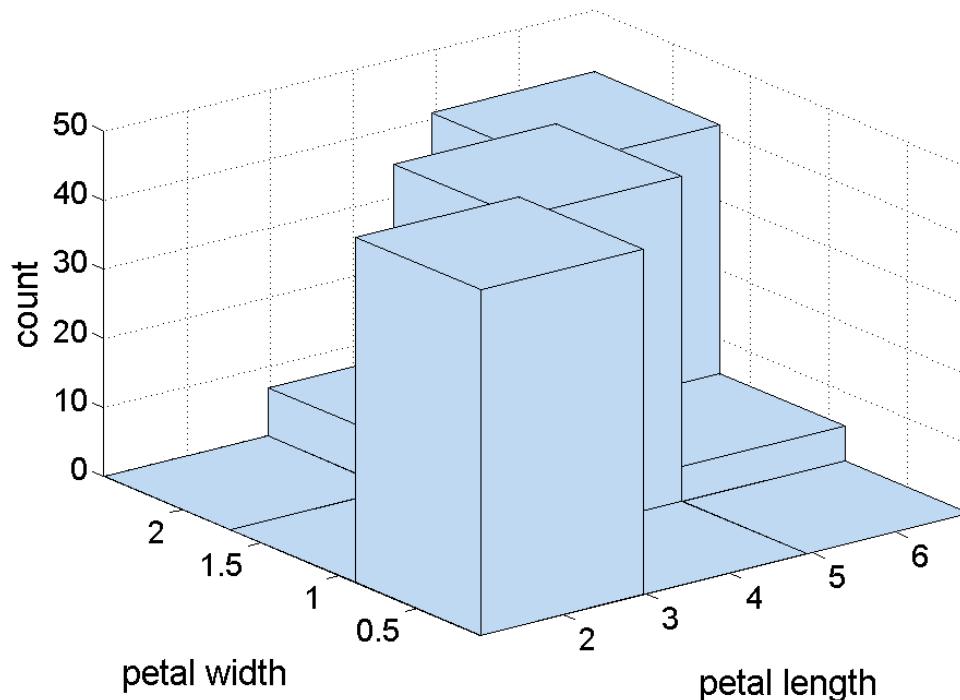
- Genellikle **tek bir değişkenin değerlerinin dağılımını** gösterir.
- **Değerler bölmelere (bins) dağıtıılır** ve her bölmedeki nesnelerin sayısının çubuk grafiği gösterilir.
- Her çubuğun yüksekliği nesnelerin sayısını gösterir
- Histogramın şekli, bölme sayısına bağlıdır

- Example: Petal Width (10 and 20 bins, respectively)



# Two-Dimensional Histograms

- İki özniteliğin değerlerinin ortak dağılımını (*joint distribution*) gösterir
- Example: petal width and petal length
  - What does this tell us?



İki boyutlu histogramlar, **iki özniteliğin değerlerinin birlikte nasıl oluştuğuna ilişkin ilginç gerçekleri keşfetmek için kullanılabılırken**, görsel olarak daha karmaşıktır.

Çiçeklerin çoğu sadece üç bölmeye düşüyor—köşegen boyunca olanlar.

Tek boyutlu dağılımlara bakarak bunu görmek mümkün değil.

# Pie Chart

- **Pie Chart (Pasta Grafik)**

- histograma benzer, ancak **tipik olarak** nispeten az sayıda değere sahip **kategorik özelliklerle** kullanılır.
- göreceli frekansı belirtmek için dairenin göreceli alanını kullanır.
- teknik yaynlarda daha az sıkılıkla kullanılır çünkü göreceli alanların boyutunun değerlendirilmesi zor olabilir

Her üç çiçek türünün de frekansı  
(sıklığı) aynı

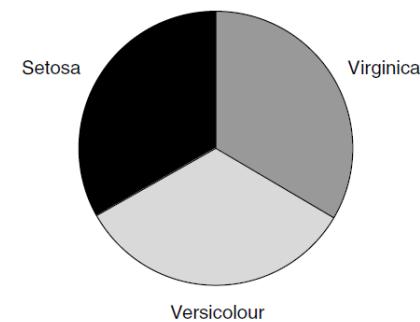
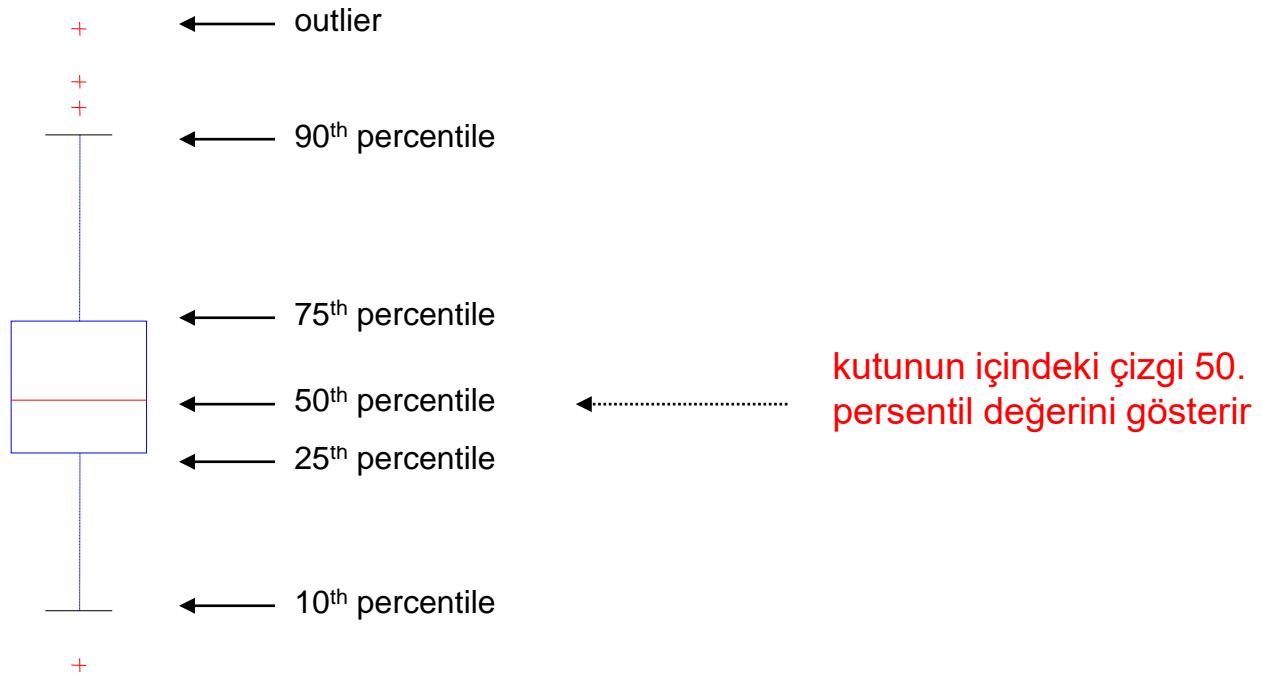


Figure 3.13. Distribution of the types of Iris flowers.

# Visualization Techniques: Box Plots

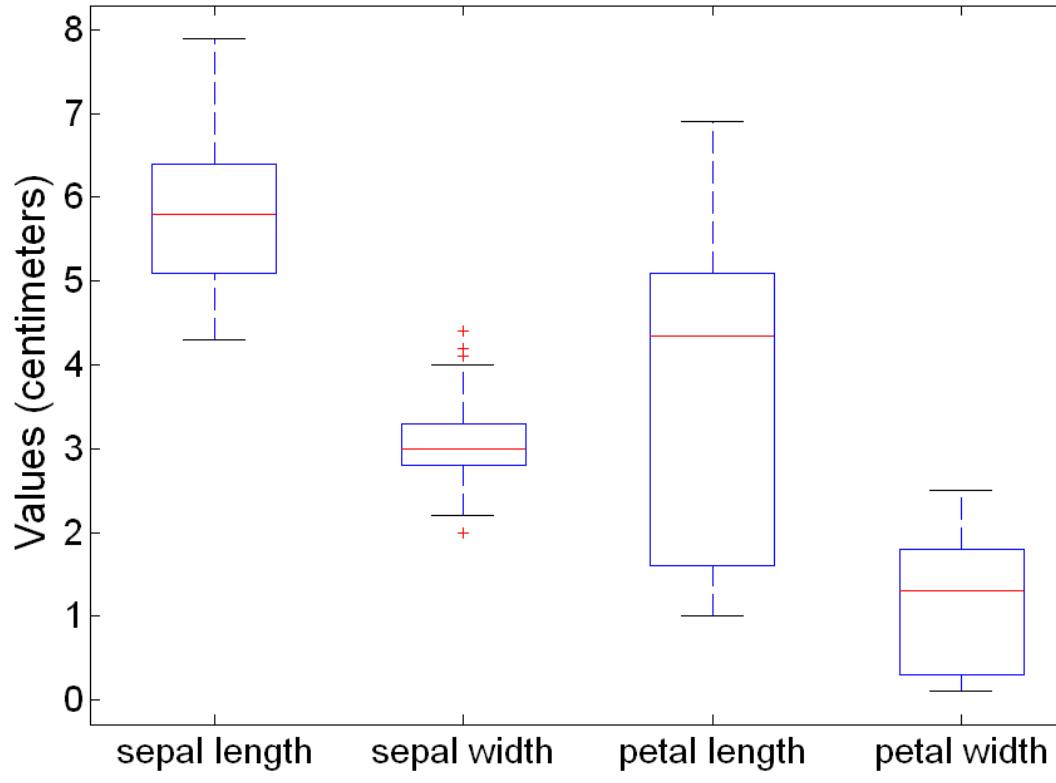
## ● Box Plots

- J. Tukey tarafından icat edilmiştir
- Veri dağılımını göstermenin başka bir yolu
- Aşağıdaki şekilde bir kutu grafiğinin temel bölümünü göstermektedir



# Example of Box Plots

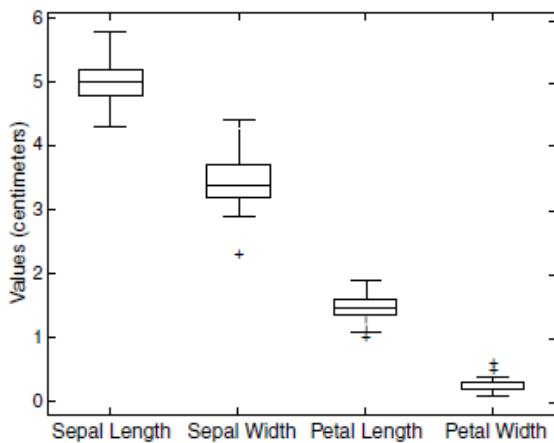
- Öznitelikleri karşılaştırmak için kutu grafikleri kullanılabilir



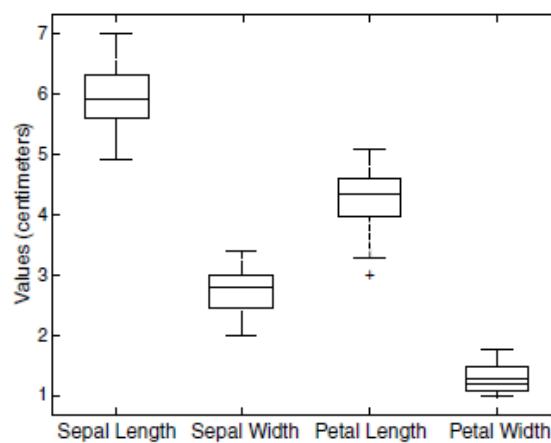
Box plot for Iris attributes

# Example of Box Plots

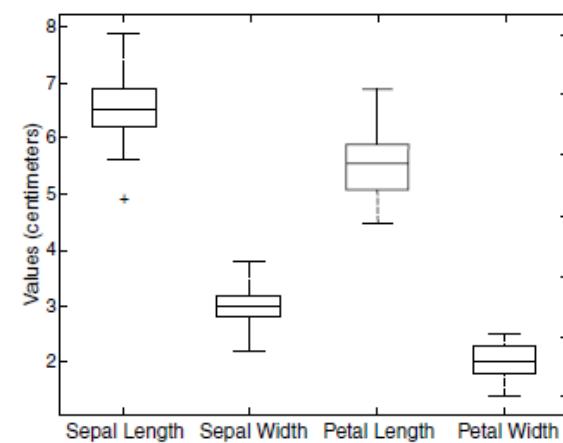
- Kutu grafikleri, özniteliklerin farklı nesne sınıfları arasında nasıl değiştiğini karşılaştırmak için de kullanılabilir.



(a) Setosa.



(b) Versicolour.



(c) Virginica.

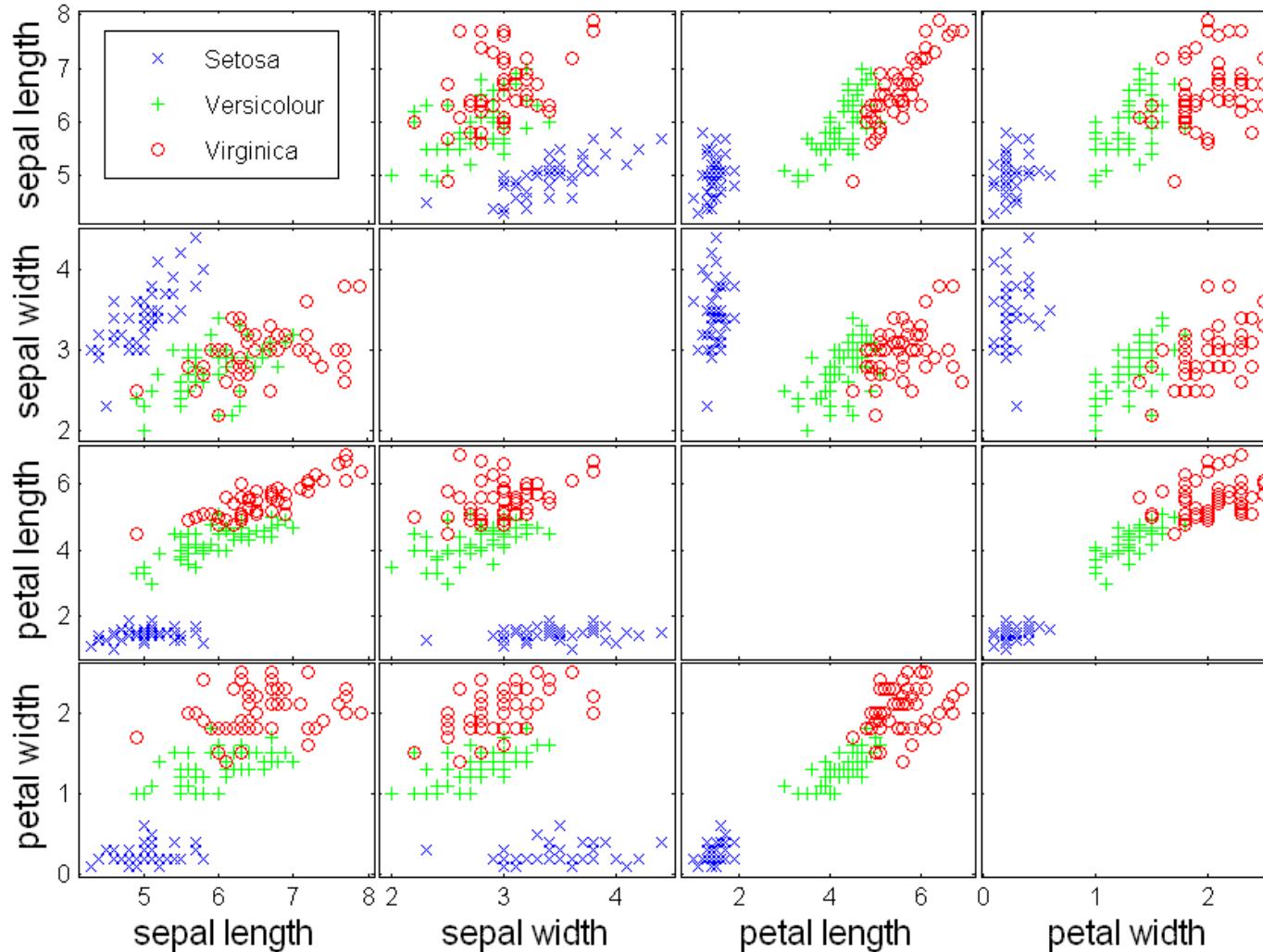
Box plots of attributes by Iris species

# Visualization Techniques: Scatter Plots

---

- Scatter plots
  - Özniteliklerin değerleri **konumu** belirler
  - En yaygın olan iki boyutlu dağılım (scatter) grafikleri, ancak **üç boyutlu dağılım grafikleri** olabilir
  - Genellikle, nesneleri temsil eden belirteçlerin (*markers*) **boyutu**, **şekli** ve **renki** kullanılarak ek öznitelikler görüntülenebilir.
  - **Dağılım grafiği dizileri**, **birkaç öznitelik çiftinin** ilişkilerini kompakt bir şekilde özetleyebilmesi açısından kullanışlıdır.
    - ◆ Sonraki slayttaki örnek

# Scatter Plot Array of Iris Attributes

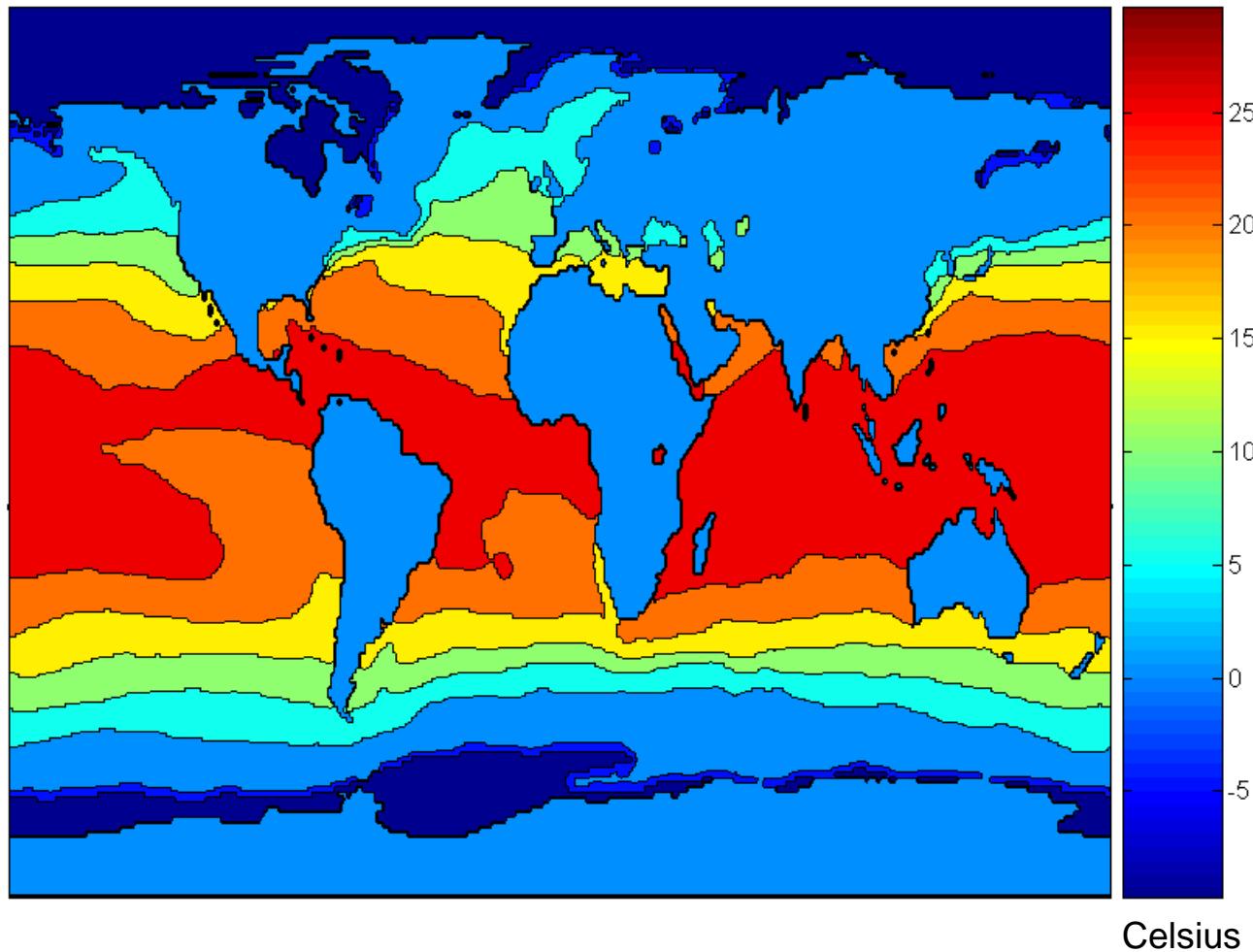


# Visualization Techniques: Contour Plots

---

- Bazı üç boyutlu veriler için, **iki öznitelik bir düzlemedeki bir konumu belirtirken**, üçüncüsü sıcaklık veya yükselti gibi **sürekli bir değere sahiptir**.
- Contour plots
  - Uzamsal bir ızgarada (**spatial grid**) **sürekli bir öznitelik** ölçüldüğünde kullanışlıdır.
  - **Düzlemi benzer değerlere sahip bölgelere ayıırlar**
  - düzlemi, üçüncü özelliğin (sıcaklık, yükselti) değerlerinin yaklaşık olarak aynı olduğu ayrı bölgelere ayırır
  - En yaygın örnek, arazi konumlarının yükseltilerinin kontur haritalarıdır.
  - Ayrıca sıcaklık, yağış, hava basıncı vb. görüntülenebilir.
    - ◆ Deniz Yüzeyi Sıcaklığına (SST) bir örnek sonraki slaytta verilmiştir.

# Contour Plot Example: SST Dec, 1998



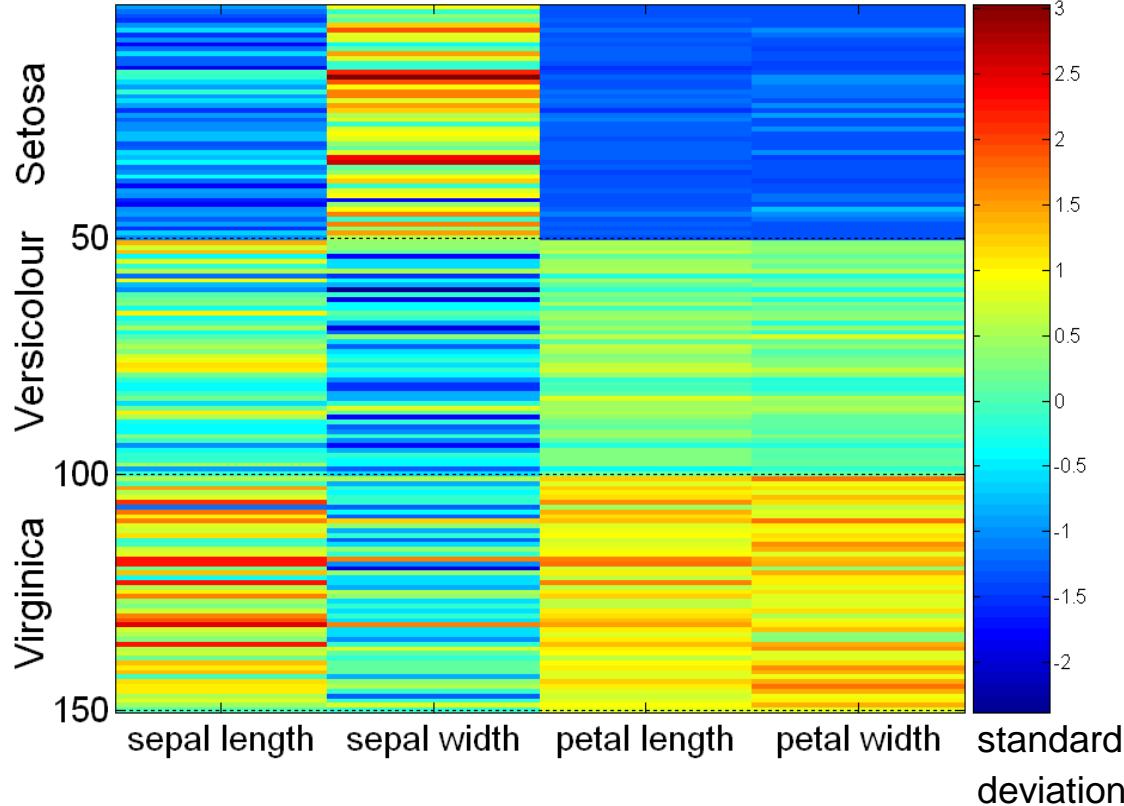
# Visualization Techniques: Matrix Plots

---

## ● Matrix plots

- Veri matrisinin her girdisi görüntüdeki bir piksel ile ilişkilendirilerek bir veri matrisi **bir görüntü olarak görselleştirilebilir.**
- **nesneler sınıfa göre sıralanır** (Sınıf etiketleri biliniyorsa)
  - böylece bir sınıfın tüm nesneleri bir arada olur
- Tipik olarak, **bir öznitelliğin grafiğe hakim olmasını önlemek için öznitelikler normalleştirilir**
  - ◆ Farklı özniteliklerin farklı aralıkları varsa öznitelikler genellikle sıfır ortalamaya (**mean of zero**) ve 1 standart sapmaya (**standard deviation of 1**) sahip olacak şekilde **standartlaştırılır**.
- Benzerlik veya uzaklık matrislerinin grafikleri, nesneler arasındaki ilişkileri görselleştirmek için de yararlı olabilir.
- Matris grafiklerinin örnekleri sonraki iki slaytta sunulmuştur.

# Visualization of the Iris Data Matrix

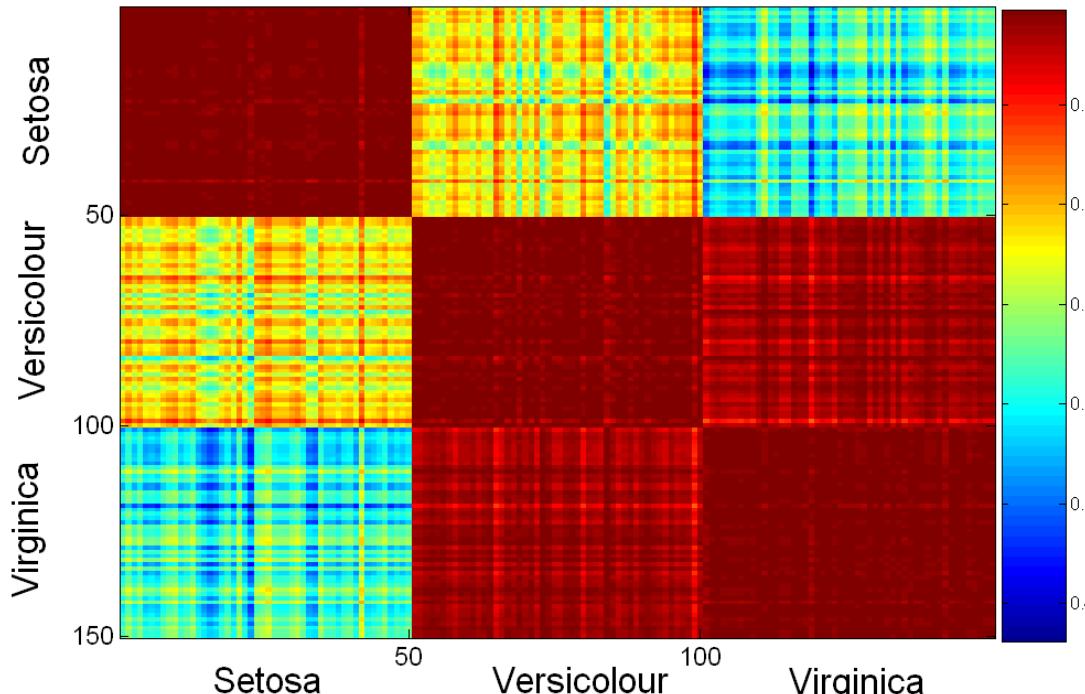


İlk 50 sıra **Setosa**, sonraki 50 **Versicolour** ve son 50 **Virginica** türünden Iris çiçeklerini temsil eder.

Setosa çiçeklerinin taç yaprağı (*petal*) genişliği ve uzunluğu **ortalamanın çok altındadır**, Versicolour çiçekleri ise **ortalama** taç genişliği ve **uzunluğuna** sahiptir. Virginica çiçeklerinin taç yaprağı genişliği ve uzunluğu **ortalamanın üzerinde**.

Sütunların ortalama 0 ve standart sapma 1 olacak şekilde standardize edildiği **Iris data matrix** grafiği

# Visualization of the Iris Correlation Matrix



Plot of the **Iris correlation matrix**.

Her gruptaki çiçekler birbirine en çok benziyor, ancak Versicolour ve Virginica Setosa'dan çok birbirine benziyor.

Bir dizi veri nesnesi için yakınlık matrisinin grafiğinde yapı (*structure*) aramak da yararlı olabilir.

Yine, **benzerlik matrisinin satırlarını ve sütunlarını** (sınıf etiketleri bilindiğinde), **bir sınıf**ındaki tüm nesnelerin bir arada olması için **sıralamak** yararlıdır.

Bu, **her bir sınıfın bağılılığını** ve diğer sınıflardan ayrılığının **görsel bir değerlendirmesine** izin verir.

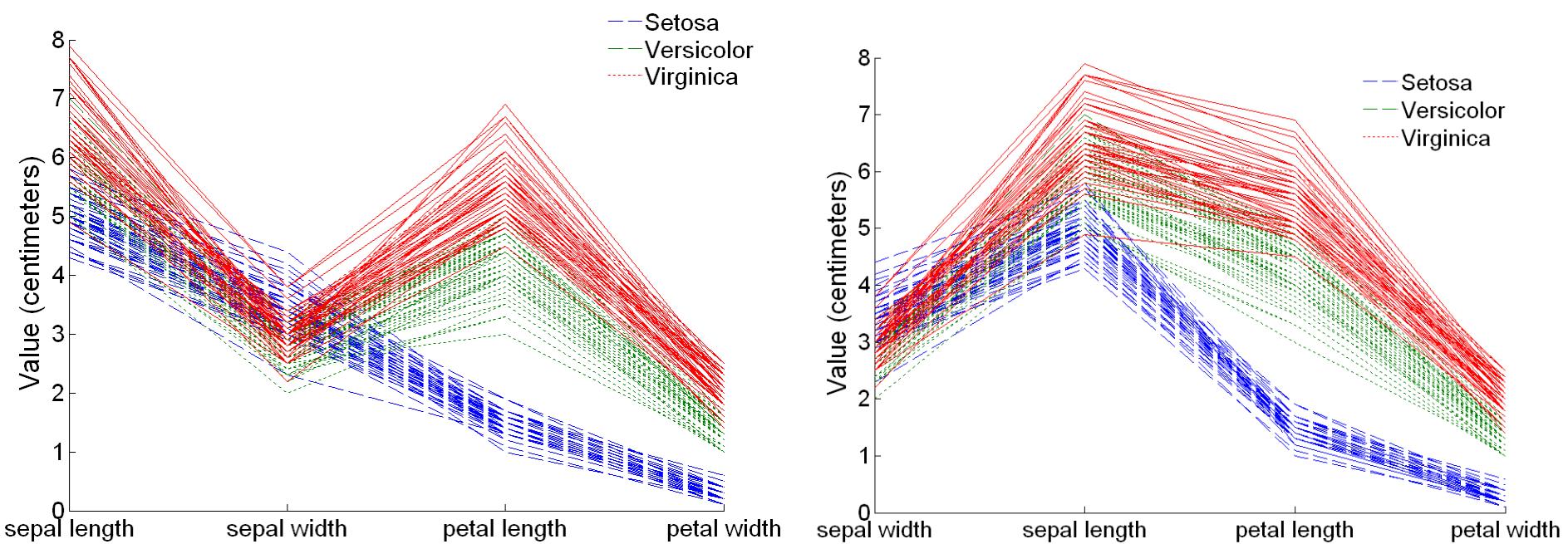
# Visualization Techniques: Parallel Coordinates

---

- Paralel Koordinatlar

- Yüksek boyutlu verilerin öznitelik değerlerini çizmek için kullanılır
- Dikey eksenler kullanmak yerine bir dizi paralel eksen kullanılır
- Her nesnenin öznitelik değerleri, **karşılık gelen her koordinat ekseninde bir nokta** olarak çizilir ve noktalar bir çizgi ile bağlanır.
- Böylece, **her nesne bir çizgi olarak temsil edilir**
- Çoğu zaman, **belli bir nesne sınıfını temsil eden çizgiler**, en azından bazı öznitelikler için **birlikte gruplanır**.
- Özniteliklerin sıralanması, bu tür gruplamaları görmede önemlidir

# Parallel Coordinates Plots for Iris Data

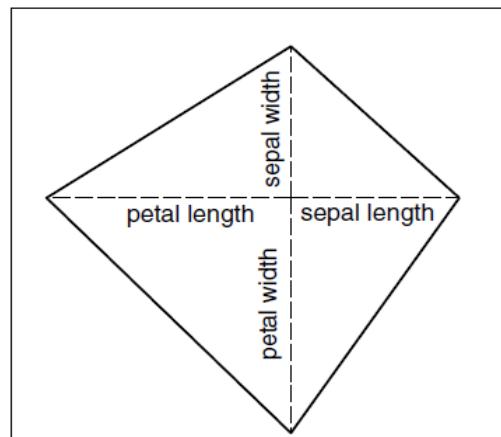


# Other Visualization Techniques

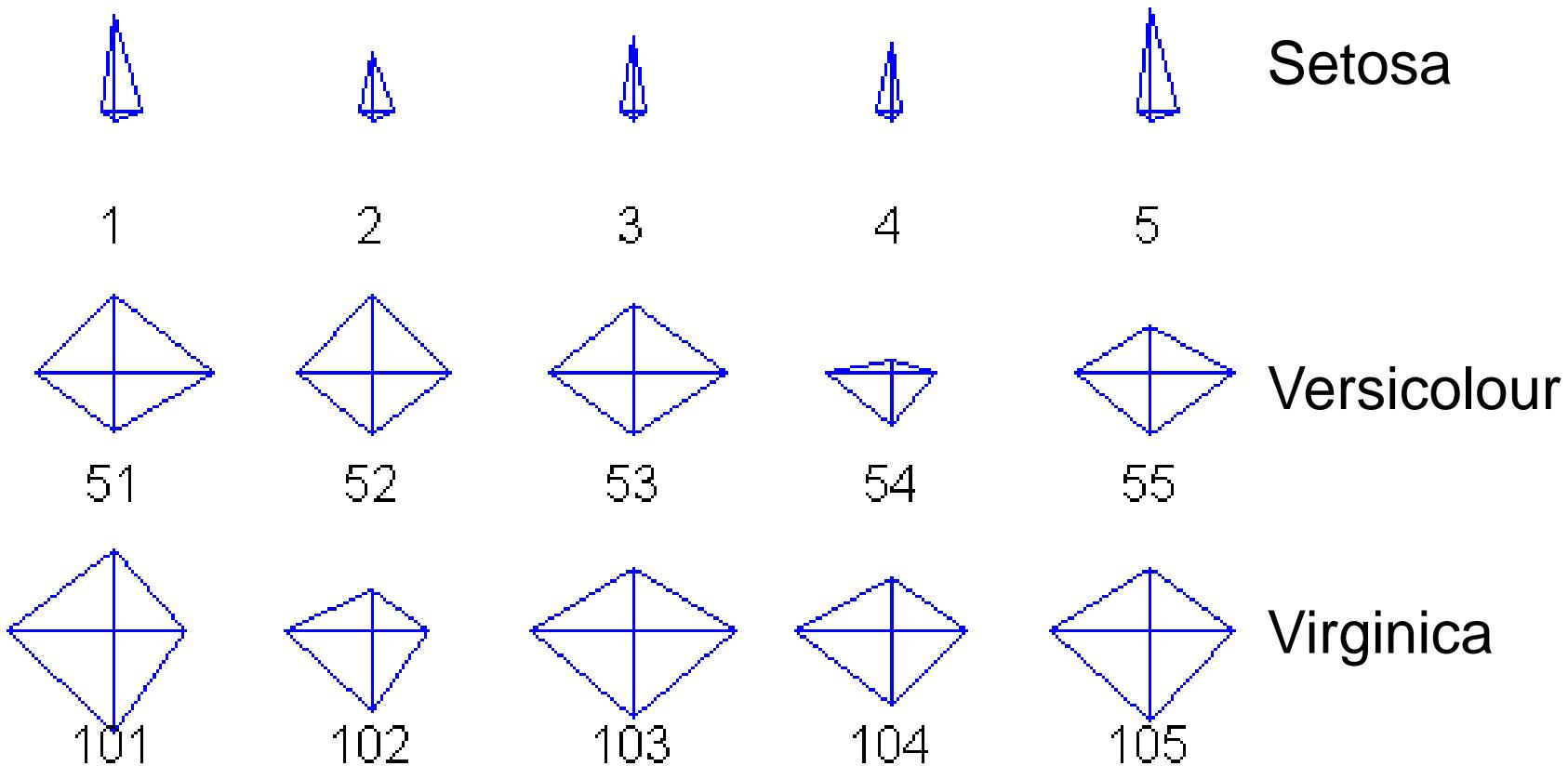
## ● Star Plots

- Paralel koordinatlara benzer yaklaşım, ancak **eksenler merkezi bir noktadan yayılır**
- Bu teknik, her özellik için bir eksen kullanır.
- Tipik olarak, tüm öznitelik değerleri [0,1] aralığına eşlenir.
- Bir nesnenin değerlerini birleştiren çizgi bir **çokgendir**

Iris veri  
kümesinin 150.  
çiçeğinin yıldız  
koordinatları  
grafigi



# Star Plots for Iris Data

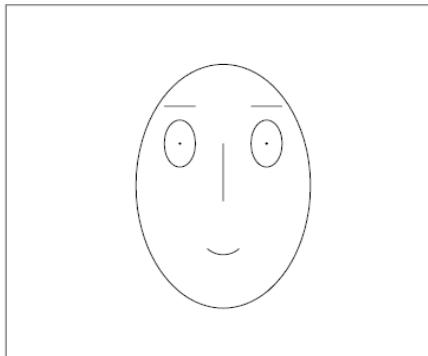


# Other Visualization Techniques

## ● Chernoff Faces

- Herman Chernoff tarafından oluşturulan yaklaşım
- Bu yaklaşım, her bir özelliği yüzün bir özelliğiyile ilişkilendirir.
- Her özelliğin değerleri, **karşılık gelen yüz karakteristiğinin görünümünü belirler.**
- **Her nesne ayrı bir yüz** olur
- **İnsanın yüzleri ayırt etme yeteneğine** dayanır

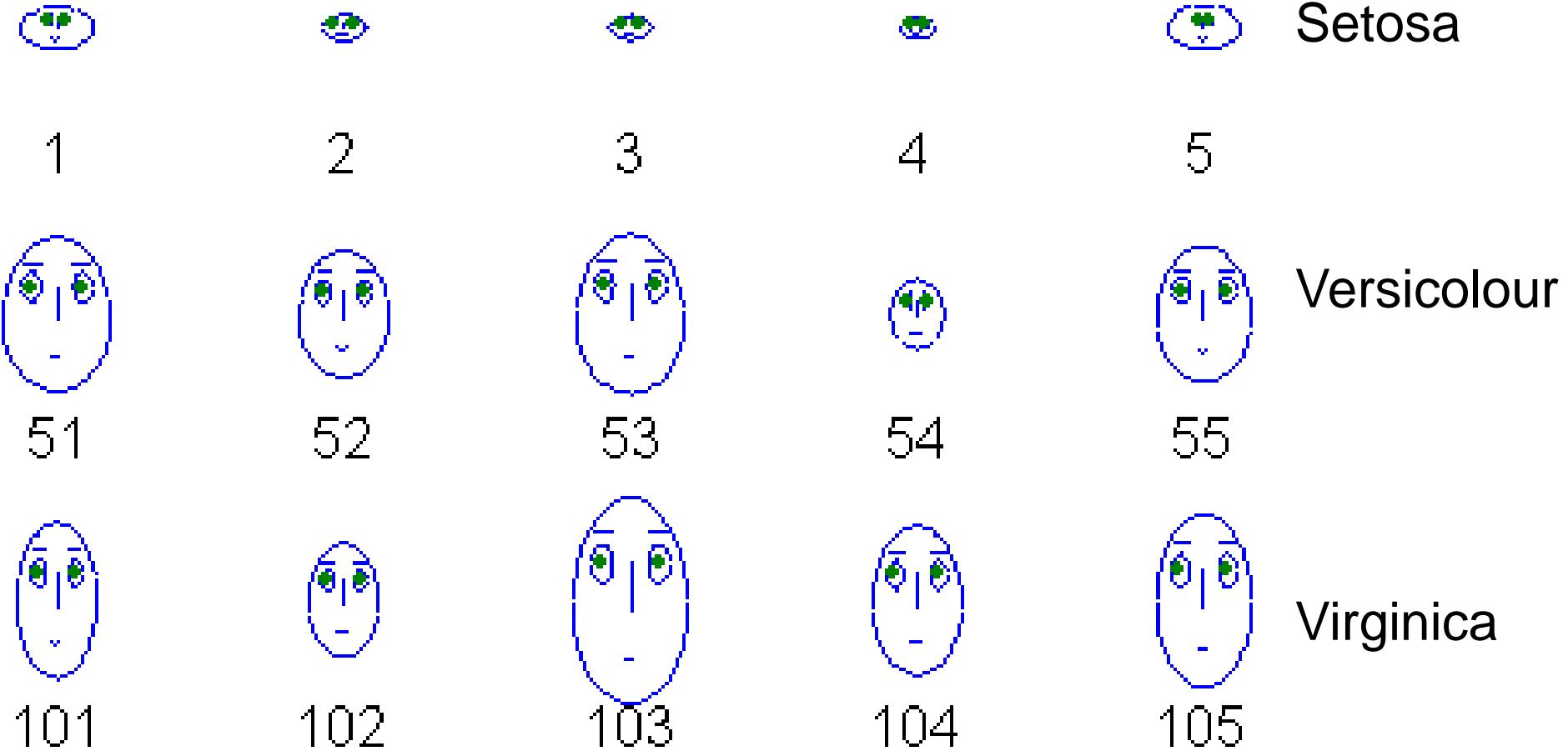
Iris veri  
kümesinin  
150.  
çiçeğinin  
Chernoff  
yüzü



Data Feature	Facial Feature
sepal length	size of face
sepal width	forehead/jaw relative arc length
petal length	shape of forehead
petal width	shape of jaw

Gözler arası genişlik ve ağız uzunluğu gibi yüzün **düzenli** özelliklerine **varsayılan değerler** verilmiştir.

# Chernoff Faces for Iris Data



# OLAP

---

- **On-Line Analytical Processing (OLAP)** ilişkisel veritabanının babası olarak bilinen Edgar Frank Codd tarafından önerildi.
- İlişkisel veritabanları verileri tablolara koyarken, **OLAP çok boyutlu bir dizi temsili kullanır.**
  - Verilerin bu tarz temsil edilmesi daha önce istatistik ve diğer alanlarda olmuştur.
- Böyle bir veri temsiliyle daha kolay hale gelen bir dizi veri analizi ve veri keşfi işlemi vardır.

# Creating a Multidimensional Array

---

- Tablo verilerinin çok boyutlu bir diziye dönüştürülmesinde iki temel adım.
  - İlk olarak, **hangi özniteliklerin boyutları olacağını** ve **hangi özniteliğin değerleri çok boyutlu dizide girişler (entry) olarak görünen hedef öznitelik (target attribute)** olacağını belirleyin.
    - ◆ **Boyut** olarak kullanılan öznitelikler **ayrık değerlere sahip olmalıdır**
    - ◆ **Hedef değer** tipik olarak **bir sayı veya sürekli bir değerdir**, örneğin bir öğenin maliyeti
  - İkinci olarak, (hedef özniteliğin) değerlerini veya o girdiye karşılık gelen öznitelik değerlerine sahip tüm nesnelerin sayısını toplayarak **çok boyutlu dizideki her girdinin değerini** bulun.

# Example: Iris data

- Özniteliklerin (petal length, petal width, and species type) çok boyutlu bir diziye nasıl dönüştürüleceği:
  - İlk olarak, petal width ve petal length'ı kategorik değerlere sahip olacak şekilde ayıriklaştırırız: *low*, *medium*, ve *high*
  - Aşağıdaki tabloyu elde ederiz - count özniteligiine dikkat edin

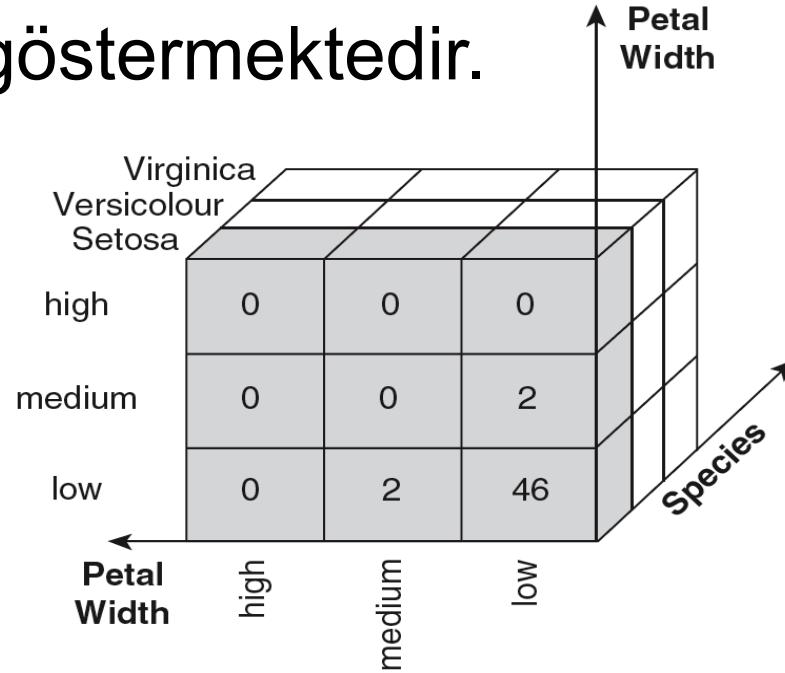
Petal Length	Petal Width	Species Type	Count
low	low	Setosa	46
low	medium	Setosa	2
medium	low	Setosa	2
medium	medium	Versicolour	43
medium	high	Versicolour	3
medium	high	Virginica	3
high	medium	Versicolour	2
high	medium	Virginica	3
high	high	Versicolour	2
high	high	Virginica	44

## Discretization

- Category boundaries for **petal width**
  - *low* → [0, 0.75)
  - *medium* → [0.75, 1.75)
  - *high* → [1.75, ∞)
- Category boundaries for **petal length**
  - *low* → [0, 2.5)
  - *medium* → [2.5, 5)
  - *high* → [5, ∞)

# Example: Iris data (continued)

- Petal width, petal length ve species type'ın her benzersiz demeti (tuple), dizinin (array) bir öğesini tanımlar.
- Bu elemana karşılık gelen sayı değeri atanır.
- Yandaki şekil sonucu göstermektedir.
- Belirtilmemiş tüm demetler 0'dır.  
*(All non-specified tuples are 0.)*



A multidimensional data representation for the Iris data set

# Example: Iris data (continued)

- Çok boyutlu dizinin dilimleri aşağıdaki çapraz tablolarla (cross-tabulations) gösterilmiştir.
- Bu tablolar bize ne anlatıyor?

		Width		
		low	medium	high
Length	low	46	2	0
	medium	2	0	0
	high	0	0	0

Cross-tabulation of flowers according to petal length and width for flowers of the **Setosa species**.

		Width		
		low	medium	high
Length	low	0	0	0
	medium	0	43	3
	high	0	2	2

Cross-tabulation of flowers according to petal length and width for flowers of the **Versicolour species**.

Bu tablolar, her Iris türünün **petal uzunluğu ve genişliğinin farklı bir değer kombinasyonu ile karakterize olduğunu göstermektedir**.

		Width		
		low	medium	high
Length	low	0	0	0
	medium	0	0	3
	high	0	3	44

Cross-tabulation of flowers according to petal length and width for flowers of the **Virginica species**.

**Setosa** çiçekleri düşük genişlik ve uzunluktadır, **Versicolour** çiçekleri orta genişlik ve uzunluktadır ve **Virginica** çiçekleri yüksek genişlik ve uzunluktadır.

# OLAP Operations: Data Cube

---

- Bir OLAP'ın temel işlemi bir veri küpünün (**data cube**) oluşmasıdır
- Veri küpü, verilerin tüm olası toplamlarla (aggregates) birlikte çok boyutlu bir temsilidir.
- Olası tüm toplamlar derken, boyutların uygun bir alt kümesini seçerek ve kalan tüm boyutların toplamını alarak sonuçlanan toplamaları kastediyoruz.
- Örneğin, Iris verilerinin **species type** boyutunu **seçersek** ve diğer tüm boyutların toplamını alırsak, sonuç, her biri her bir türün çiçek sayısını veren üç entry'li tek boyutlu bir girdi olacaktır.

# Data Cube Example

- Çeşitli tarihlerde bir dizi şirket mağazasında ürünlerin satışını kaydeden bir veri kümlesi düşünün.

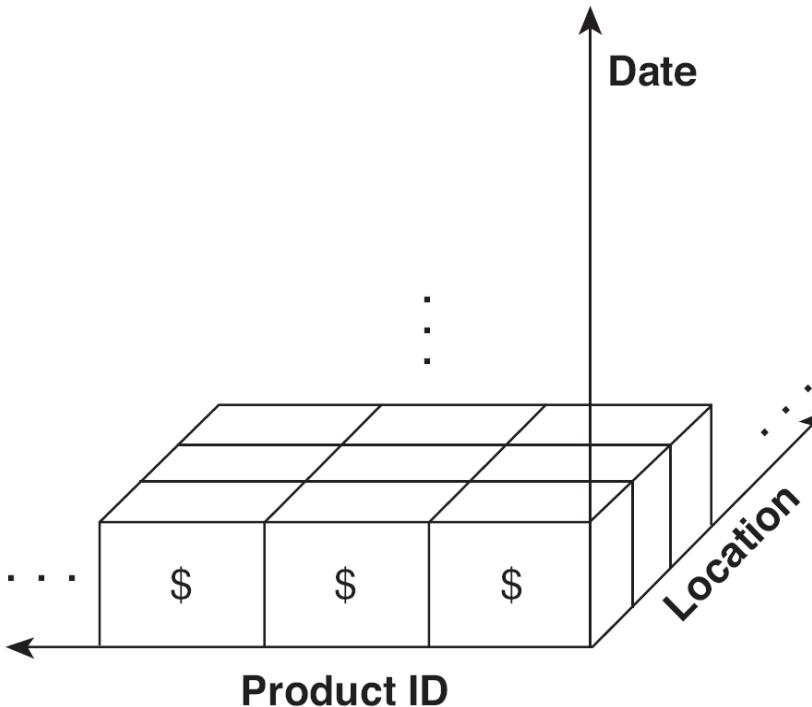
Table 3.11. Sales revenue of products (in dollars) for various locations and times.

Product ID	Location	Date	Revenue
:	:	:	:
1	Minneapolis	Oct. 18, 2004	\$250
1	Chicago	Oct. 18, 2004	\$79
:	:	:	:
1	Paris	Oct. 18, 2004	301
:	:	:	:
27	Minneapolis	Oct. 18, 2004	\$2,321
27	Chicago	Oct. 18, 2004	\$3,278
:	:	:	:
27	Paris	Oct. 18, 2004	\$1,325
:	:	:	:

The dimensions of the multidimensional representation are the **product ID, location, and date** attributes, while the target attribute is the **revenue**.

# Data Cube Example

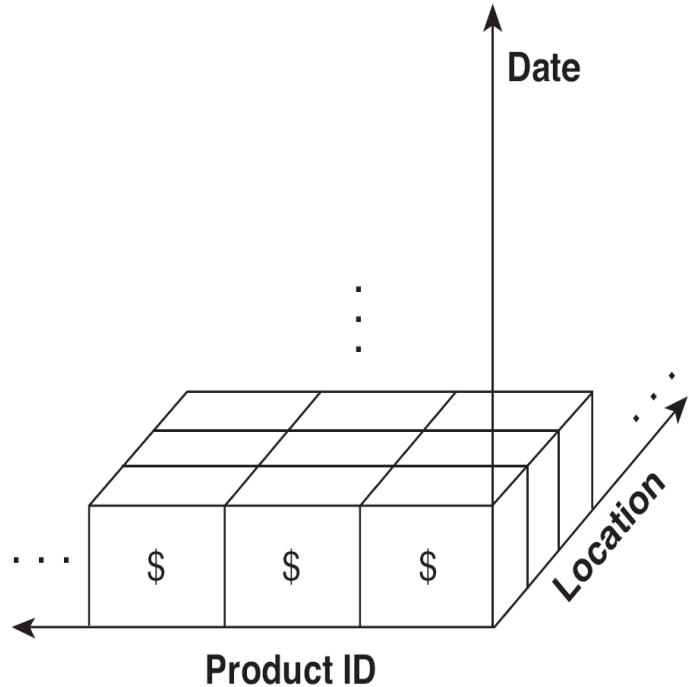
- Bu veriler 3 boyutlu bir dizi olarak gösterilebilir



Multidimensional data representation for sales data.

# Data Cube Example

- There are 3 two-dimensional aggregates,  
3 one-dimensional aggregates,  
and 1 zero-dimensional aggregate (the overall total)



# Data Cube Example (continued)

- Tablo, çeşitli tarih (date) ve ürün (product) kombinasyonları için **tüm konumların toplamının (summing over all locations)** sonucunu gösterir.

Table 3.12. Totals that result from summing over all locations for a fixed time and product.

product ID	date			
	Jan 1, 2004	Jan 2, 2004	...	Dec 31, 2004
1	\$1,001	\$987	...	\$891
:	:			:
27	\$10,265	\$10,225	...	\$9,325
:	:			:

Basit olması için, tüm tarihlerin bir yıl içinde olduğunu varsayıyalım. Yılda 365 gün ve 1000 ürün varsa, Tablo 3.12'de her ürün-veri çifti için bir tane olmak üzere **365.000 girdi (toplam) vardır**.

We could also specify

- the store location and date and **sum over products**, or
- the location and product and **sum over all dates**.

# Data Cube

---

- Verilerin çok boyutlu temsili (multidimensional representation), tüm olası toplamlarla (aggregates) birlikte **veri küpü (data cube)** olarak bilinir.
- İsmine rağmen, her boyutun büyüklüğünün (öznitelik değerlerinin sayısı) **eşit olması gerekmek**.
- Ayrıca, bir veri küpünün üçten fazla veya daha az boyutu olabilir.
- Daha da önemlisi, bir veri küpü, istatistiksel terminolojide çapraz tablo (**cross-tabulation**) olarak bilinen şeyin bir genellemesidir

# Data Cube Example (continued)

- Aşağıdaki şekildeki tablo, iki boyutlu toplamalardan (*two dimensional aggregates*) birini, iki tane tek boyutlu toplamayı (*one-dimensional aggregates*) ve genel toplamı (*overall total*) gösterir.

Table 3.13. Table 3.12 with marginal totals.

product ID	date				total
	Jan 1, 2004	Jan 2, 2004	...	Dec 31, 2004	
1	\$1,001	\$987	...	\$891	\$370,000
:	:			:	:
27	\$10,265	\$10,225	...	\$9,325	\$3,800,020
:	:			:	:
total	\$527,362	\$532,953	...	\$631,221	\$227,352,127

These totals are the result of **further summing over** either **dates** or **products**.

# OLAP Operations: Slicing and Dicing

---

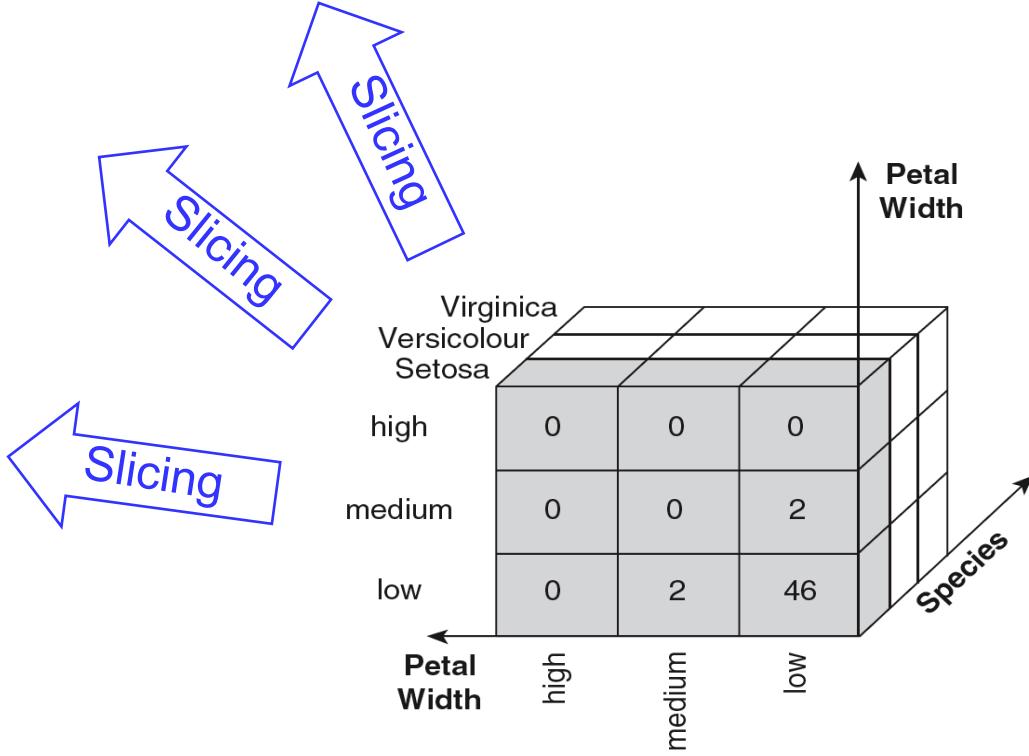
- **Slicing**, bir veya daha fazla boyut için belirli bir değer belirterek **tüm çok boyutlu diziden bir hücre grubu seçmektir.**
- **Dicing**, bir öznitelik değerleri aralığı belirleyerek **bir hücre alt kümesini seçmeyi** içerir.
  - Bu, **tüm diziden bir alt dizi** tanımlamaya eşdeğerdir.
- Uygulamada, her iki işleme de bazı boyutlarda birleştirme (*aggregation*) eşlik edebilir.

# Slicing operation

		Width		
		low	medium	high
Length	low	46	2	0
	medium	2	0	0
	high	0	0	0

		Width		
		low	medium	high
Length	low	0	0	0
	medium	0	43	3
	high	0	2	2

		Width		
		low	medium	high
Length	low	0	0	0
	medium	0	0	3
	high	0	3	44



# OLAP Operations: Roll-up and Drill-down

---

- Öznitelik değerleri genellikle **hiyerarşik bir yapıya** sahiptir.
  - Her tarih; bir yıl, ay ve hafta ile ilişkilendirilir.
  - Bir konum; kıta, ülke, eyalet (il vb.) ve şehir ile ilişkilidir.
  - Ürünler; giyim, elektronik ve mobilya gibi çeşitli kategorilere ayrılabilir.
- Bu kategorilerin genellikle iç içe geçtiğini ve bir ağaç (*tree*) veya kafes (*lattice*) oluşturduğunu unutmayın.
  - Bir yıl, günleri içeren ayları içerir
  - Bir ülke, eyaletleri içerir ve onlar da şehirleri içerir.

# OLAP Operations: Roll-up and Drill-down

---

- Bu hiyerarşik yapı, **roll-up** (yuvarlama) ve **drill-down** (detaya inme) işlemlerine imkan tanır.
  - Satış verileri için, satışları bir aydaki tüm tarihlerdekileri toplayarak birleştirebiliriz. (**roll up**)
  - Tersine, zaman boyutunun aylara bölündüğü verilerin bir görünümü verildiğinde, **aylık satış toplamlarını detayına** inerek **günlük satış toplamlarına** geçebiliriz. (**drill down**)
    - ❖ Elbette **bu**, **temel satış verilerinin günlük ayrıntı düzeyinde (daily granularity)** mevcut olmasını **gerektirir**.
  - Aynı şekilde, konum (location) veya ürün numarası (product ID) özelliklerinde **roll up** veya **drill-down** yapılabilir.

# **Data Mining Classification: Basic Concepts and Techniques, Decision Trees**

---

---

Lecture Notes for Chapter 4

Introduction to Data Mining, 2<sup>nd</sup> Edition

by

Tan, Steinbach, Karpatne, Kumar

# Classification: Definition

---

- Bir kayıt koleksiyonu verildiğinde (*training set*)
  - Her kayıt bir çok-ögeli bir veri grubu  $(x, y)$  ile karakterize edilir, burada  $x$  öznitelik kümesidir ve  $y$  sınıf etiketidir.
    - ◆  $x$ : öznitelik (*attribute*), öngösterge (*predictor*), bağımsız değişken (*independent variable*), input
    - ◆  $y$ : sınıf (*class*), response, bağımlı değişken (*dependent variable*), output
- Görev:
  - Her bir öznitelik kümesi  $x$  'i önceden tanımlanmış sınıf etiketlerinden ( $y$ ) birine eşleyen bir model öğrenmek

# Classification: Descriptive Modeling

Bir sınıflandırma modeli, farklı sınıflardan nesneler arasında ayırmak için açıklayıcı bir araç görevi görebilir.

Table 4.1. The vertebrate data set. (*Omurgalılar veri seti*)

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard	cold-blooded	scales	yes	yes	no	no	no	fish
shark								
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

# Classification: Predictive Modeling

**Predictive Modeling (Öngörücü modelleme)** Bir sınıflandırma modeli, bilinmeyen kayıtların sınıf etiketini tahmin etmek için de kullanılabilir. Şekil 4.2'de gösterildiği gibi, bir sınıflandırma modeli, bilinmeyen bir kaydın öznitelik kümesiyle sunulduğunda otomatik olarak bir sınıf etiketi atayan bir kara kutu (**black box**) olarak değerlendirilebilir.

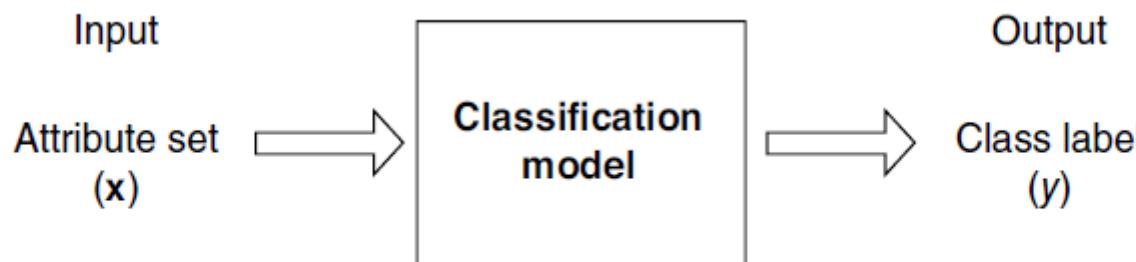


Figure 4.2. Classification as the task of mapping an input attribute set  $x$  into its class label  $y$ .

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class Label
gila monster	cold-blooded	scales	no	no	no	yes	yes	?

# Examples of Classification Task

---

Task	Attribute set, $x$	Class label, $y$
Categorizing email messages	Features extracted from email message header and content	spam or non-spam
Identifying tumor cells	Features extracted from MRI scans	malignant or benign cells
Cataloging galaxies	Features extracted from telescope images	Elliptical, spiral, or irregular-shaped galaxies

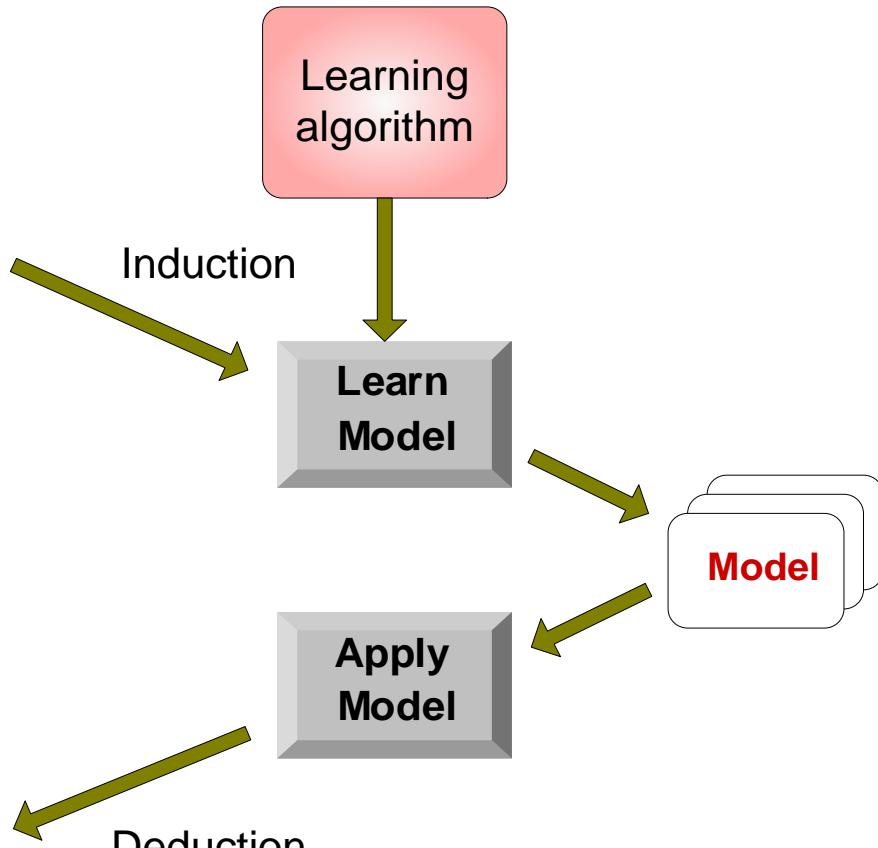
# General Approach for Building Classification Model

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



# Classification Techniques

---

- Base Classifiers
  - Decision Tree based Methods
  - Rule-based Methods
  - Nearest-neighbor
  - Neural Networks
  - Deep Learning
  - Naïve Bayes and Bayesian Belief Networks
  - Support Vector Machines
- Ensemble Classifiers
  - Boosting, Bagging, Random Forests

# Decision Tree Induction: How a Decision Tree Works

---

- Ağacın üç tür düğümü vardır:
  - Giren kenarı olmayan (**no incoming edges**) ve sıfır veya daha fazla çıkan kenarı olan kök düğüm (**root node** )
  - İç düğüm (**Internal nodes**), Her biri tam olarak bir giren kenara (**one incoming edge** ) ve iki veya daha fazla çıkan kenara (**two or more outgoing edges**) sahip olan iç düğümler.
  - Yaprak veya uç düğümler (**Leaf or terminal nodes**), her biri tam olarak bir giren kenara (**one incoming edge** ) sahiptir ve çıkan kenarı yoktur (**no outgoing edges**).

# Decision Tree Induction: How a Decision Tree Works

---

- Bir karar ağacında, her **yaprak düğüme** bir **sınıf etiketi** atanır.
- Kök ve diğer iç düğümleri içeren uç-birim olmayan düğümler (**nonterminal nodes**), farklı özelliklere sahip kayıtları ayırmak için öznitelik test koşullarını (**attribute test conditions**) içerir.
  - Örneğin, Şekil 4.4'te gösterilen kök düğüm, sıcakkanlıları (**warm-blooded**) soğukkanlı (**cold-blooded**) omurgalılardan ayırmak için Body Temperature özniteliğini kullanır.

# Decision Tree Induction: How a Decision Tree Works

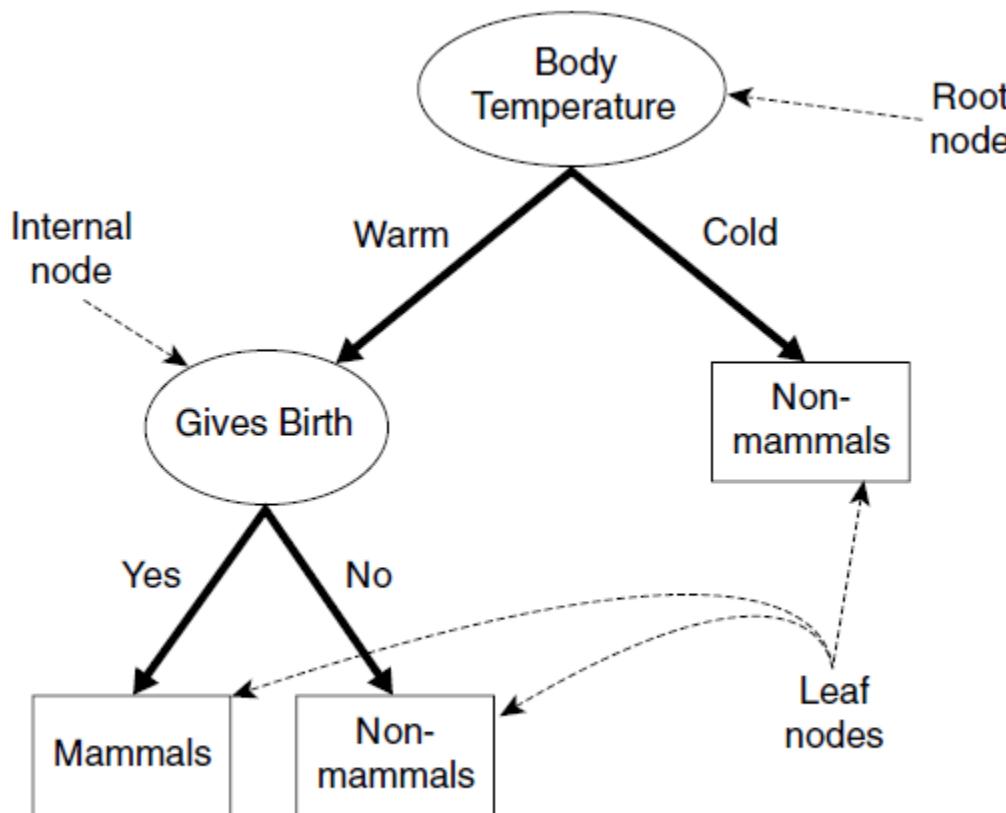


Figure 4.4. A decision tree for the mammal classification problem.

Tüm soğukkanlı omurgalılar **non-mammals** olduğu için, kök düğümün sağ çocuğu olarak Non-mammals etiketli bir yaprak düğümü oluşturulur.

Bir karar ağacı oluşturulduktan sonra, **bir test kaydının sınıflandırılması kolaydır**. Kök düğümden başlayarak, test koşulunu kayda uygularız ve testin sonucuna göre uygun dalı takip ederiz.

# Decision Tree Induction: How a Decision Tree Works

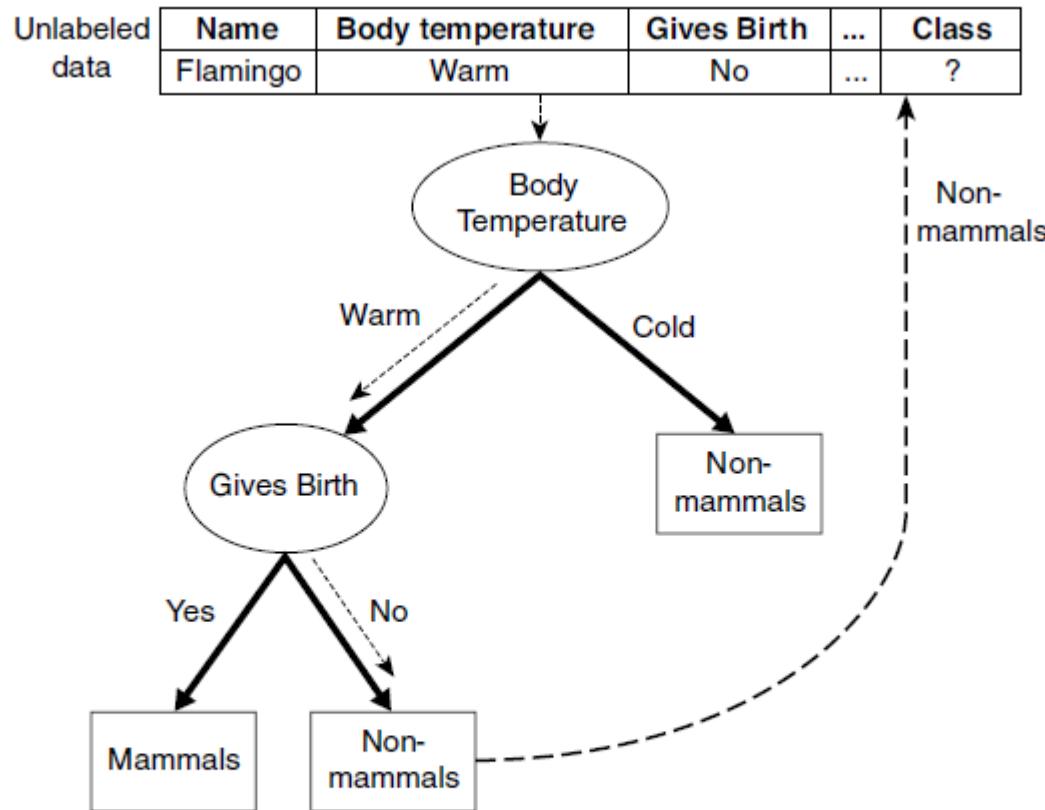
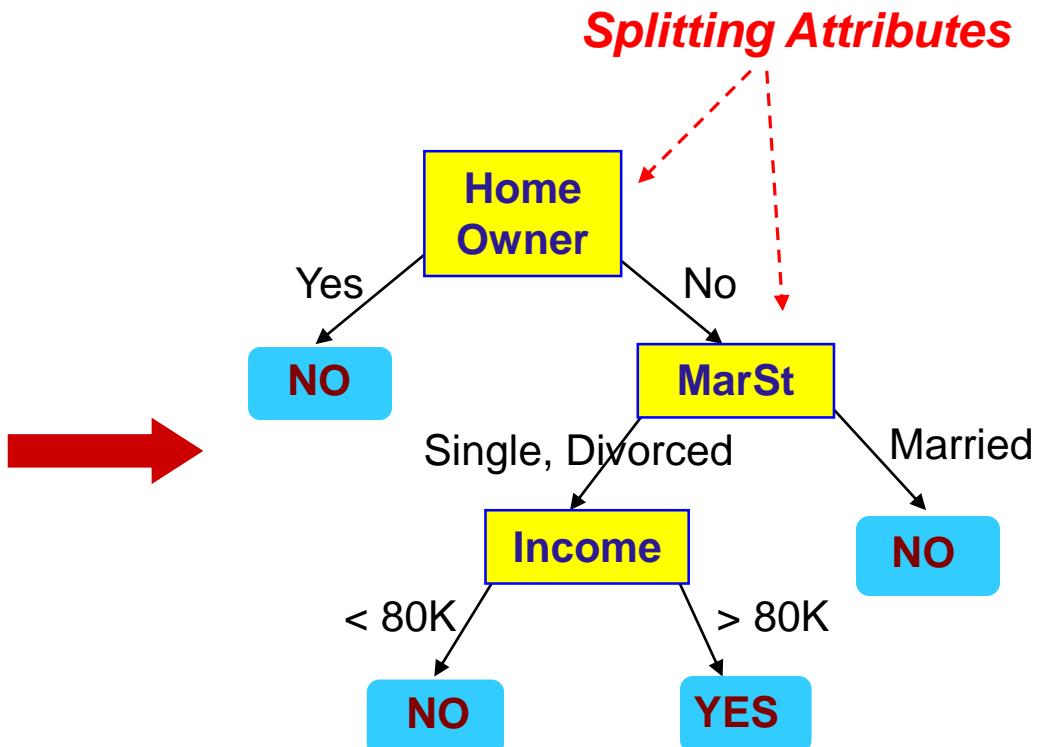


Figure 4.5. Classifying an unlabeled vertebrate. The dashed lines represent the outcomes of applying various attribute test conditions on the unlabeled vertebrate. The vertebrate is eventually assigned to the Non-mammal class.

# Example of a Decision Tree

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower	class
1	Yes	Single	125K	No	
2	No	Married	100K	No	
3	No	Single	70K	No	
4	Yes	Married	120K	No	
5	No	Divorced	95K	Yes	
6	No	Married	60K	No	
7	Yes	Divorced	220K	No	
8	No	Single	85K	Yes	
9	No	Married	75K	No	
10	No	Single	90K	Yes	

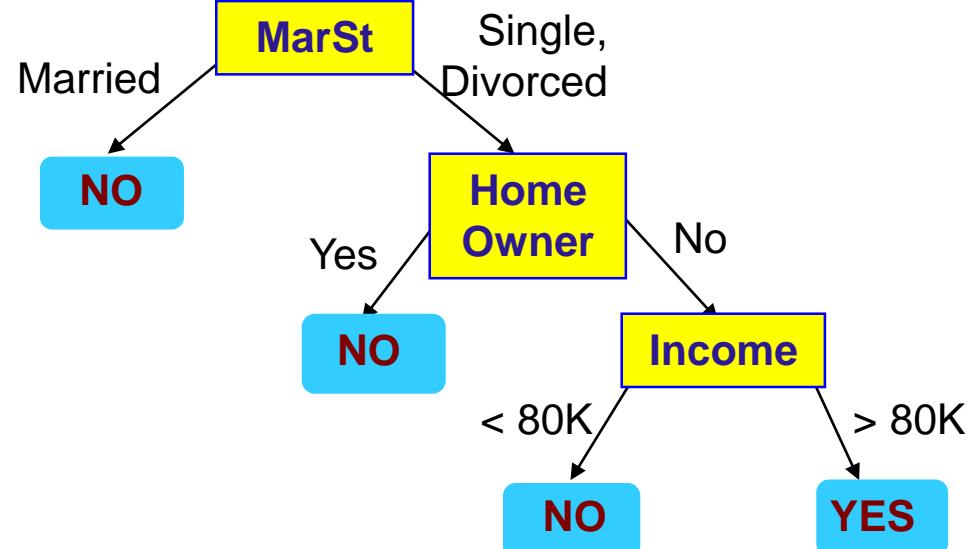


Training Data

Model: Decision Tree

# Another Example of Decision Tree

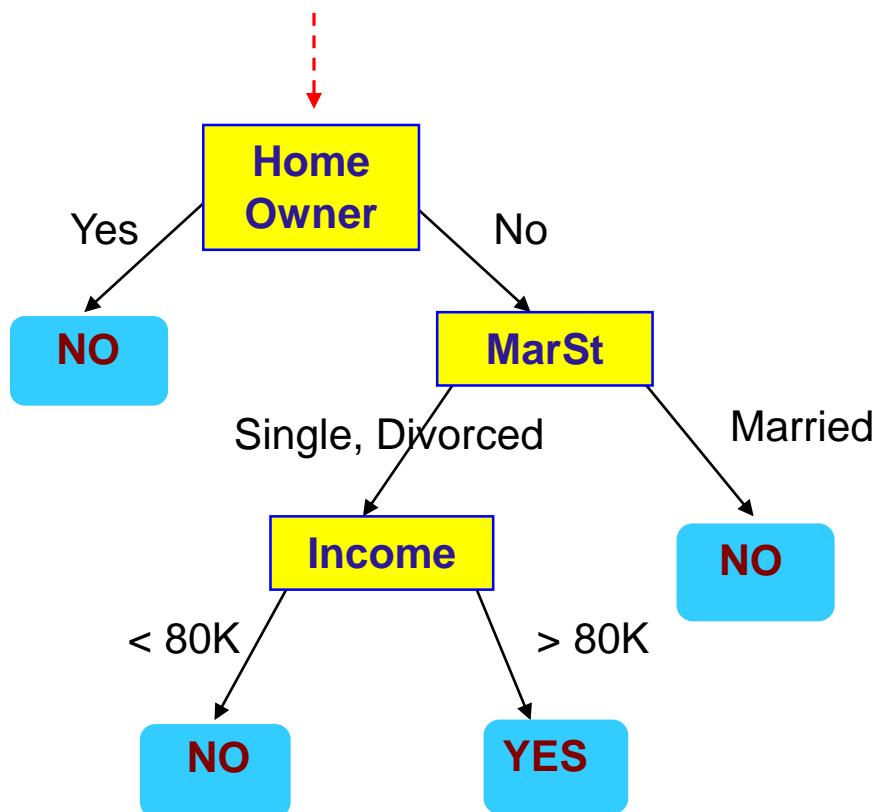
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower	
				categorical	categorical
1	Yes	Single	125K	No	continuous
2	No	Married	100K	No	class
3	No	Single	70K	No	
4	Yes	Married	120K	No	
5	No	Divorced	95K	Yes	
6	No	Married	60K	No	
7	Yes	Divorced	220K	No	
8	No	Single	85K	Yes	
9	No	Married	75K	No	
10	No	Single	90K	Yes	



Aynı verilere uyan birden fazla ağaç olabilir!

# Apply Model to Test Data

Start from the root of tree.



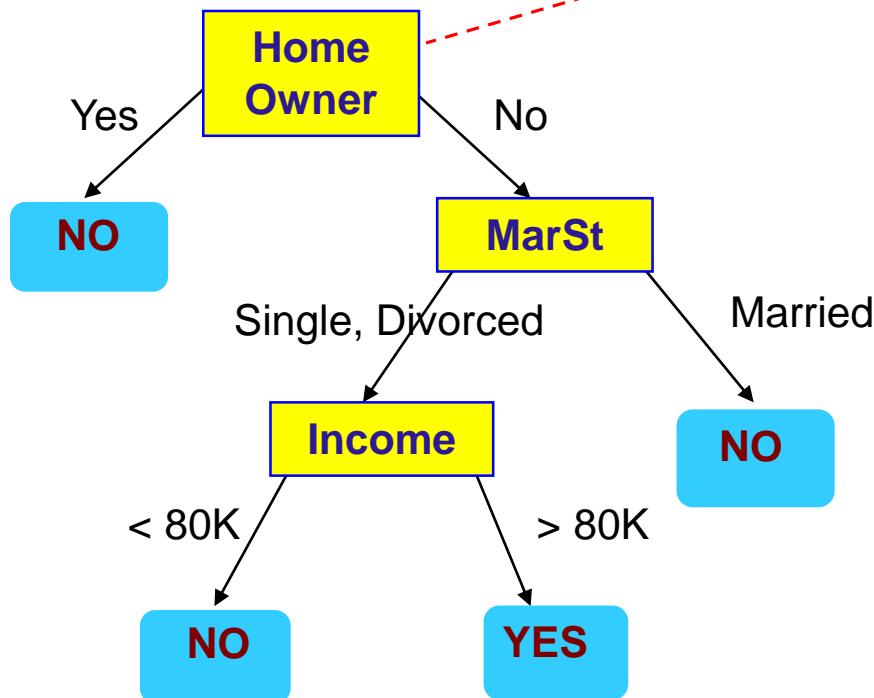
## Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

# Apply Model to Test Data

Test Data

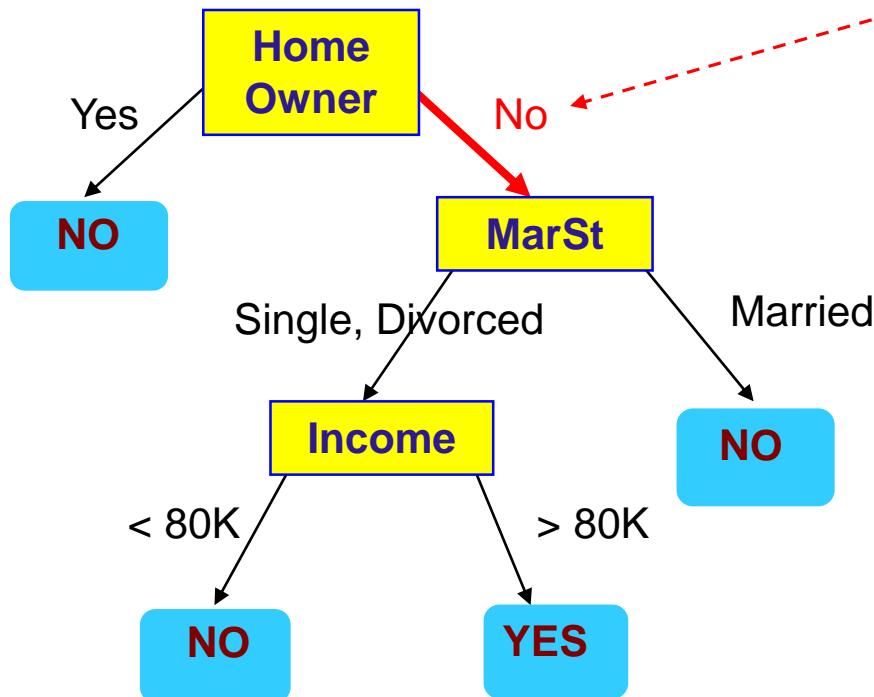
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



# Apply Model to Test Data

Test Data

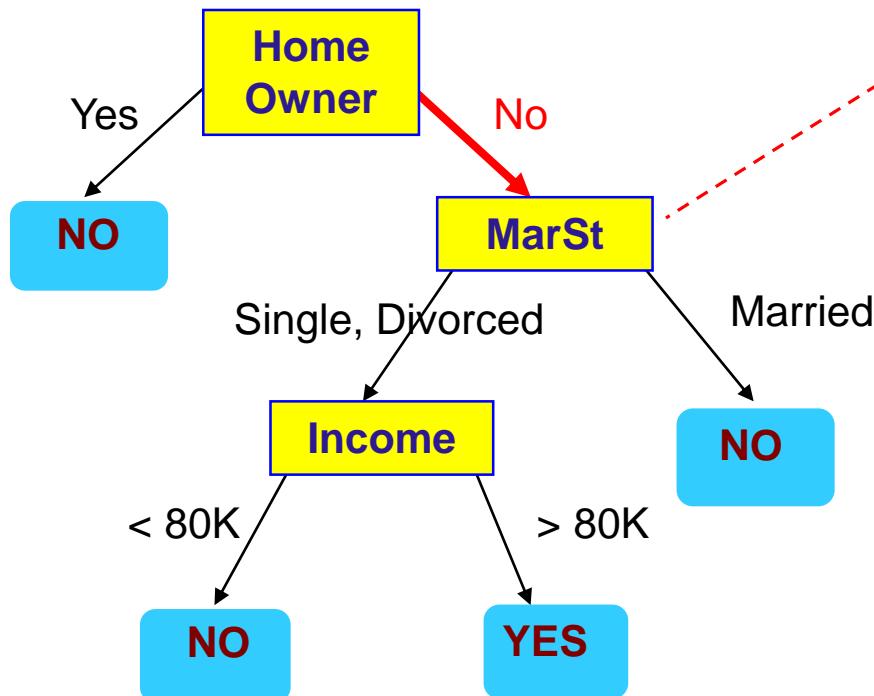
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



# Apply Model to Test Data

Test Data

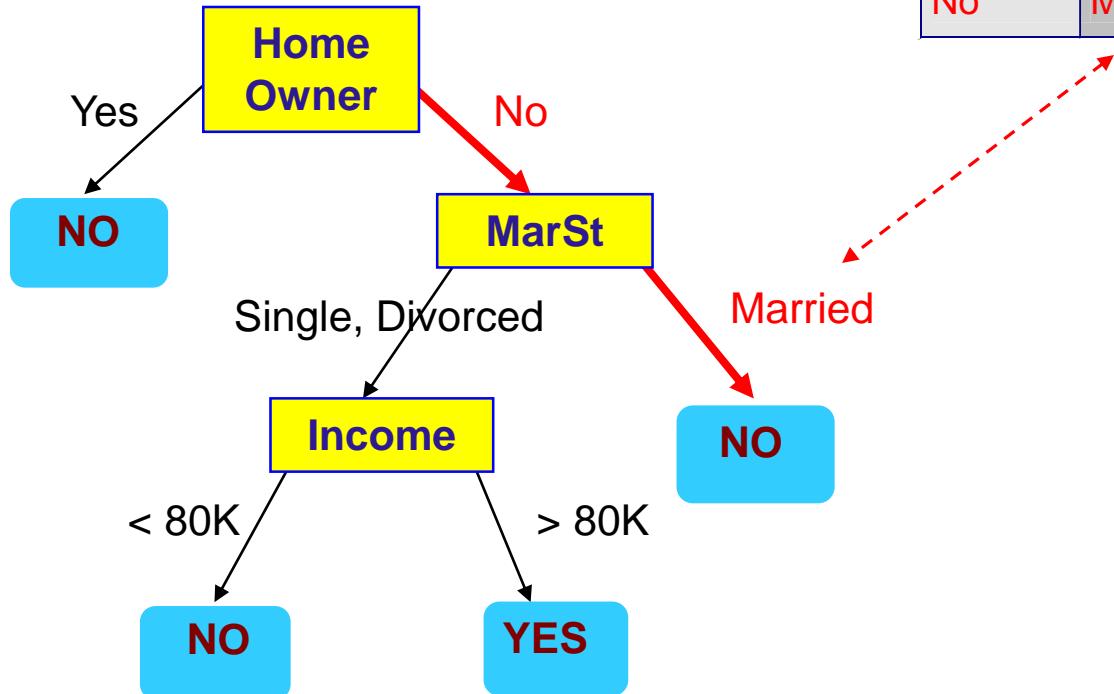
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



# Apply Model to Test Data

## Test Data

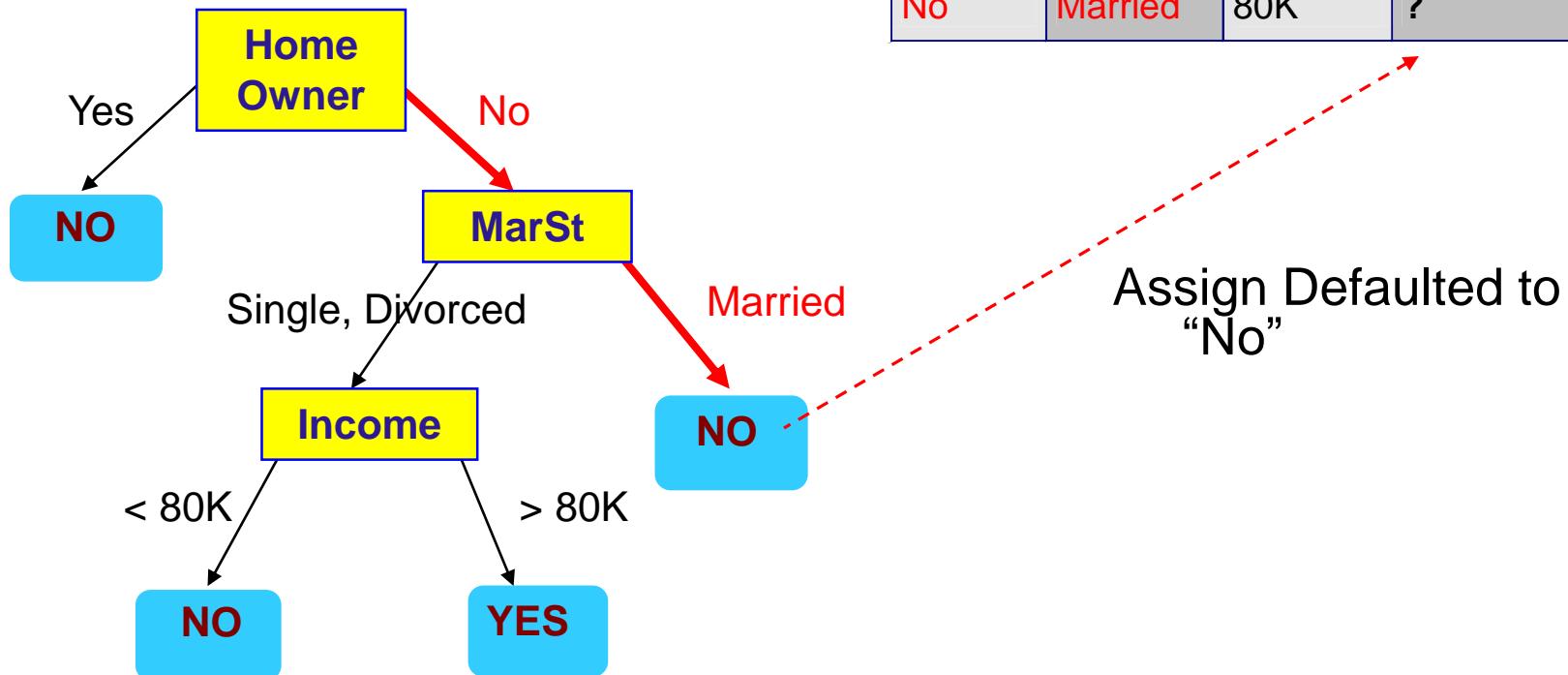
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



# Apply Model to Test Data

## Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



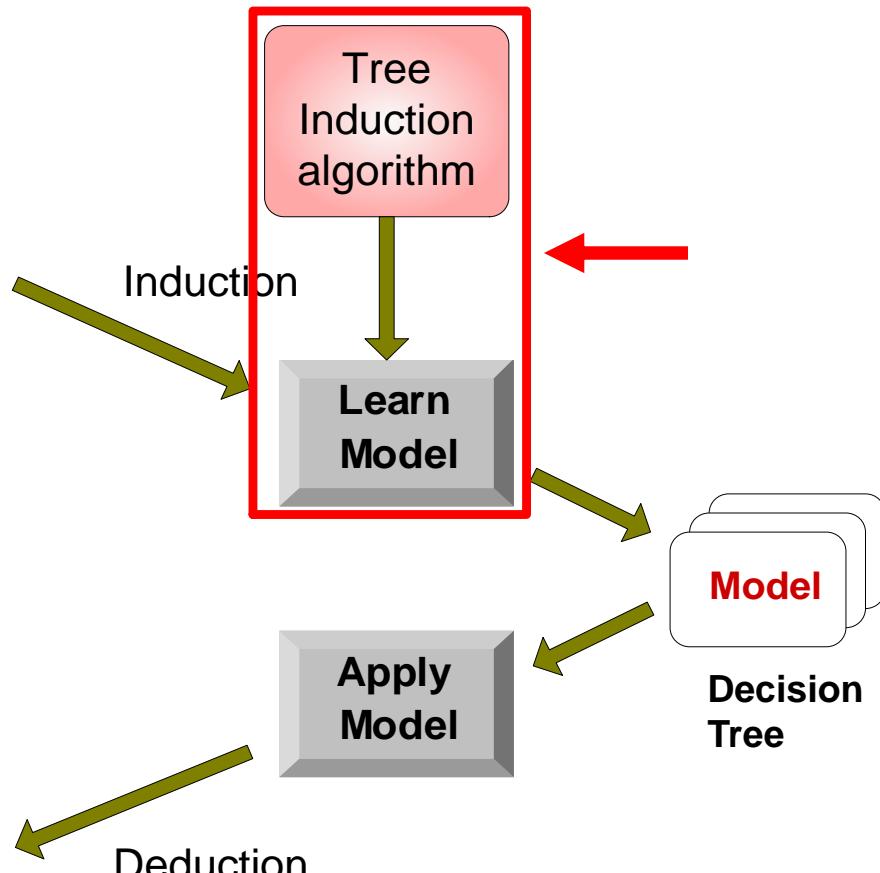
# Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



# How to Build a Decision Tree

---

- Prensipte, belirli bir özellik kümelerinden oluşturulabilen üstel olarak çok sayıda karar ağacı vardır.
- Ağaçların bazıları diğerlerinden daha doğru olsa da, **en uygun ağacı bulmak, arama uzayının üstel boyutu nedeniyle hesaplama açısından mümkün değildir.**
- Bununla birlikte, **verimli algoritmalar makul bir süre içinde** (optimal olmasa da) makul ölçüde doğru bir karar ağacı oluşturmak için geliştirilmiştir.
- Bu algoritmalar genellikle **verileri bölgülemek için hangi özniteliğin** kullanılacağına dair bir dizi yerel olarak optimum kararlar alarak bir karar ağacı oluşturan açgözlü bir strateji (**greedy strategy**) kullanır.

# Decision Tree Induction

---

- Many Algorithms:
  - Hunt's Algorithm (one of the earliest)
  - CART
  - ID3, C4.5
  - SLIQ, SPRINT

# General Structure of Hunt's Algorithm

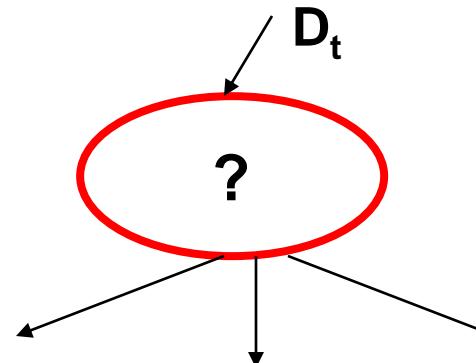
---

- Hunt'ın algoritmasında, eğitim kayıtlarını ardışık olarak daha saf alt kümelere (**purer subsets**) bölgerek yinelemeli bir şekilde (**in a recursive fashion**) bir karar ağacı büyütülür.
  - $D_t$ ,  $t$  düğümü ile ilişkili eğitim kayıtları kümesi olsun ve
  - $y = \{y_1, y_2, \dots, y_C\}$  sınıf etiketleri olsun. Aşağıda, Hunt algoritmasının yinelemeli bir tanımı bulunmaktadır.

# General Structure of Hunt's Algorithm

- |  $D_t$ , t düşümüne ulaşan eğitim kayıtları kümesi olsun
- | Genel Prosedür:
  - Eğer  $D_t$ , aynı sınıfı ( $y_t$ ) ait kayıtları içeriyorsa  $t$ ,  $y_t$  olarak etiketlenmiş bir yaprak düşümdür (leaf node).
  - Eğer  $D_t$  birden fazla sınıfı ait kayıtlar içeriyorsa, verileri daha küçük alt kümelere (oluşturulan alt düşümler- **child nodes** ) bölmek için bir öznitelik testi kullanın. **Prosedürü her alt kümeye yinelemeli olarak uygulayın.**

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



# Hunt's Algorithm

Defaulted = No

(7,3)

(a)

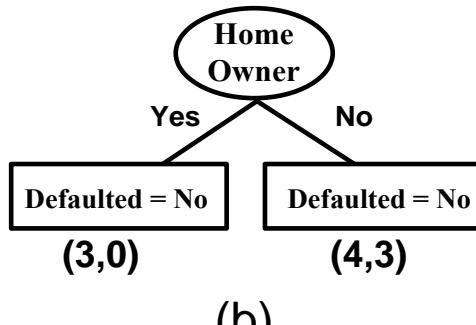
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

# Hunt's Algorithm

Defaulted = No

(7,3)

(a)



(b)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

# Hunt's Algorithm

Defaulted = No

(7,3)

(a)

Home Owner

Yes

No

Defaulted = No

(3,0)

Defaulted = No

(4,3)

(b)

Home Owner

Yes

No

Defaulted = No

(3,0) Single,  
Divorced

Marital Status

Married

Defaulted = Yes

(1,3)

Defaulted = No

(3,0)

(c)

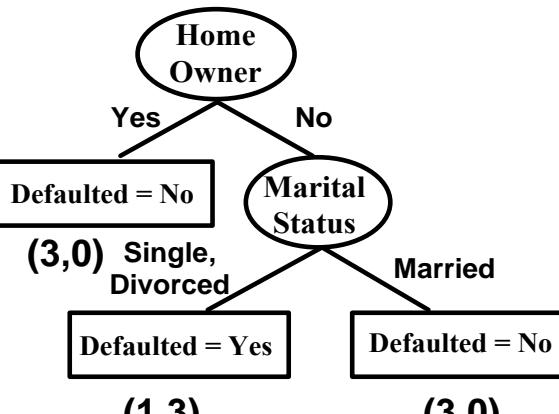
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

# Hunt's Algorithm

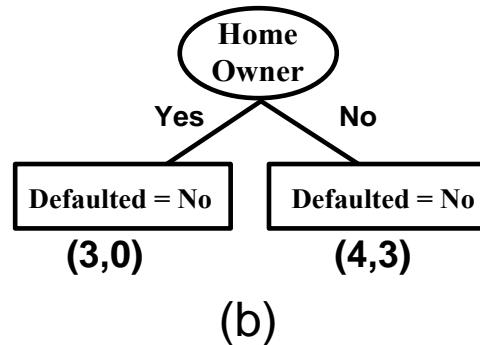
Defaulted = No

(7,3)

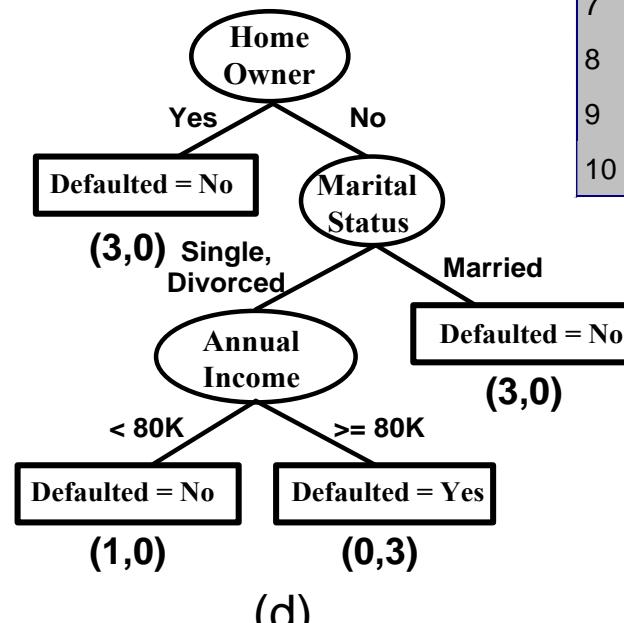
(a)



(c)



(b)



(d)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

# Design Issues of Decision Tree Induction

---

- | Eğitimdataları/kayıtları nasıl bölünmeli?
  - ◆ öznitelik türlerine bağlı olarak
  - Bir test koşulunun iyiliğini (*goodness*) değerlendirmek için ölçüt
- | Bölme prosedürü (*splitting procedure*) nasıl durdurulmalı?
  - Tüm kayıtlar aynı sınıfı aitse veya aynı öznitelik değerlerine sahipse bölmeyi durdurun
  - Erken sonlandırma (*early termination*)

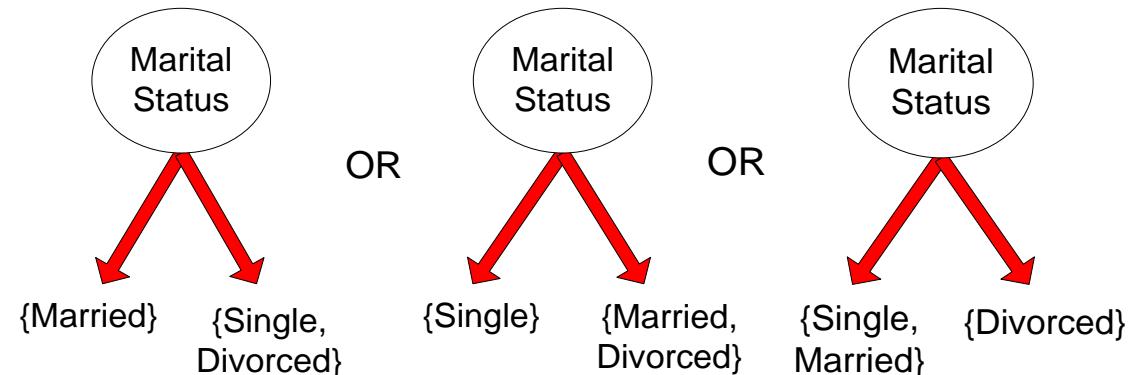
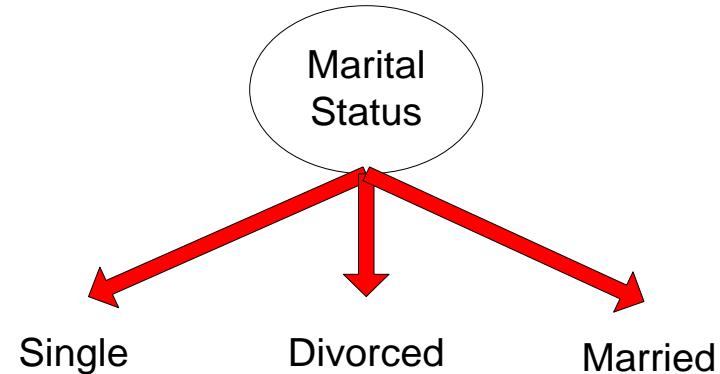
# Methods for Expressing Test Conditions

---

- | Depends on attribute types
  - Binary
  - Nominal
  - Ordinal
  - Continuous
- | Depends on number of ways to split
  - 2-way split
  - Multi-way split

# Test Condition for Nominal Attributes

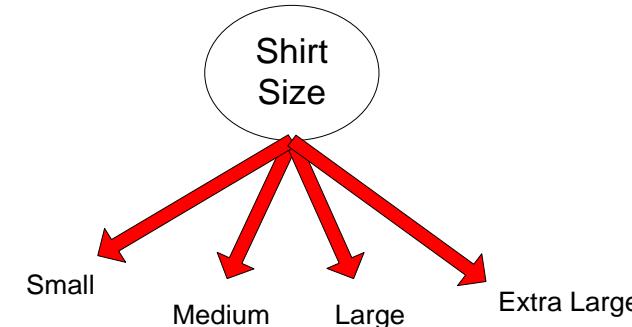
- Multi-way split:
  - Farklı değerlerin (*distinct values*) sayısı kadar bölüm kullanır
- Binary split:
  - Değerleri iki alt gruba ayırır



# Test Condition for Ordinal Attributes

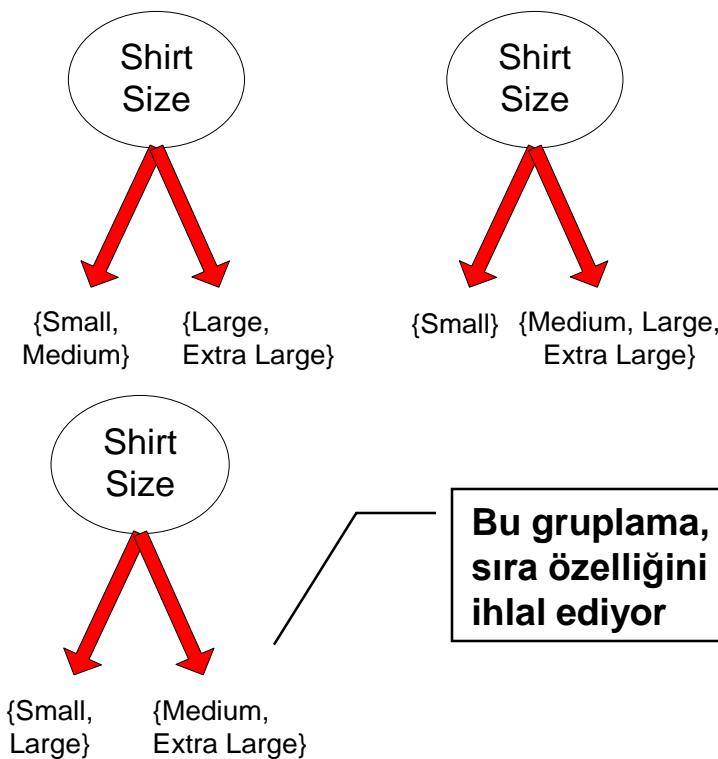
## | Multi-way split:

- Farklı değerlerin (*distinct values*) sayısı kadar bölüm kullanır

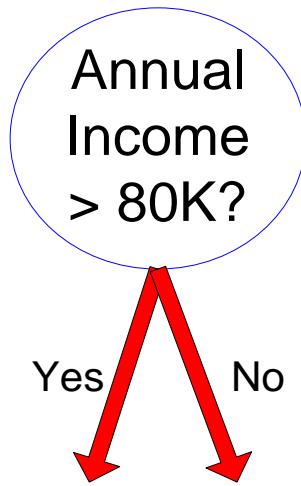


## | Binary split:

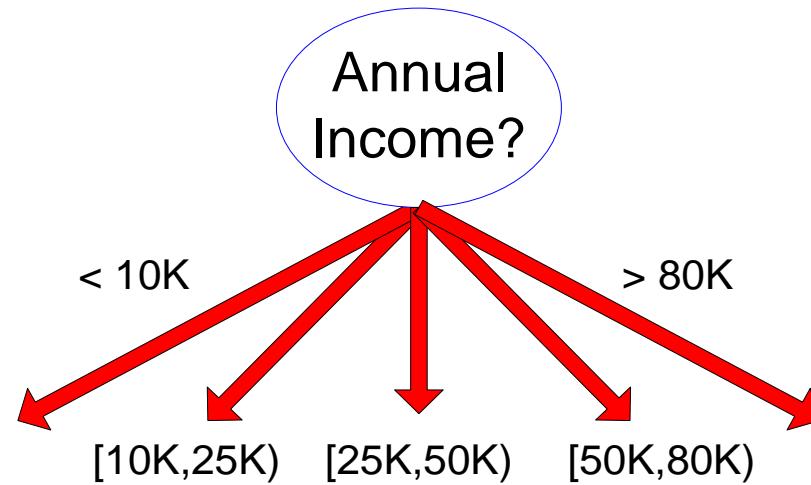
- Değerleri iki alt gruba ayırır
- Öznitelik değerleri arasında sıra özelliğini (*order property*) korunur



# Test Condition for Continuous Attributes



(i) Binary split



(ii) Multi-way split

# Splitting Based on Continuous Attributes

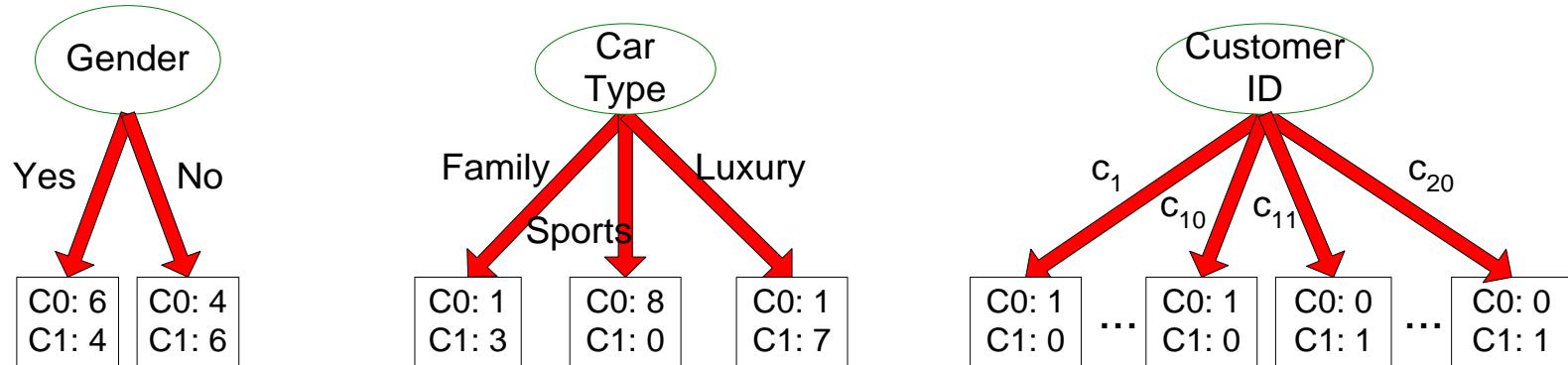
---

- Farklı şekillerde ele alınabilir
  - Sıralı bir kategorik öznitelik (*ordinal categorical attribute*) oluşturmak için ayrıklaştırma (**Discretization**)  
Aralıklar (*ranges*), eşit aralıklı kümeleme, eşit sıklıkta kümeleme (yüzdelikler) veya kümeleme ile bulunabilir.
    - ◆ Static – discretize once at the beginning
    - ◆ Dynamic – repeat at each node
  - **Binary Decision:**  $(A < v)$  or  $(A \geq v)$ 
    - ◆ olası tüm bölümlemeleri düşünüp en iyi kesimi bulur
    - ◆ daha yoğun işlem gerektirebilir

# How to determine the Best Split

**Before Splitting:** 10 records of class 0,  
10 records of class 1

Customer Id	Gender	Car Type	Shirt Size	Class
1	M	Family	Small	C0
2	M	Sports	Medium	C0
3	M	Sports	Medium	C0
4	M	Sports	Large	C0
5	M	Sports	Extra Large	C0
6	M	Sports	Extra Large	C0
7	F	Sports	Small	C0
8	F	Sports	Small	C0
9	F	Sports	Medium	C0
10	F	Luxury	Large	C0
11	M	Family	Large	C1
12	M	Family	Extra Large	C1
13	M	Family	Medium	C1
14	M	Luxury	Extra Large	C1
15	F	Luxury	Small	C1
16	F	Luxury	Small	C1
17	F	Luxury	Medium	C1
18	F	Luxury	Medium	C1
19	F	Luxury	Medium	C1
20	F	Luxury	Large	C1



**Which test condition is the best?**

# How to determine the Best Split

---

- | Greedy approach:
  - Nodes with **purer** class distribution are preferred
- | Need a measure of node impurity:

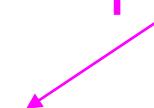
C0: 5
C1: 5

High degree of impurity

C0: 9
C1: 1

Low degree of impurity

**purer**



# Measures of Node Impurity

---

---

## | Gini Index

$$GINI(t) = 1 - \sum_j [p(j | t)]^2$$

## | Entropy

$$Entropy(t) = -\sum_j p(j | t) \log p(j | t)$$

## | Misclassification error

$$Error(t) = 1 - \max_i P(i | t)$$

# Finding the Best Split

---

1. Compute impurity measure ( $P$ ) before splitting
2. Compute impurity measure ( $M$ ) after splitting
  - | Compute impurity measure of each child node
  - |  $M$  is the weighted impurity of children
3. Choose the attribute test condition that produces the **highest gain**

$$\text{Gain} = P - M$$

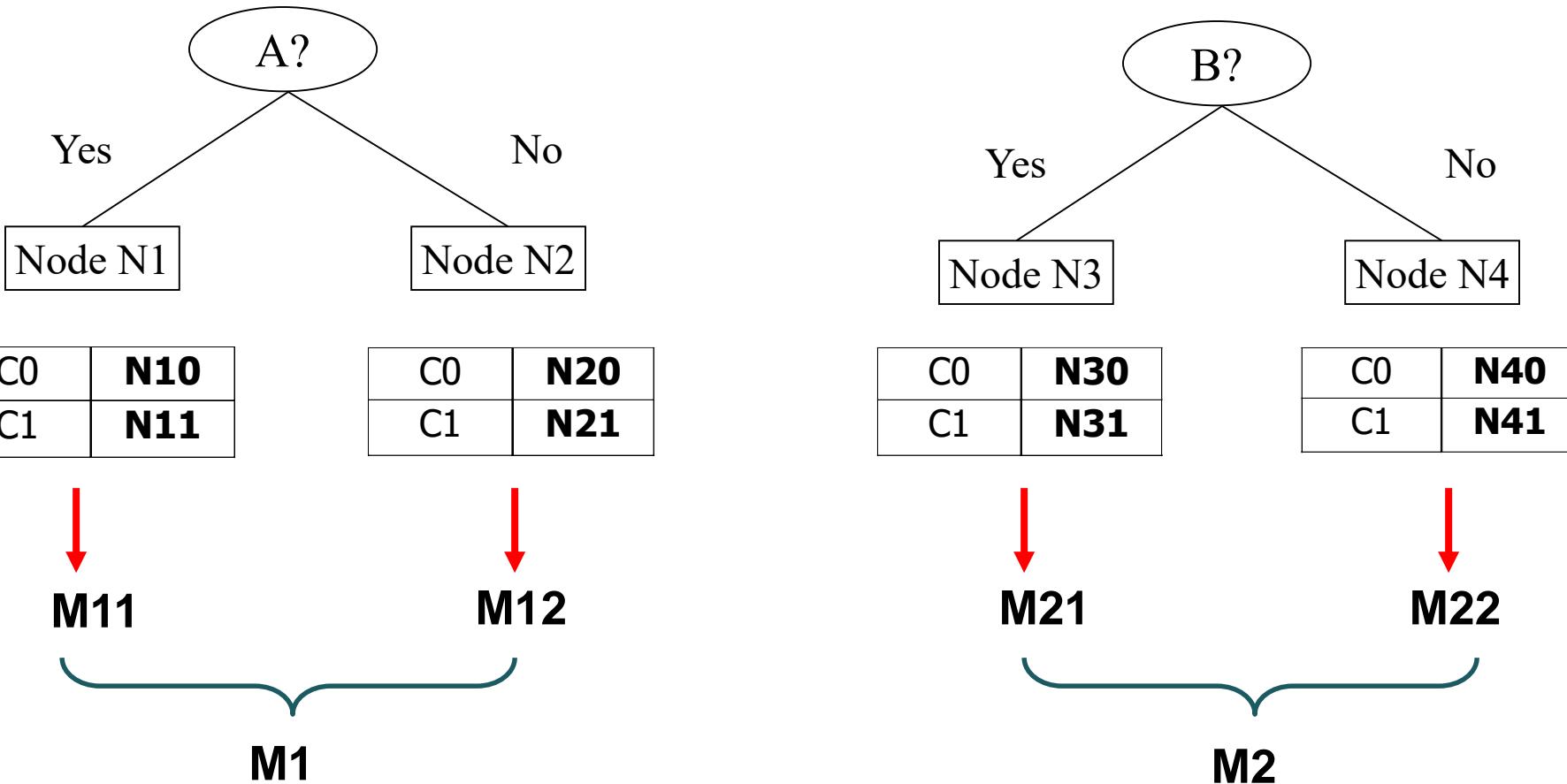
or equivalently, **lowest impurity** measure after splitting ( $M$ )

# Finding the Best Split

Before Splitting:

C0	<b>N00</b>
C1	<b>N01</b>

→ P



$$\text{Gain} = P - M_1 \quad \text{vs} \quad P - M_2$$

# Measure of Impurity: GINI

---

- Gini Index for a given node t :

$$GINI(t) = 1 - \sum_j [p(j | t)]^2$$

(NOTE:  $p(j / t)$  is the relative frequency of class j at node t).

- **Maximum** ( $1 - 1/n_c$ ) when records are **equally distributed** among all classes, implying **least interesting information**
- **Minimum** (0.0) when all records belong to **one class**, implying **most interesting information**

# Measure of Impurity: GINI

- Gini Index for a given node t :

$$GINI(t) = 1 - \sum_j [p(j | t)]^2$$

(NOTE:  $p(j | t)$  is the relative frequency of class j at node t).

- For 2-class problem (p, 1 – p):
  - ◆  $GINI = 1 - p^2 - (1 - p)^2 = 2p(1-p)$

C1	<b>0</b>
C2	<b>6</b>
<b>Gini=0.000</b>	

C1	<b>1</b>
C2	<b>5</b>
<b>Gini=0.278</b>	

C1	<b>2</b>
C2	<b>4</b>
<b>Gini=0.444</b>	

C1	<b>3</b>
C2	<b>3</b>
<b>Gini=0.500</b>	

# Computing Gini Index of a Single Node

---

$$GINI(t) = 1 - \sum_j [p(j | t)]^2$$

C1	<b>0</b>
C2	<b>6</b>

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$\text{Gini} = 1 - P(C1)^2 - P(C2)^2 = 1 - 0 - 1 = 0$$

C1	<b>1</b>
C2	<b>5</b>

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

$$\text{Gini} = 1 - (1/6)^2 - (5/6)^2 = 0.278$$

C1	<b>2</b>
C2	<b>4</b>

$$P(C1) = 2/6 \quad P(C2) = 4/6$$

$$\text{Gini} = 1 - (2/6)^2 - (4/6)^2 = 0.444$$

# Computing Gini Index for a Collection of Nodes

- | When a node p is split into k partitions (children)

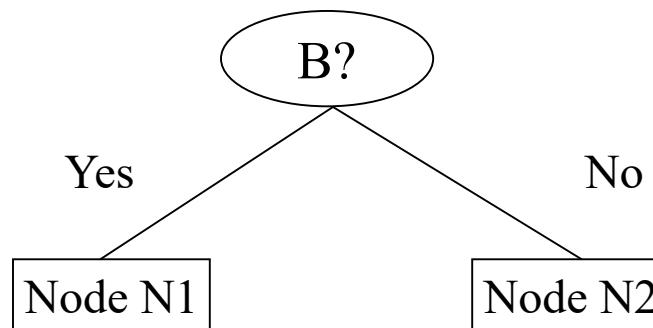
$$GINI_{split} = \sum_{i=1}^k \frac{n_i}{n} GINI(i)$$

where,       $n_i$  = number of records at child i,  
                 $n$  = number of records at parent node p.

- | Choose the attribute that **minimizes weighted average Gini index of the children**
- | Gini index is used in decision tree algorithms such as CART, SLIQ, SPRINT

# Binary Attributes: Computing GINI Index

- Splits into two partitions
- Effect of Weighing partitions:
  - **Larger and Purer Partitions** are sought for.



**Gini(N1)**

$$= 1 - (5/6)^2 - (1/6)^2 \\ = 0.278$$

**Gini(N2)**

$$= 1 - (2/6)^2 - (4/6)^2 \\ = 0.444$$

	<b>N1</b>	<b>N2</b>
C1	5	2
C2	1	4
<b>Gini=0.361</b>		

	<b>Parent</b>
C1	7
C2	5
<b>Gini = 0.486</b>	

**Weighted Gini of N1 N2**

$$= 6/12 * 0.278 + \\ 6/12 * 0.444 \\ = 0.361$$

$$\text{Gain} = 0.486 - 0.361 = 0.125$$

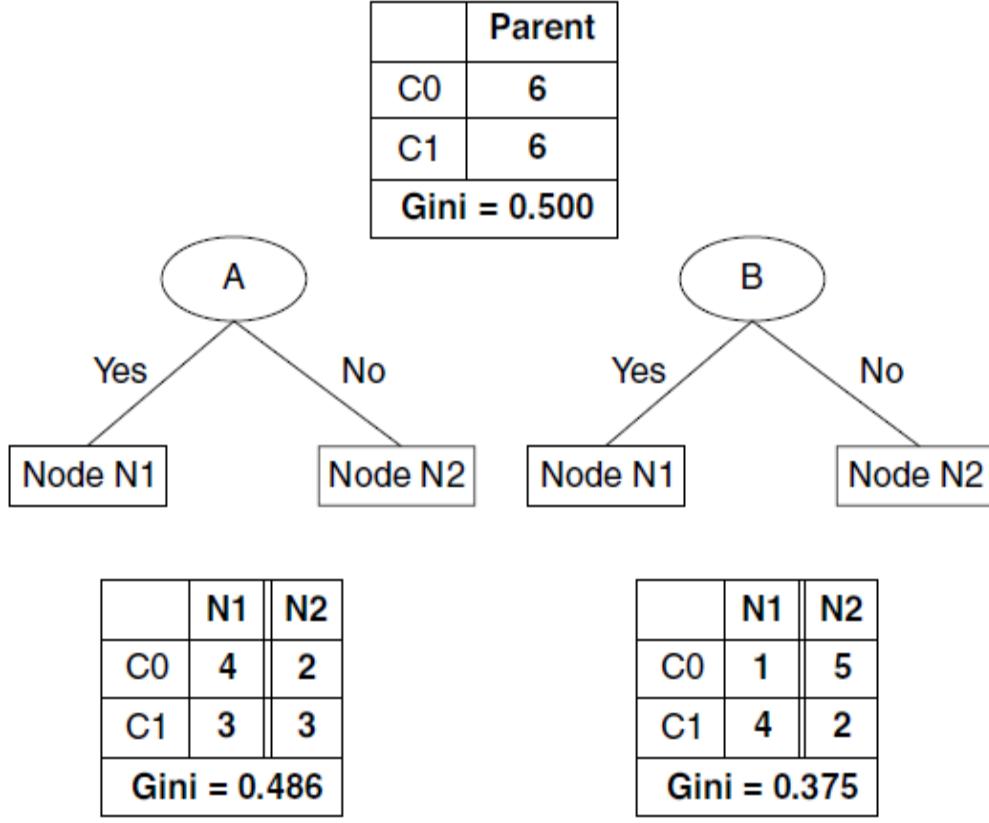


Figure 4.14. Splitting binary attributes.

- Bölünmeden önce, her iki sınıfın eşit sayıda kayıt olduğundan Gini indeksi 0.5'tir.
- Verileri bölmek için A özniteliği seçilirse, N1 düğümü için Gini indeksi 0,490 ve N2 düğümü için 0,480'dir.
- Alt düğümler için Gini indeksinin ağırlıklı ortalaması  $(7/12) \times 0.4898 + (5/12) \times 0.480 = 0.486$ .
- Benzer şekilde, B özniteliği için Gini indeksinin ağırlıklı ortalamasının 0,375 olduğunu gösterebiliriz.
- B özniteliğinin alt kümeleri (*subsets*) **daha küçük bir Gini indeksine sahip** olduğundan, **A özniteliği yerine tercih edilir.**

# Categorical Attributes: Computing Gini Index

- | Her farklı değer için, veri kümesindeki her bir sınıfın sayılarını toplayın
- | Karar vermek için sayım matrisini (*count matrix*) kullanın

Multi-way split

CarType			
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini	0.163		

Two-way split

(find best partition of values)

CarType		
	{Sports, Luxury}	{Family}
C1	9	1
C2	7	3
Gini	0.468	

CarType		
	{Sports}	{Family, Luxury}
C1	8	2
C2	0	10
Gini	0.167	

«Multiway split», her iki «Two-way split» kıyasla daha küçük bir Gini indeksine sahiptir.

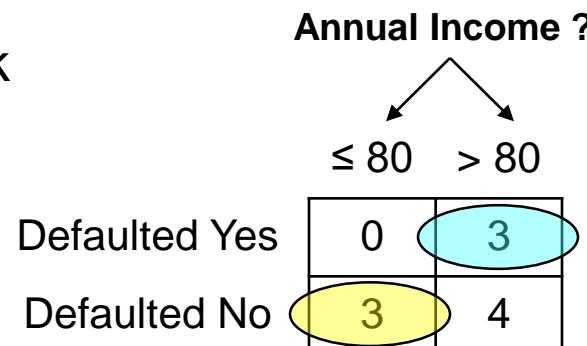
Which of these is the best?

Üçüncü grulamanın daha düşük bir Gini indeksi vardır çünkü karşılık gelen alt kümeleri çok daha saftır.

# Continuous Attributes: Computing Gini Index

- | Tek bir değere dayalı İkili Kararlar (*Binary Decisions*) kullanın
- | Bölme değeri (*splitting value*) için birçok seçenek
  - Olası bölme değerlerinin sayısı = Farklı değerlerin sayısı
- | Her bölme değerinin kendisiyle ilişkili bir sayı matrisi vardır
  - Her bölümdeki (partitions) sınıf sayıları,  $A < v$  ve  $A \geq v$
- | En iyi  $v$ 'yi seçmek için basit yöntem
  - Her  $v$  için, sayı matrisini toplamak ve Gini indeksini hesaplamak için veritabanını tarayın
  - Computationally Inefficient!  
Repetition of work.

ID	Home Owner	Marital Status	Annual Income	Defaulted
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



# Continuous Attributes: Computing Gini Index...

- Verimli hesaplama için: her öznitelik için,
  - Öznitelik değerlerini sıralayın
  - Her seferinde sayımları matrisini güncelleyerek ve gini indeksini hesaplayarak bu değerleri doğrusal olarak tarayın
  - En düşük gini indeksine sahip bölme konumu (*split position*)** seçin

Defaulted	No	No	No	Yes	Yes	Yes	No	No	No	No
Annual Income										
Sorted Values	60	70	75	85	90	95	100	120	125	220

# Continuous Attributes: Computing Gini Index...

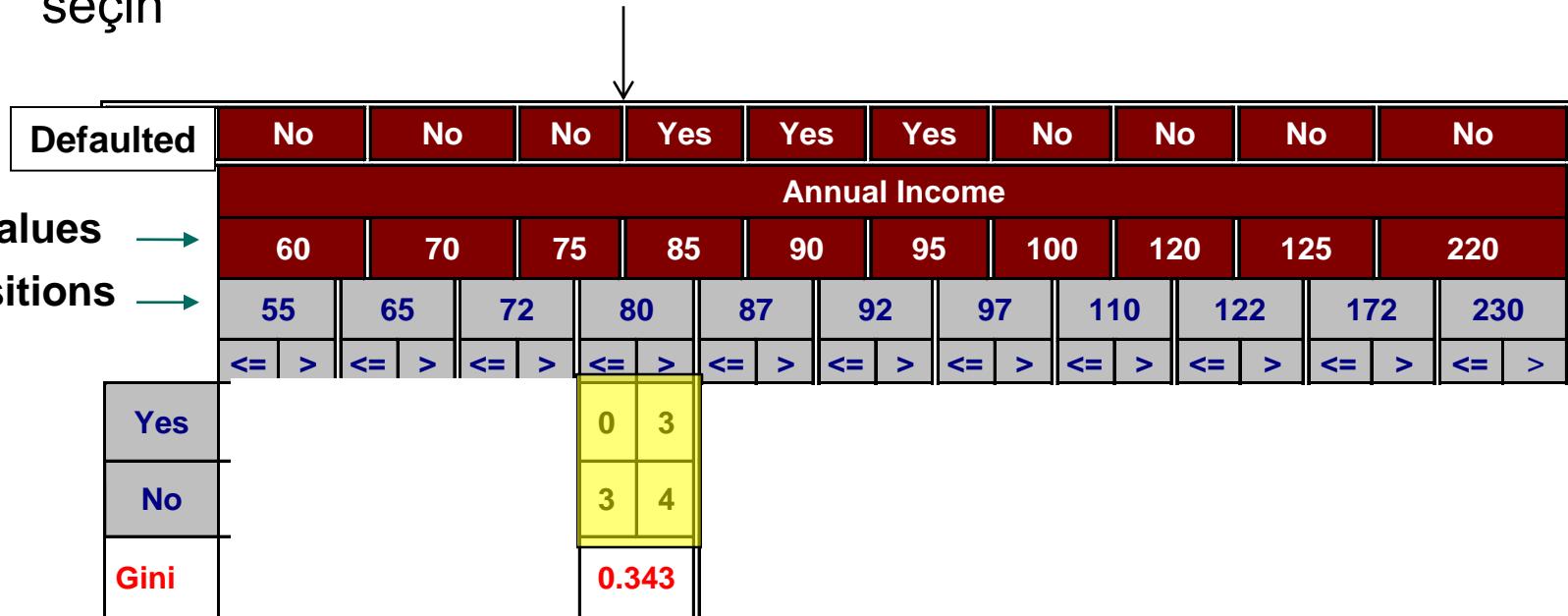
- Verimli hesaplama için: her öznitelik için,
  - Öznitelik değerlerini sıralayın
  - Her seferinde sayım matrisini güncelleyerek ve gini indeksini hesaplayarak bu değerleri doğrusal olarak tarayın
  - En düşük gini indeksine sahip bölme konumu (*split position*) seçin**

Defaulted	No	No	No	Yes	Yes	Yes	No	No	No	No	
Annual Income											
Sorted Values →	60	70	75	85	90	95	100	120	125	220	
Split Positions →	55	65	72	80	87	92	97	110	122	172	230
	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	

Candidate split positions are identified by taking the **midpoints** between two adjacent sorted values: 55, 65, 72, and so on.

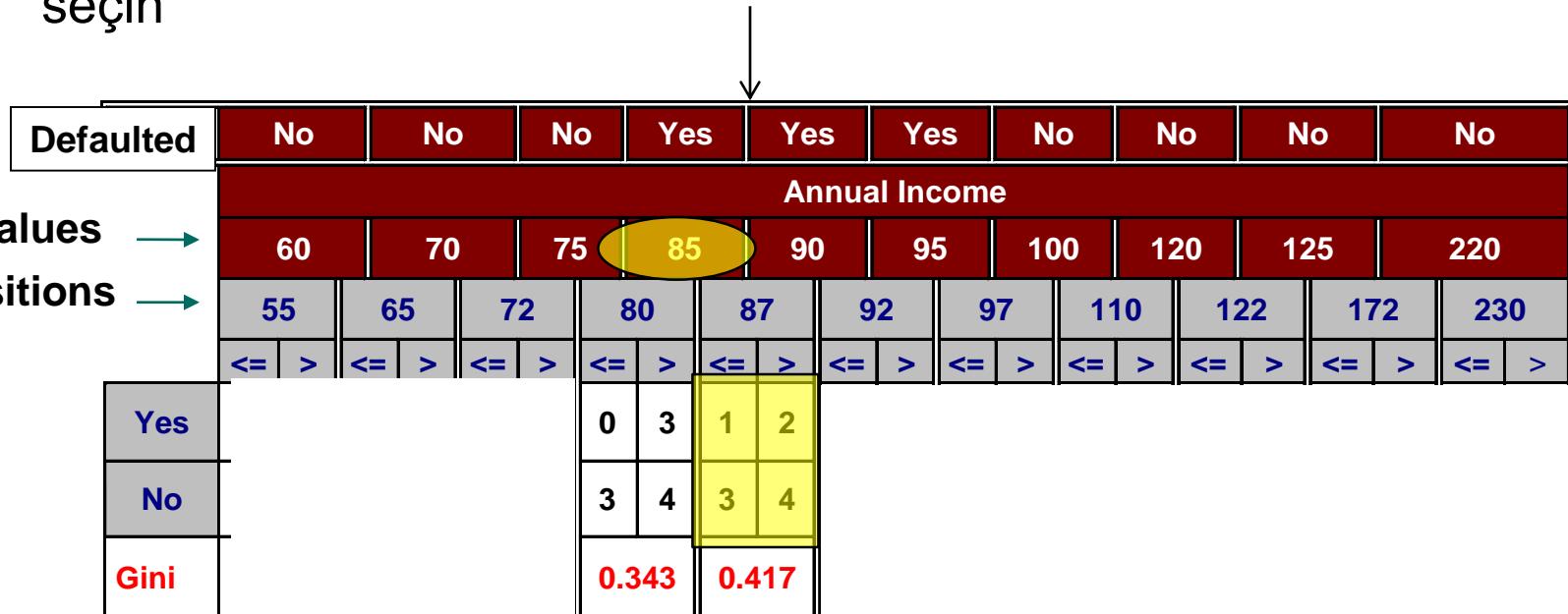
# Continuous Attributes: Computing Gini Index...

- Verimli hesaplama için: her öznitelik için,
  - Öznitelik değerlerini sıralayın
  - Her seferinde sayım matrisini güncelleyerek ve gini indeksini hesaplayarak bu değerleri doğrusal olarak tarayın
  - En düşük gini indeksine** sahip bölme konumu (*split position*) seçin



# Continuous Attributes: Computing Gini Index...

- Verimli hesaplama için: her öznitelik için,
  - Öznitelik değerlerini sıralayın
  - Her seferinde sayımları matrisini güncelleştirerek ve gini indeksini hesaplayarak bu değerleri doğrusal olarak tarayın
  - En düşük gini indeksine sahip bölme konumu (*split position*)** seçin



# Continuous Attributes: Computing Gini Index...

- Verimli hesaplama için: her öznitelik için,
  - Öznitelik değerlerini sıralayın
  - Her seferinde sayımları matrisini güncelleyerek ve gini indeksini hesaplayarak bu değerleri doğrusal olarak tarayın
  - En düşük gini indeksine sahip bölme konumu (*split position*)** seçin

Defaulted	No	No	No	Yes	Yes	Yes	No	No	No	No	
Annual Income											
Sorted Values →	60	70	75	85	90	95	100	120	125	220	
Split Positions →	55	65	72	80	87	92	97	110	122	172	230
	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	
Yes	0 3	0 3	0 3	0 3	1 2	2 1	3 0	3 0	3 0	3 0	
No	0 7	1 6	2 5	3 4	3 4	3 4	3 4	4 3	5 2	6 1	
Gini	0.420	0.400	0.375	0.343	0.417	0.400	0.300	0.343	0.375	0.400	

# Measure of Impurity: Entropy

---

- | Entropy at a given node t:

$$Entropy(t) = -\sum_j p(j | t) \log p(j | t)$$

(NOTE:  $p(j / t)$  is the relative frequency of class j at node t).

- ◆ **Maximum** ( $\log n_c$ ) when records are **equally distributed** among all classes **implying least information**
- ◆ **Minimum** (0.0) when all records belong to **one class**, **implying most information**

- Entropy based computations are quite similar to the GINI index computations

# Computing Entropy of a Single Node

$$Entropy(t) = -\sum_j p(j | t) \log_2 p(j | t)$$

C1	<b>0</b>
C2	<b>6</b>

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$\text{Entropy} = -0 \log 0 - 1 \log 1 = -0 - 0 = 0$$

C1	<b>1</b>
C2	<b>5</b>

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

$$\text{Entropy} = -(1/6) \log_2 (1/6) - (5/6) \log_2 (5/6) = 0.65$$

C1	<b>2</b>
C2	<b>4</b>

$$P(C1) = 2/6 \quad P(C2) = 4/6$$

$$\text{Entropy} = -(2/6) \log_2 (2/6) - (4/6) \log_2 (4/6) = 0.92$$

# Computing Information Gain After Splitting

---

## | Information Gain:

$$GAIN_{split} = Entropy(p) - \left( \sum_{i=1}^k \frac{n_i}{n} Entropy(i) \right)$$

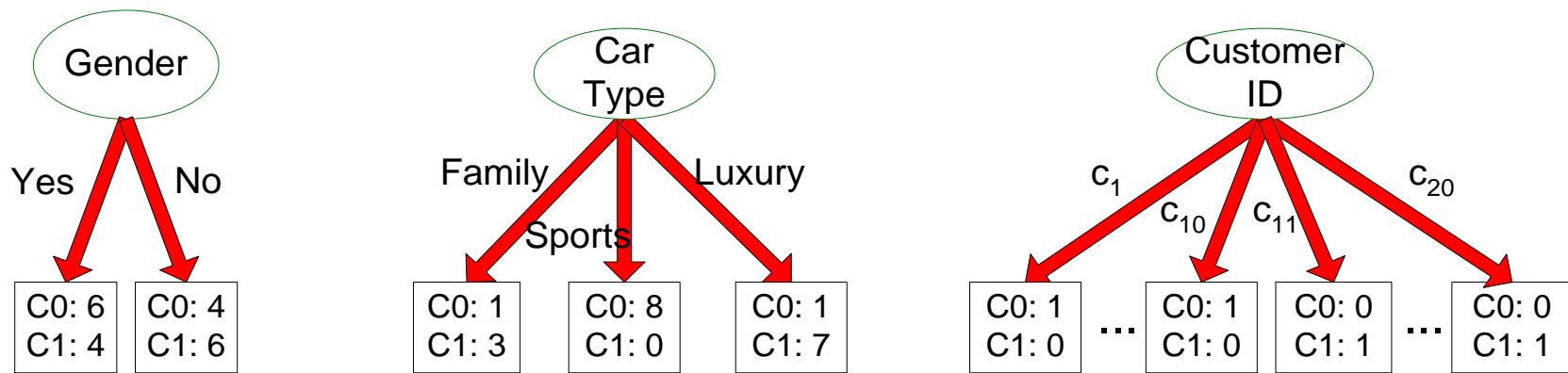
Parent Node, p is split into k partitions;

$n_i$  is number of records in partition i

- Measures **Reduction in Entropy** achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN)
- Used in the ID3 and C4.5 decision tree algorithms

# Problem with large number of partitions

- «Node impurity» ölçütü, her biri küçük ancak saf olan çok sayıda bölümle sonuçlanan bölümlemeleri tercih etme eğilimindedir.



- Customer ID has en yüksek bilgi kazancına (**highest information gain**) sahiptir çünkü tüm çocuklar için entropi sıfırdır

# Gain Ratio

## I Gain Ratio:

$$GainRATIO_{split} = \frac{GAIN_{split}}{SplitINFO}$$

$$SplitINFO = -\sum_{i=1}^k \frac{n_i}{n} \log \frac{n_i}{n}$$

Parent Node, p is split into k partitions

$n_i$  is the number of records in partition i

- Adjusts Information Gain by the entropy of the partitioning (SplitINFO).
  - ◆ Higher entropy partitioning (large number of small partitions) is penalized!
- Used in C4.5 algorithm
- Designed to overcome the disadvantage of Information Gain

# Gain Ratio

| Gain Ratio:

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO}$$

$$SplitINFO = -\sum_{i=1}^k \frac{n_i}{n} \log \frac{n_i}{n}$$

Parent Node, p is split into k partitions

$n_i$  is the number of records in partition i

	CarType		
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini	<b>0.163</b>		

$$\text{SplitINFO} = 1.52$$

	CarType	
	{Sports, Luxury}	{Family}
C1	9	1
C2	7	3
Gini	<b>0.468</b>	

$$\text{SplitINFO} = 0.72$$

	CarType	
	{Sports}	{Family, Luxury}
C1	8	2
C2	0	10
Gini	<b>0.167</b>	

$$\text{SplitINFO} = 0.97$$

$$\text{SplitINFO} = -(16/20) * \log_2(16/20) - (4/20) * \log_2(4/20)$$

$$= 0.72$$

$$SplitINFO = -\sum_{i=1}^k \frac{n_i}{n} \log \frac{n_i}{n}$$

CarType			
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini	<b>0.163</b>		

$$\text{SplitINFO} = 1.52$$

CarType		
	{Sports, Luxury}	{Family}
C1	9	1
C2	7	3
Gini	<b>0.468</b>	

$$\text{SplitINFO} = 0.72$$

CarType		
	{Sports}	{Family, Luxury}
C1	8	2
C2	0	10
Gini	<b>0.167</b>	

$$\text{SplitINFO} = 0.97$$

# Measure of Impurity: Classification Error

---

| Classification error at a node  $t$  :

$$Error(t) = 1 - \max_i P(i | t)$$

- **Maximum** ( $1 - 1/n_c$ ) when records are **equally distributed** among all classes, implying **least interesting information**
- **Minimum** (0) when all records belong to **one class**, implying **most interesting information**

# Computing Error of a Single Node

---

$$Error(t) = 1 - \max_i P(i | t)$$

C1	<b>0</b>
C2	<b>6</b>

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$\text{Error} = 1 - \max(0, 1) = 1 - 1 = 0$$

C1	<b>1</b>
C2	<b>5</b>

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

$$\text{Error} = 1 - \max(1/6, 5/6) = 1 - 5/6 = 1/6$$

C1	<b>2</b>
C2	<b>4</b>

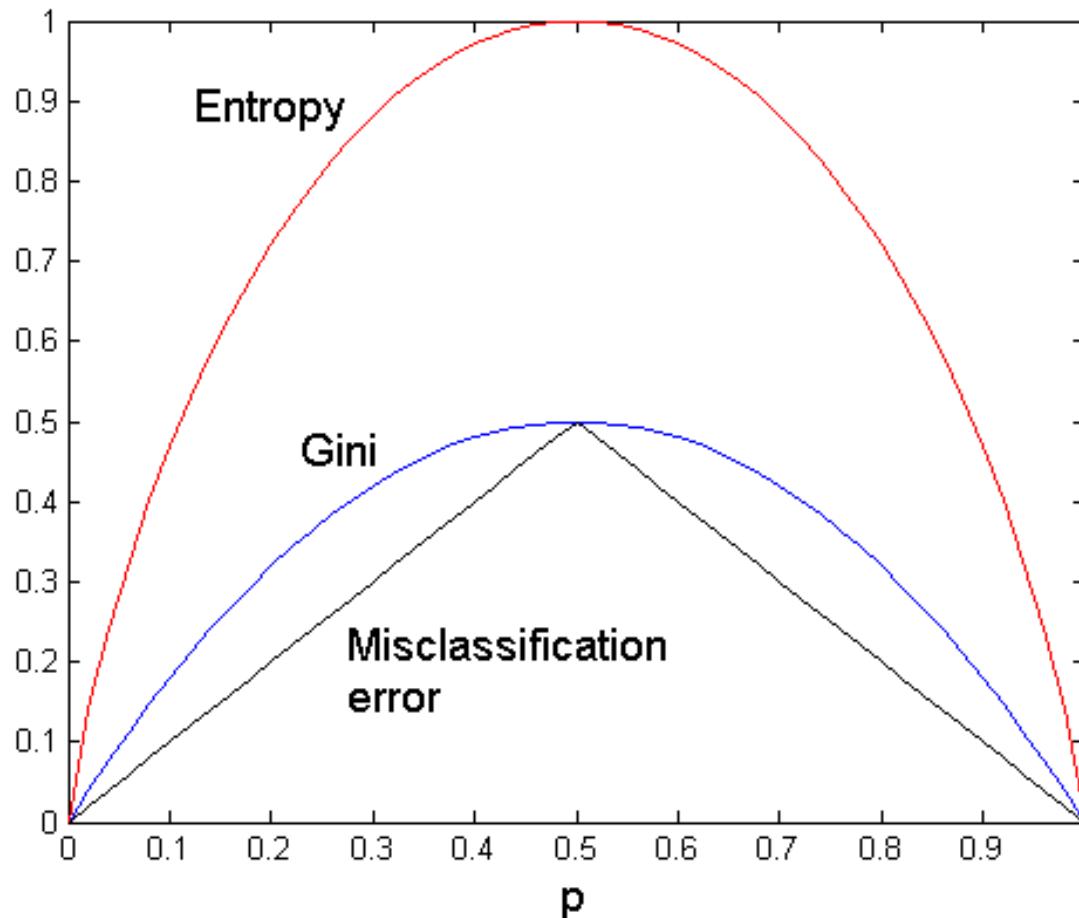
$$P(C1) = 2/6 \quad P(C2) = 4/6$$

$$\text{Error} = 1 - \max(2/6, 4/6) = 1 - 4/6 = 1/3$$

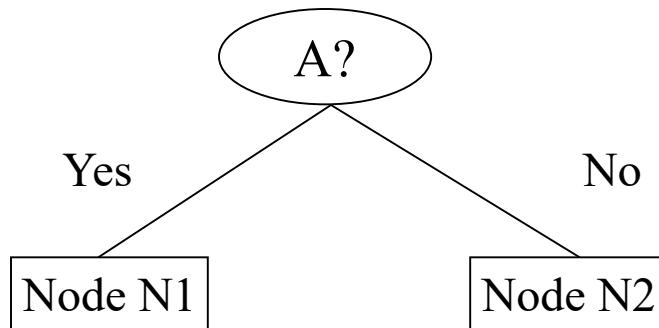
# Comparison among Impurity Measures

---

For a 2-class problem:



# Misclassification Error vs Gini Index



	<b>Parent</b>
C1	<b>7</b>
C2	<b>3</b>
<b>Gini = 0.42</b>	

**Gini(N1)**

$$\begin{aligned} &= 1 - (3/3)^2 - (0/3)^2 \\ &= 0 \end{aligned}$$

**Gini(N2)**

$$\begin{aligned} &= 1 - (4/7)^2 - (3/7)^2 \\ &= 0.489 \end{aligned}$$

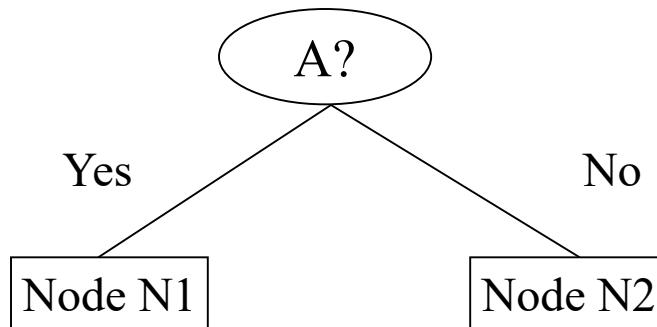
	<b>N1</b>	<b>N2</b>
C1	<b>3</b>	<b>4</b>
C2	<b>0</b>	<b>3</b>
<b>Gini=0.342</b>		

**Gini(Children)**

$$\begin{aligned} &= 3/10 * 0 \\ &+ 7/10 * 0.489 \\ &= 0.342 \end{aligned}$$

**Gini improves but  
error remains the  
same!!**

# Misclassification Error vs Gini Index



	<b>Parent</b>
C1	<b>7</b>
C2	<b>3</b>
<b>Gini = 0.42</b>	

	<b>N1</b>	<b>N2</b>
C1	<b>3</b>	<b>4</b>
C2	<b>0</b>	<b>3</b>
<b>Gini=0.342</b>		

	<b>N1</b>	<b>N2</b>
C1	<b>3</b>	<b>4</b>
C2	<b>1</b>	<b>2</b>
<b>Gini=0.416</b>		

Misclassification error for all three cases = 0.3 !

# Decision Tree Based Classification

---

## | Avantajları:

- İnşa etmesi az zahmetlidir
- Bilinmeyen kayıtları (*unknown records*) sınıflandırmada son derece hızlı
- Küçük boyutlu ağaçlar için yorumlanması kolay
- Gürültüye karşı dayanıklı (özellikle overfitting önleme yöntemleri kullanıldığında)
- Gereksiz veya alakasız öznitelikleri kolayca idare edebilir (öznitelikler birbiriyle etkileşim halinde değilse)

## | Dezavantajları :

- Olası karar aacı çözüm uzayı üstel büyülüktedir. **Greedy** yaklaşımalar çoğu zaman en iyi aacı bulamaz.
- Öznitelikler arasındaki etkileşimleri hesaba katmaz
- Her karar sınırı yalnızca tek bir özniteliği içerir

# Decision Tree Example

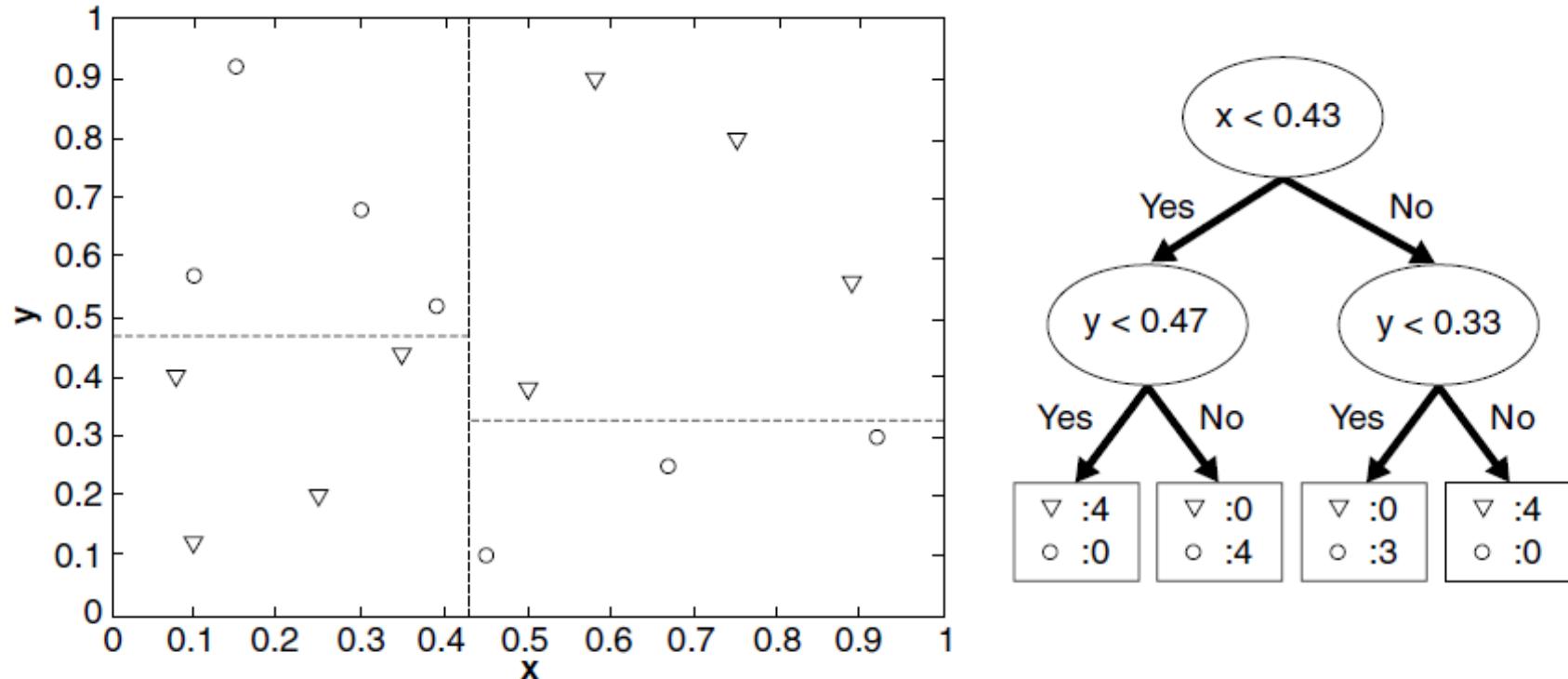


Figure 4.20. Example of a decision tree and its decision boundaries for a two-dimensional data set.

# Decision Tree Example

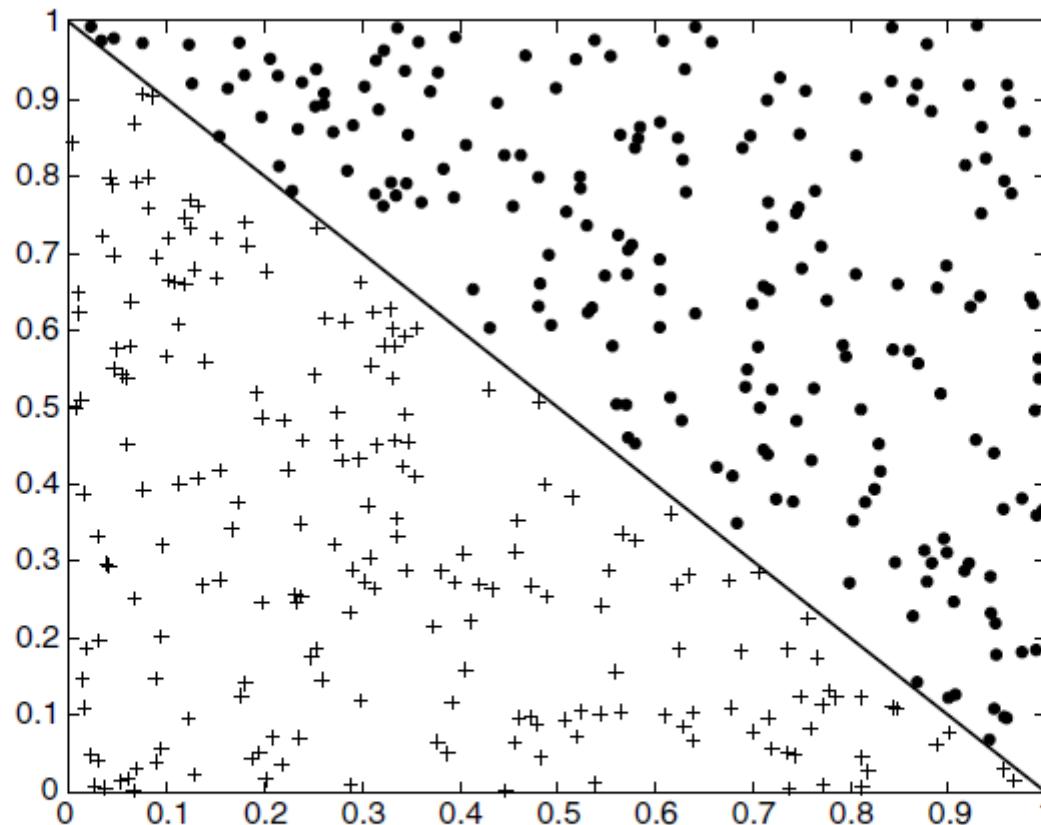
---

Bu bölümde şimdije kadar açıklanan test koşulları, bir seferde yalnızca tek bir özniteliğin kullanılmasını içerir. Sonuç olarak, ağaç büyütme prosedürü, her bölge aynı sınıfın kayıtlarını içерene kadar öznitelik uzayını ayrik bölgelere bölme işlemi olarak görülebilir (bkz. Şekil 4.20).

Farklı sınıflardan iki komşu bölge arasındaki sınır, karar sınırı (**decision boundary**) olarak bilinir. Test koşulu yalnızca tek bir özniteliği içerdiginden, karar sınırları doğrusaldır (**rectilinear**); yani "koordinat eksenlerine" paralel.

Bu, **sürekli özellikler arasındaki karmaşık ilişkileri modellemek** için karar ağaç temsilinin ifade gücünü **sınırlar**. Şekil 4.21, **bir seferde yalnızca tek bir özniteliği** içeren test koşullarını kullanan bir karar ağaç algoritmasıyla etkili bir şekilde sınıflandırılamayan bir veri setini göstermektedir.

# Decision Tree Example



**Figure 4.21.** Example of data set that cannot be partitioned optimally using test conditions involving single attributes.

# Data Mining

---

---

## Model Overfitting

Introduction to Data Mining, 2<sup>nd</sup> Edition

by

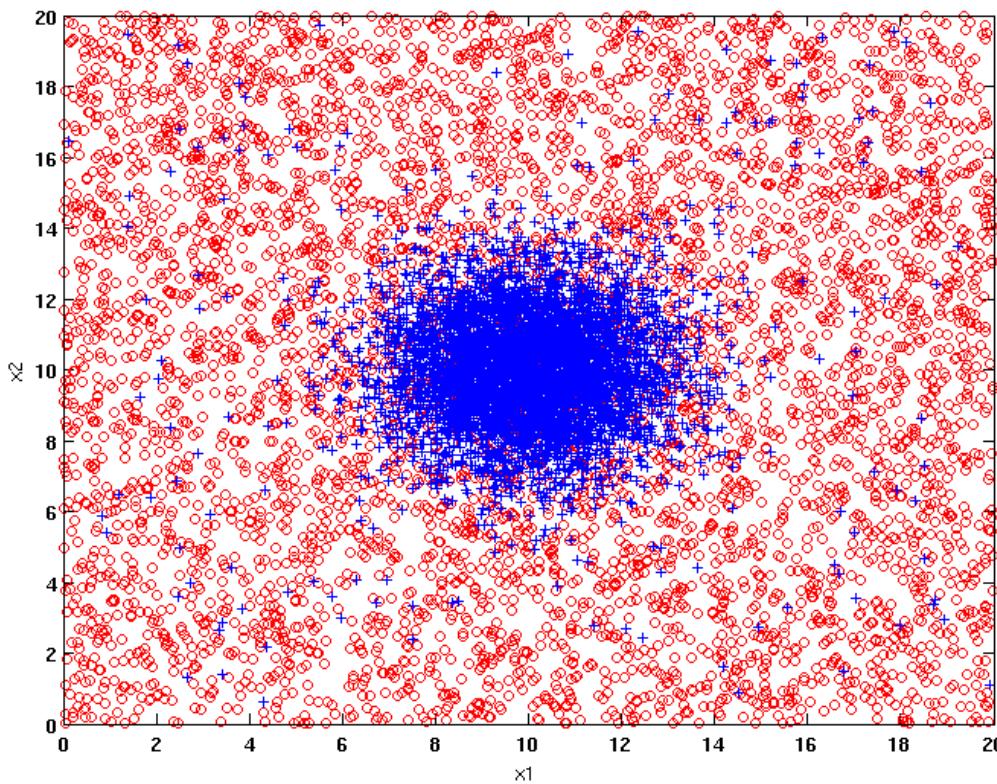
Tan, Steinbach, Karpatne, Kumar

# Classification Errors

---

- Training errors (apparent errors)
  - Eğitim setinde yapılan hatalar
- Test errors
  - Test setinde yapılan hatalar
- Generalization errors
  - Aynı dağılımdan rastgele kayıtların seçimi üzerinden bir modelin beklenen hatası

# Example Data Set



**Two class problem:**

**+ : 5200 instances**

- 5000 instances generated from a Gaussian centered at  $(10,10)$

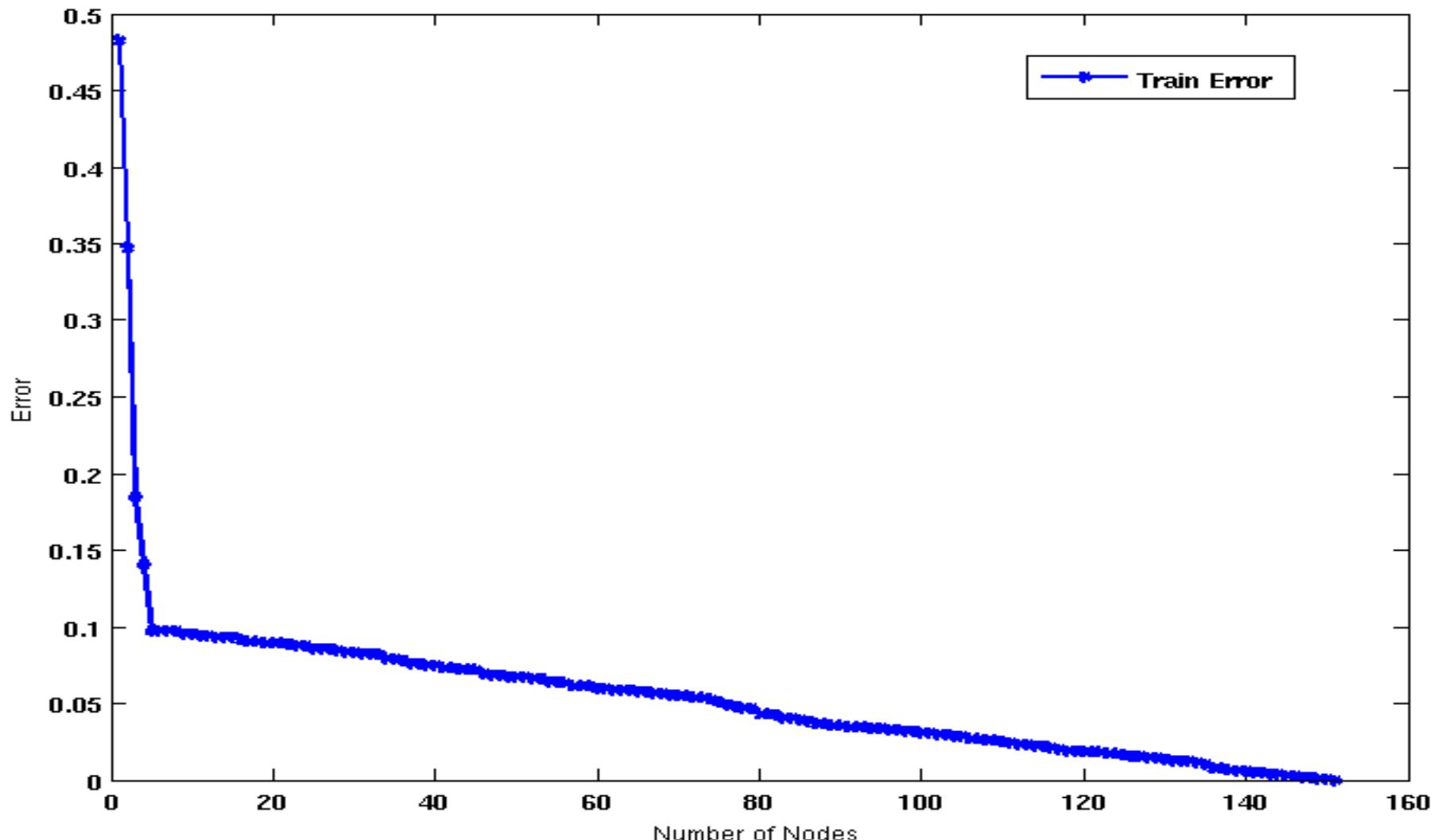
- 200 noisy instances added

**o : 5200 instances**

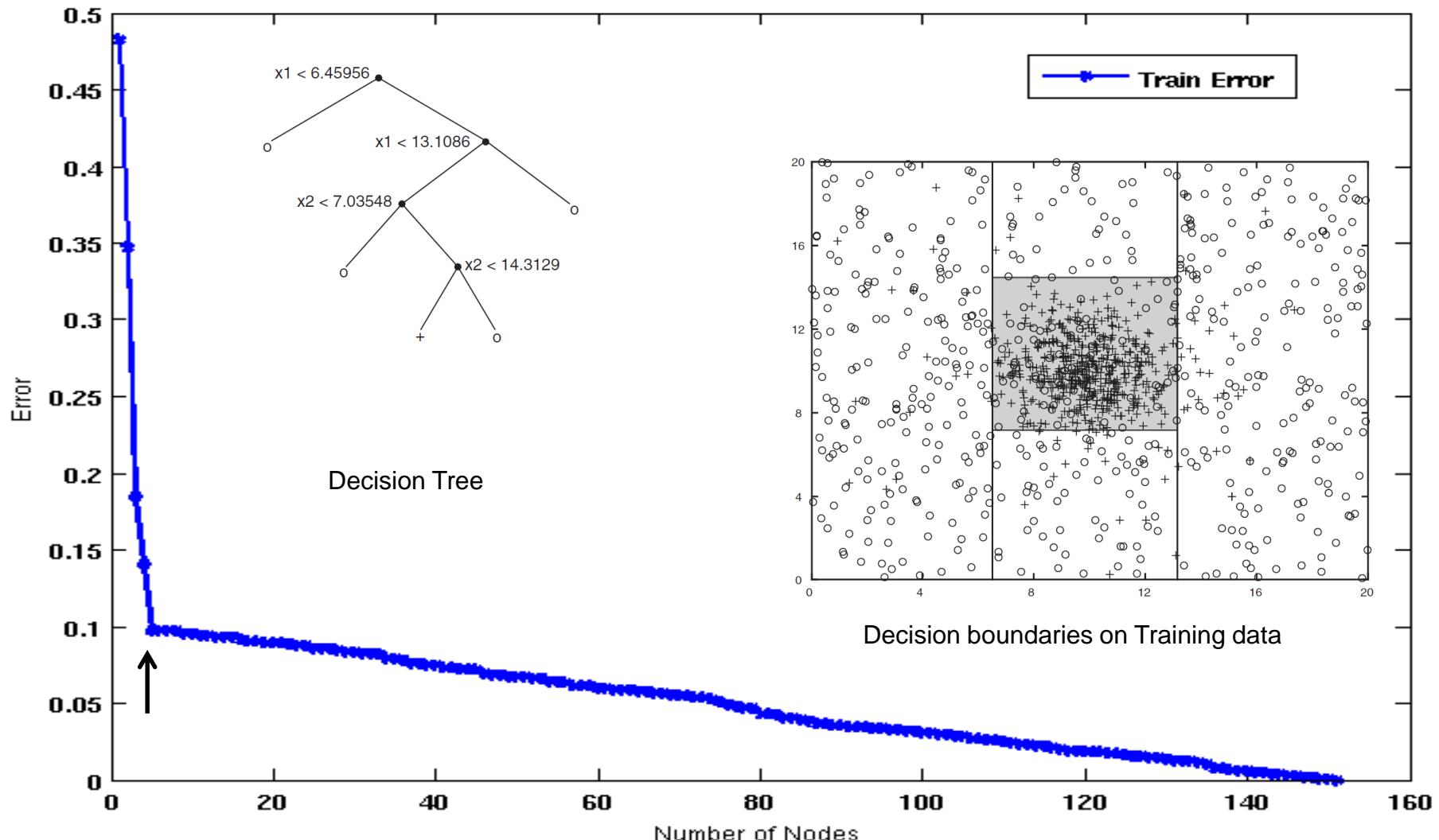
- Generated from a uniform distribution

**10 % of the data used for training and 90% of the data used for testing**

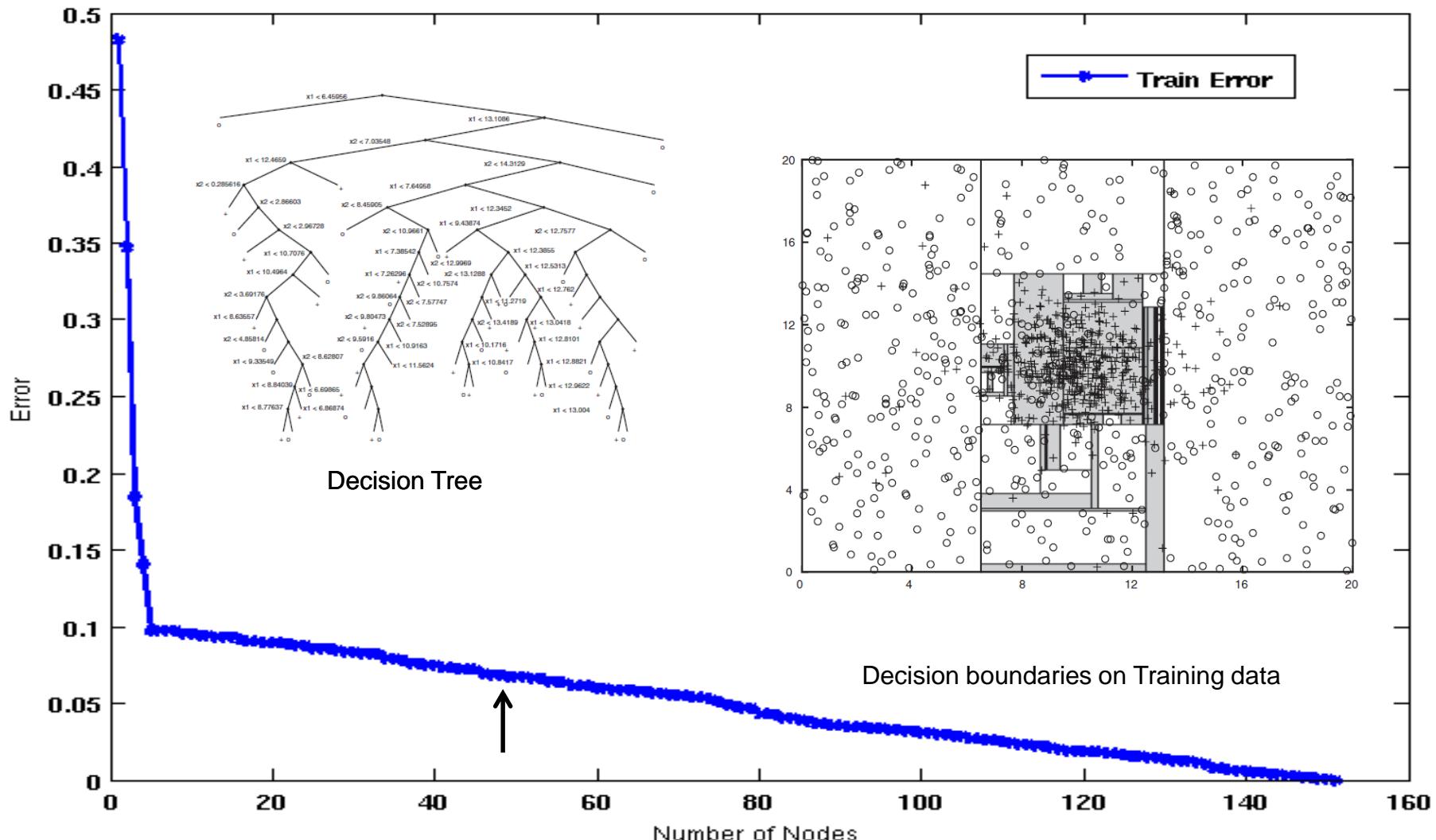
# Increasing number of nodes in Decision Trees



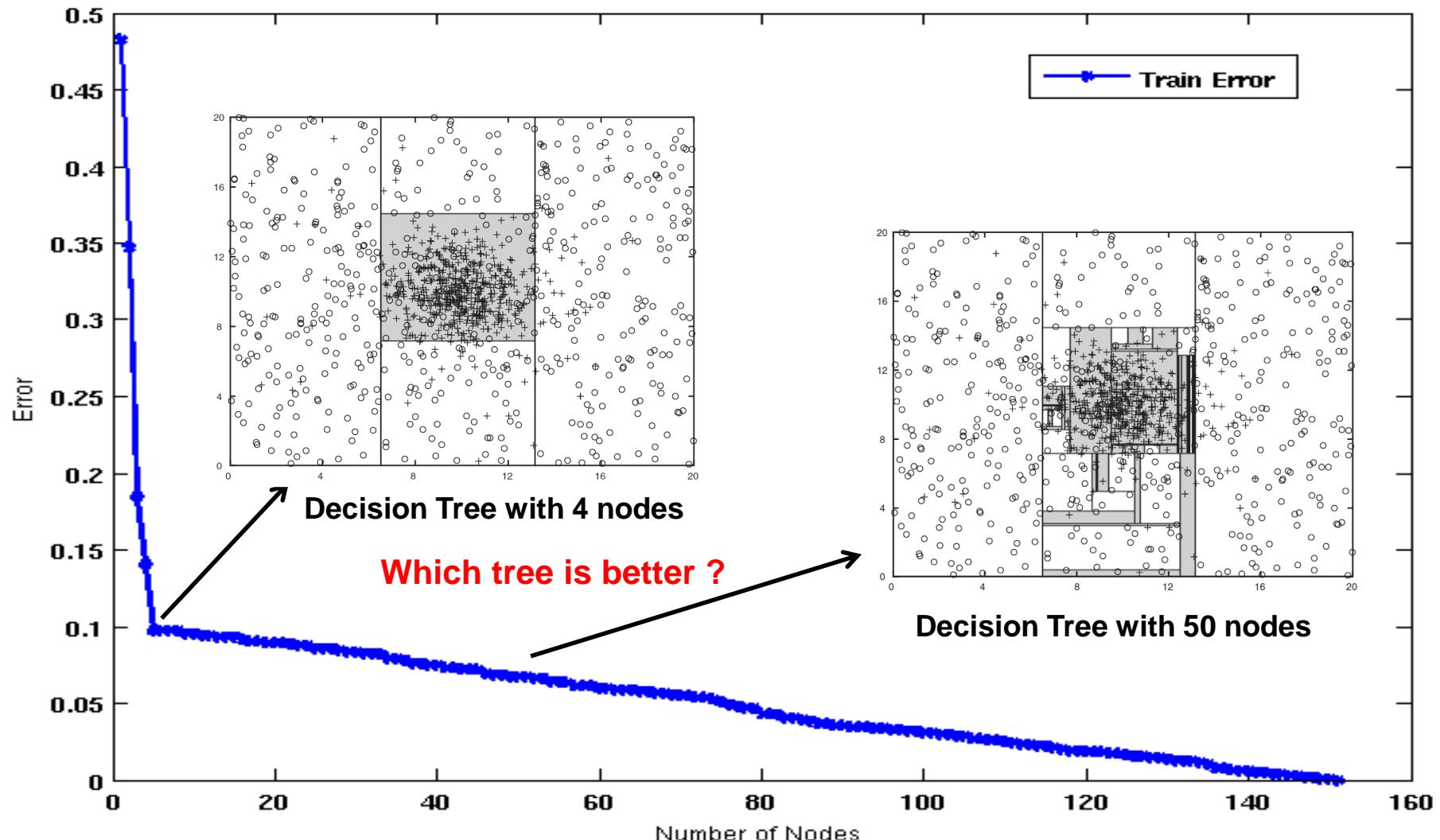
# Decision Tree with 4 nodes



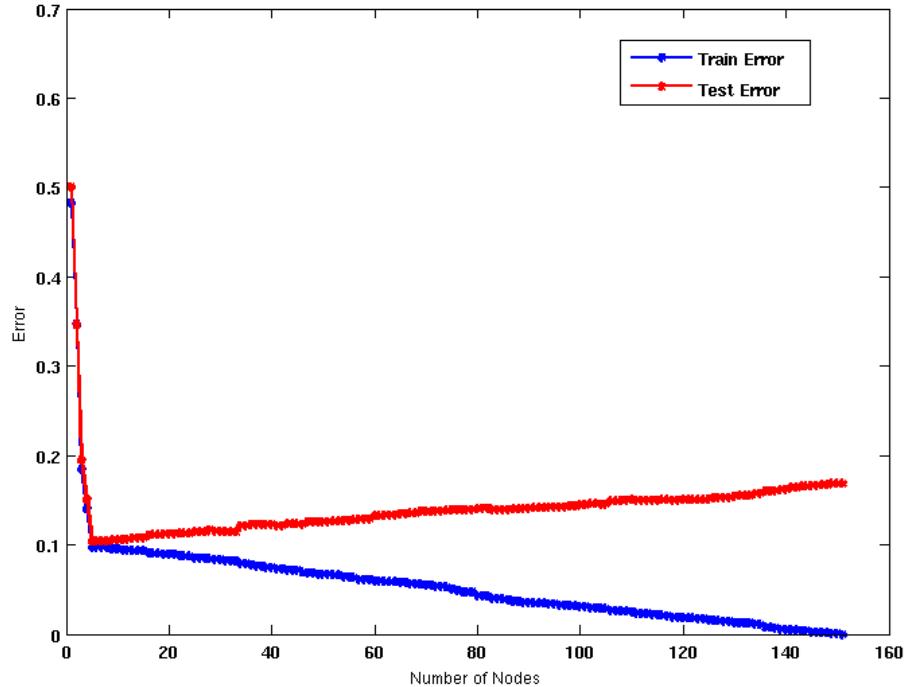
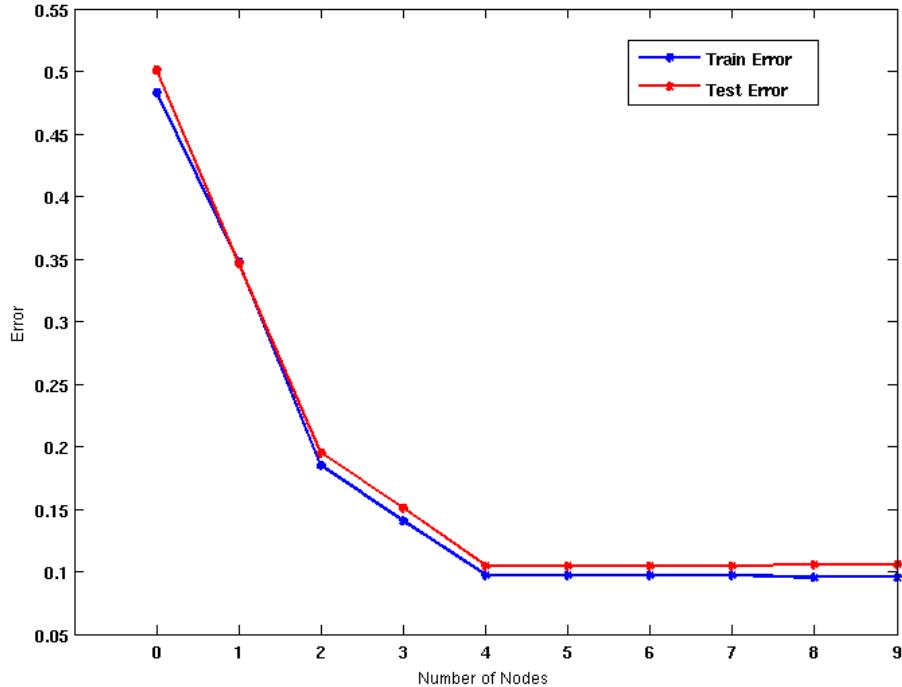
# Decision Tree with 50 nodes



# Which tree is better?



# Model Overfitting

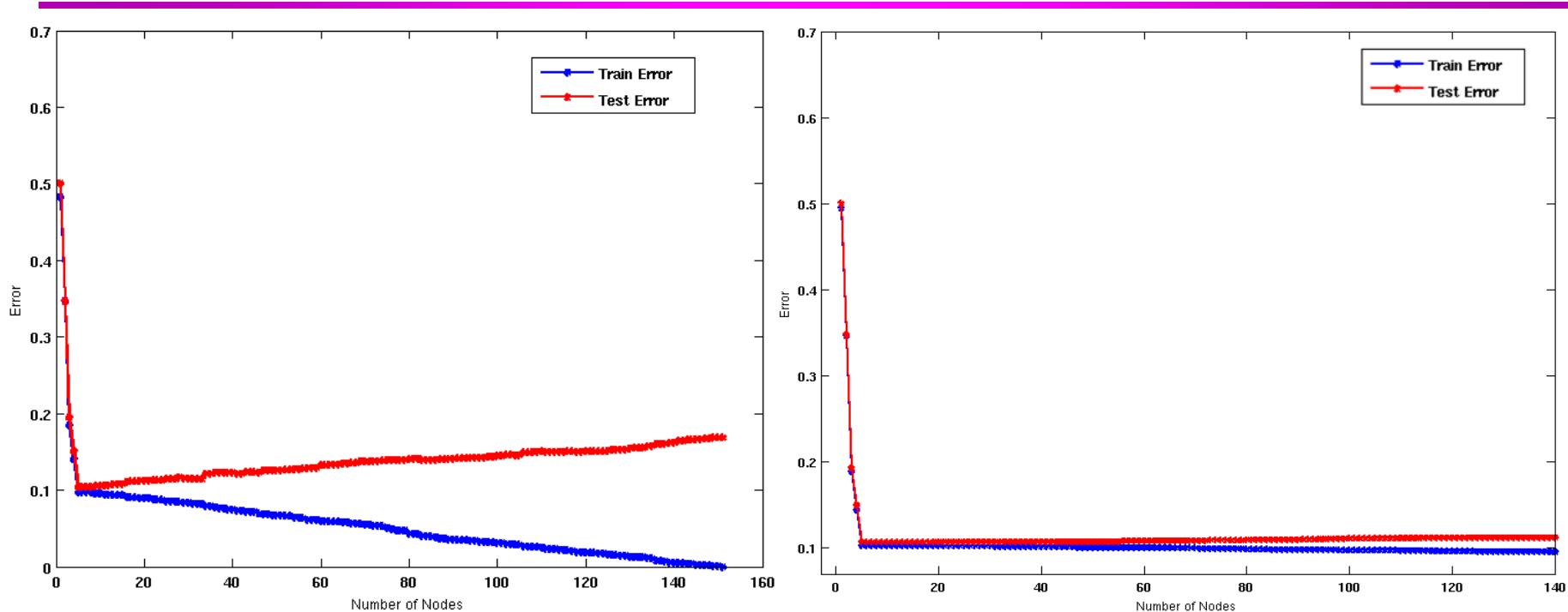


**Underfitting:** model çok basit olduğunda hem eğitim hem de test hataları büyüktür

**Overfitting:** model çok karmaşık olduğunda, eğitim hatası küçüktür ancak test hatası büyüktür

**Overfitting ve underfitting** model karmaşıklığıyla ilgili iki patolojidir.

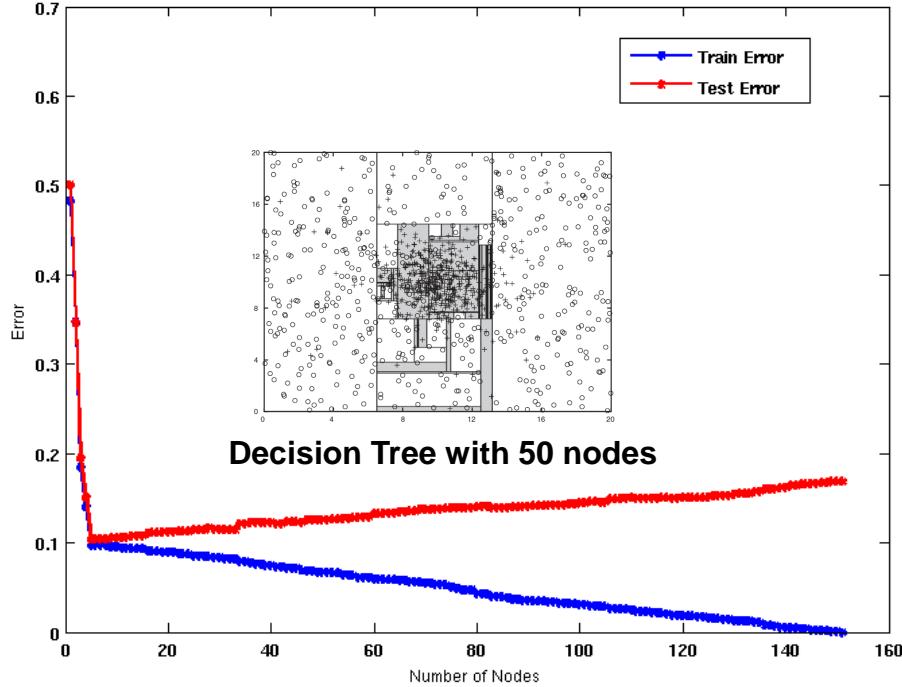
# Model Overfitting



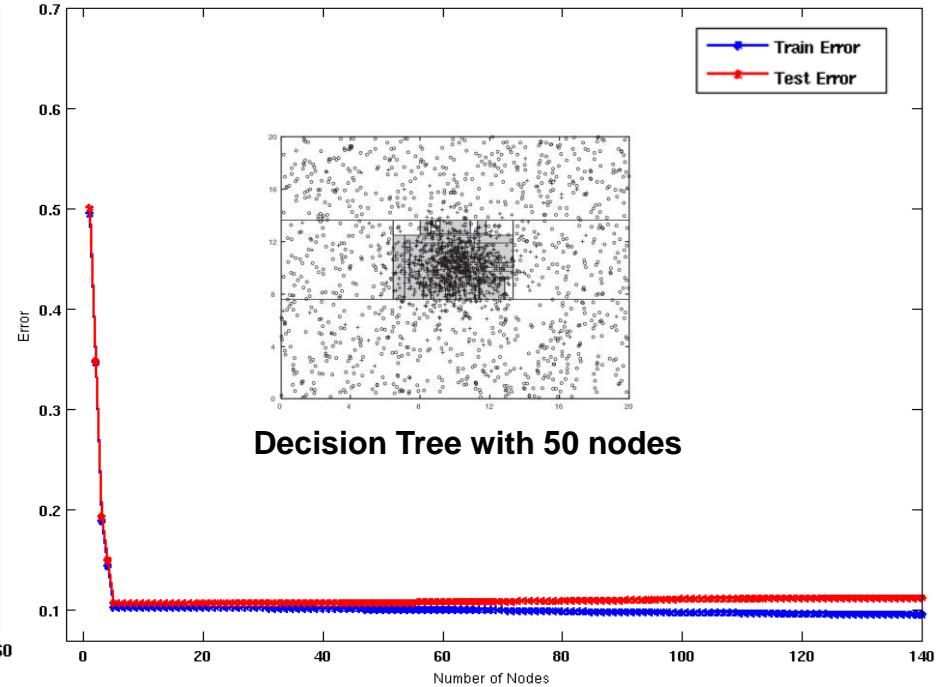
İki kat daha fazla veri örneği (data instances) kullanma

- Eğitim verileri temsili yetersizse (**under-representative**), artan düğüm sayısı ile test hataları artar ve eğitim hataları azalır
- Eğitim verilerinin boyutunu artırmak (**Increasing the size of training data**), belirli sayıda düğümde eğitim ve test hataları arasındaki farkı azaltır

# Model Overfitting



Decision Tree with 50 nodes



Decision Tree with 50 nodes

İki kat daha fazla veri örneği (data instances) kullanma

- Eğitim verileri temsili yetersizse (**under-representative**), artan düğüm sayısı ile test hataları artar ve eğitim hataları azalır
- Eğitim verilerinin boyutunu artırmak (**Increasing the size of training data**), belirli sayıda düğümde eğitim ve test hataları arasındaki farkı azaltır

# Reasons for Model Overfitting

---

---

- Limited Training Size
- High Model Complexity
  - Multiple Comparison Procedure

# Effect of Multiple Comparison Procedure

- Önümüzdeki 10 işlem gününde borsanın yükselişini/düşüşünü tahmin etme görevini düşünün
- Random guessing:  
 $P(\text{correct}) = 0.5$
- Arka arkaya 10 rastgele tahmin yapın :

$$P(\#\text{correct} \geq 8) = \frac{\binom{10}{8} + \binom{10}{9} + \binom{10}{10}}{2^{10}} = 0.0547$$

Day 1	Up
Day 2	Down
Day 3	Down
Day 4	Up
Day 5	Down
Day 6	Down
Day 7	Up
Day 8	Up
Day 9	Up
Day 10	Down

# Effect of Multiple Comparison Procedure

---

- Approach:
  - 50 analist getirin
  - Her analist 10 rastgele tahmin (*random guess*) yapar
  - En fazla sayıda doğru tahminde bulunan analisti seçin
- En az bir analistin en az 8 doğru tahmin yapma olasılığı

$$P(\# \text{correct} \geq 8) = 1 - (1 - 0.0547)^{50} = 0.9399$$

# Effect of Multiple Comparison Procedure

---

$$P(\#\text{correct} \geq 8) = 1 - (1 - 0.0547)^{50} = 0.9399$$

Her analistin en az sekiz defa doğru tahmin etme olasılığı düşük olsa da, **bunları bir araya getirsek**, bunu yapabilecek bir analyst bulma **olasılığımız yüksek**.

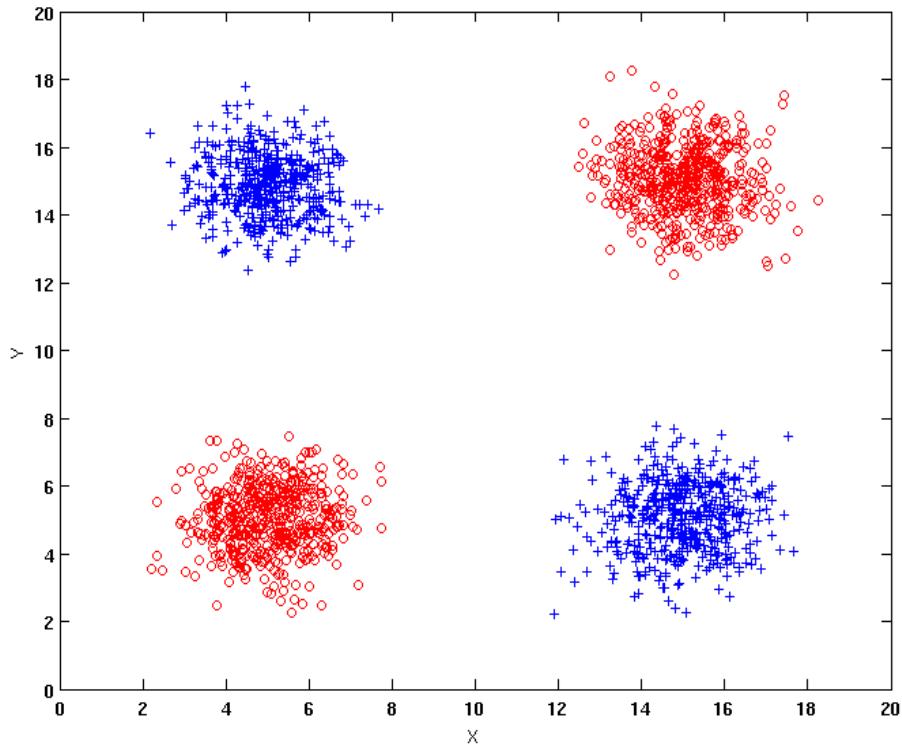
Buna ek olarak, gelecekte böyle bir analistin rastgele tahmin yoluyla doğru tahminler yapmaya devam edeceğini dair hiçbir garanti yoktur.

# Effect of Multiple Comparison Procedure

---

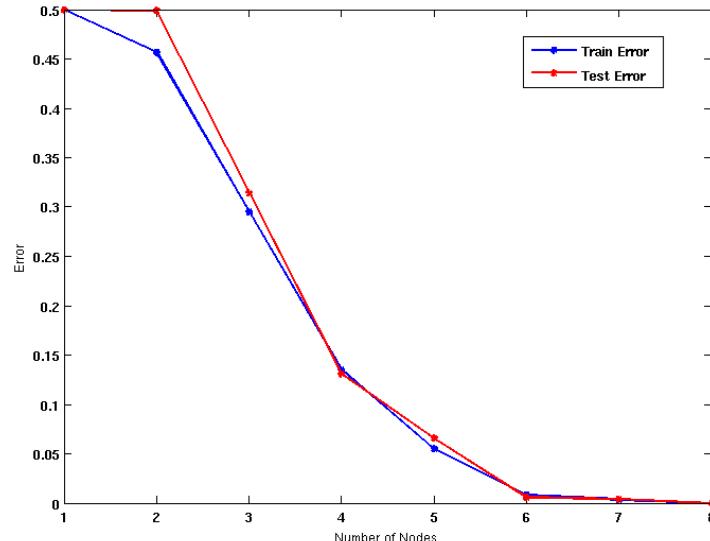
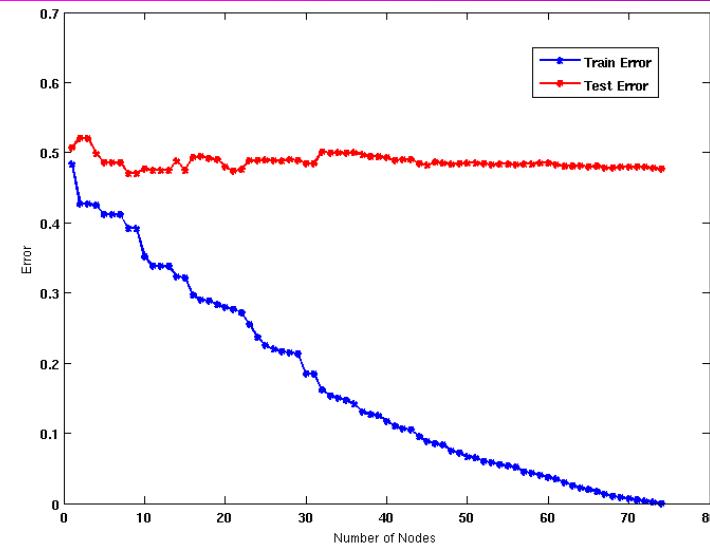
- Birçok algoritma aşağıdaki açgözlü stratejiyi (**greedy strategy**) kullanır:
  - İlk model:  $M$
  - Alternatif model:  $M' = M \cup \gamma$ ,  
burada  $\gamma$ , modele eklenecek bir bileşendir (örneğin, bir karar ağacının test koşulu)
  - İyileştirme varsa  $M'$ yi tutun,  $\Delta(M, M') > \alpha$
- Çoğu zaman,  $\gamma$  bir dizi alternatif bileşenden seçilir,  $\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_k\}$
- Birçok alternatif mevcutsa, modele istemeden alakasız bileşenler eklenebilir ve bu da modelin ezberlemesine (**overfitting**) neden olabilir.

# Effect of Multiple Comparison - Example



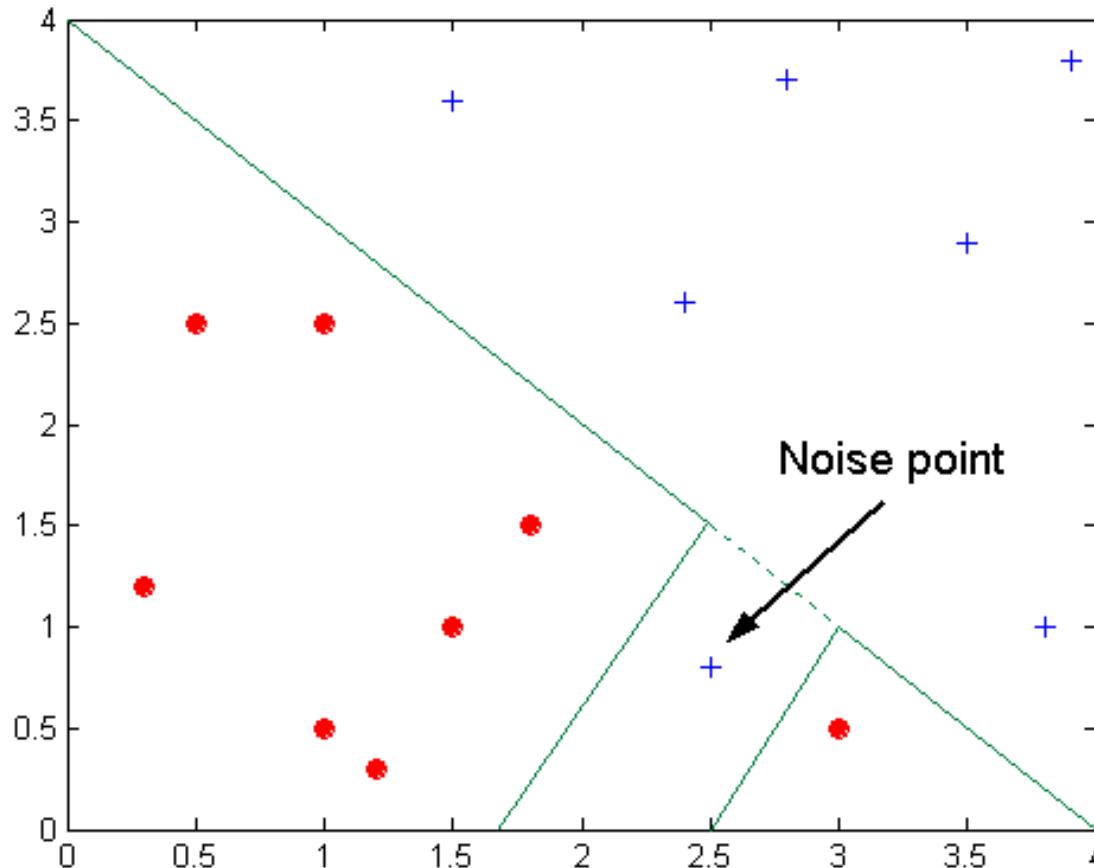
Use additional 100 noisy variables generated from a uniform distribution along with X and Y as attributes.

Use 30% of the data for training and 70% of the data for testing



Using only X and Y as attributes

# Overfitting due to Noise



Decision boundary is distorted by noise point

# Overfitting due to Noise

**Table 4.3.** An example training set for classifying mammals. Class labels with asterisk symbols represent mislabeled records.

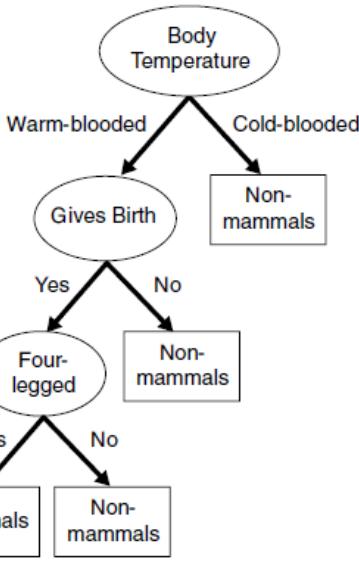
Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
porcupine	warm-blooded	yes	yes	yes	yes
cat	warm-blooded	yes	yes	no	yes
bat	warm-blooded	yes	no	yes	no*
whale	warm-blooded	yes	no	no	no*
salamander	cold-blooded	no	yes	yes	no
komodo dragon	cold-blooded	no	yes	no	no
python	cold-blooded	no	no	yes	no
salmon	cold-blooded	no	no	no	no
eagle	warm-blooded	no	no	no	no
guppy	cold-blooded	yes	no	no	no

Memelileri sınıflandırmak için kullanılan bir eğitim veri seti. Yıldız ile işaretli olalar yanlış etiketlenmiş kayıtlardır.

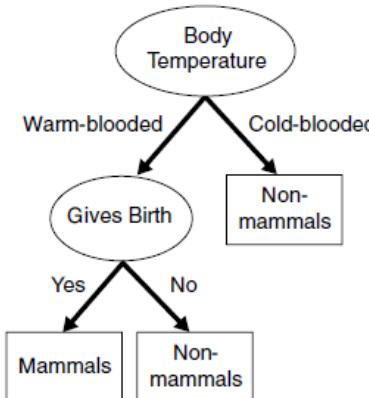
# Overfitting due to Noise

Table 4.4. An example test set for classifying mammals.

Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
human	warm-blooded	yes	no	no	yes
pigeon	warm-blooded	no	no	no	no
elephant	warm-blooded	yes	yes	no	yes
leopard shark	cold-blooded	yes	no	no	no
turtle	cold-blooded	no	yes	no	no
penguin	cold-blooded	no	no	no	no
eel	cold-blooded	no	no	no	no
dolphin	warm-blooded	yes	no	no	yes
spiny anteater	warm-blooded	no	yes	yes	yes
gila monster	cold-blooded	no	yes	yes	no



(a) Model M1



(b) Model M2

## Model M1

Training error =0%

Test error =30%

Test setinde daha düşük hata oranına sahip daha basit bir model var olduğu olduğu için, ilk karar ağacı **M1** 'in eğitim verilerini ezberlediği (**overfitted**) açıktır.

M1 modelindeki **Four-legged** öznitelik test koşulu yanlış etiketlenmiş eğitim kayıtlarına uyduğundan sahtedir (**spurious**). Bu da test setindeki kayıtların yanlış sınıflandırılmasına yol açar.

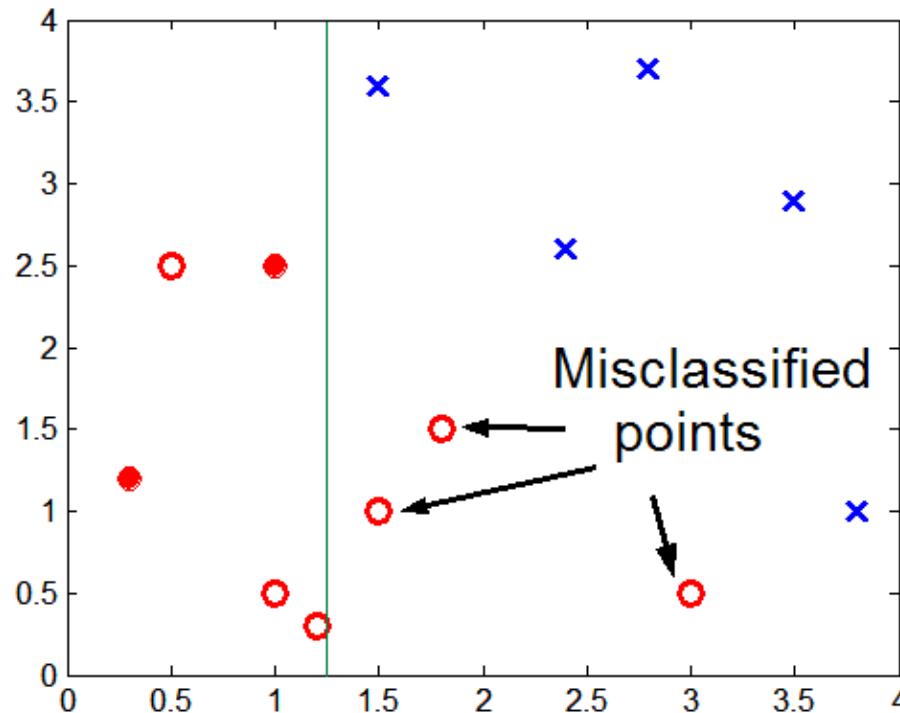
## Model M2

Training error =20%

Test error =10%

Figure 4.25. Decision tree induced from the data set shown in Table 4.3.

# Overfitting due to Insufficient Examples



Diyagramın alt yarısındaki veri noktalarının yetersiz oluşu, o bölgenin sınıf etiketlerini doğru şekilde tahmin etmeyi zorlaştırır

- **Insufficient number of training records** in the region causes the decision tree to predict the test examples using **other training records** that are **irrelevant** to the classification task

# Notes on Overfitting

---

- Ezberleme/aşırı uyum (*Overfitting*), gerekenden daha karmaşık karar ağaçlarına neden olur
- Eğitim hatası (**Training error**), ağaçın daha önce görülmemiş kayıtlar (**unseen records**) üzerinde ne kadar iyi performans göstereceğine dair iyi bir tahmin sağlamaz
- Genelleme hatalarını (*generalization error*) tahmin etmenin yollarına ihtiyacımız var

# Model Selection

---

- Model oluşturma sırasında gerçekleştirilir
- Amaç, modelin aşırı karmaşık olmamasını sağlamaktır (ezberlemeyi önlemek için)
- Genelleme hatasını tahmin etmemiz/kestirmemiz gerekiyor
  - Using Validation Set (*Doğrulama veri seti kullanma*)
  - Incorporating Model Complexity (*Model karmaşıklığını dahil etme*)
  - Estimating Statistical Bounds (*Istatistiksel sınırların tahmin edilmesi*)

# **Using Validation Set**

---

- Divide training data into two parts:
  - Training set:
    - ◆ use for model building
  - Validation set:
    - ◆ use for estimating generalization error
    - ◆ Note: validation set is not the same as test set
- Drawback:
  - Less data available for training

# **Incorporating Model Complexity**

---

- Rationale: **Occam's Razor** (or principle of parsimony:)
  - Benzer genelleme hatalarının olduğu iki model verildiğinde, daha karmaşık model yerine basit modeli tercih etmelisiniz.
  - Karmaşık bir modelin, verilerdeki hatalarla yanlışlıkla uydurulma şansı daha yüksektir
  - Bu nedenle, bir modeli değerlendirirken model karmaşıklığı işin içeresine dahil edilmelidir

$$\text{Gen. Error(Model)} = \text{Train. Error(Model, Train. Data)} + \alpha \times \text{Complexity(Model)}$$

# Estimating the Complexity of Decision Trees

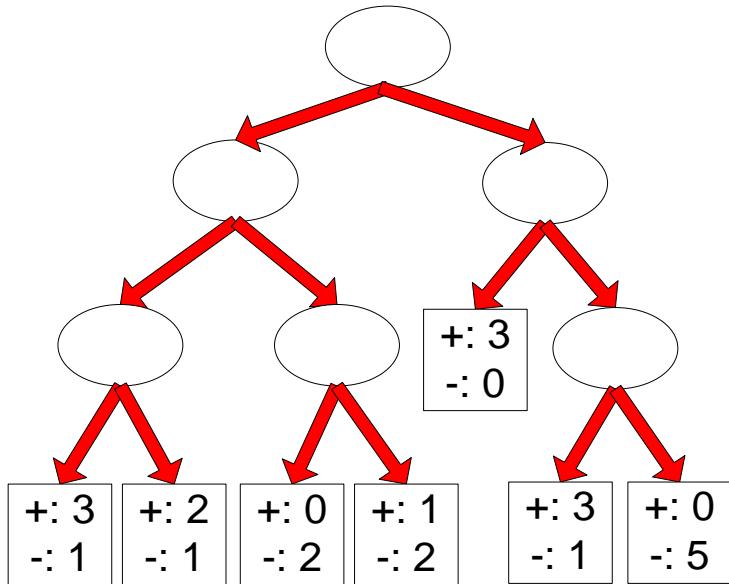
---

- **Pessimistic Error Estimate** of decision tree  $T$  with  $k$  leaf nodes:

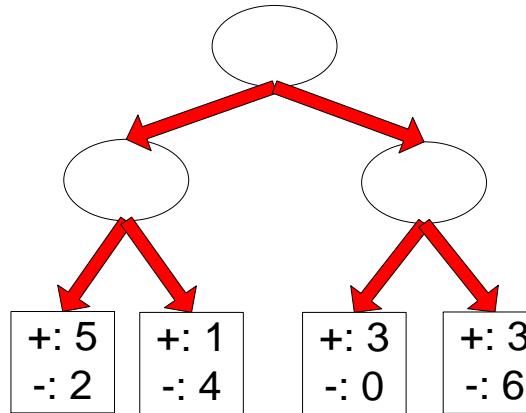
$$err_{gen}(T) = err(T) + \Omega \times \frac{k}{N_{train}}$$

- $err(T)$ : error rate on all training records
- $\Omega$ : trade-off hyper-parameter (similar to  $\alpha$ )
  - ◆ Relative cost of adding a leaf node (**penalty term for model complexity**)
- $k$ : number of leaf nodes
- $N_{train}$ : total number of training records

# Estimating the Complexity of Decision Trees: Example



Decision Tree,  $T_L$



$$e(T_L) = 4/24$$

$$e(T_R) = 6/24$$

$$\Omega = 1$$

$$e_{\text{gen}}(T_L) = 4/24 + 1 * 7/24 = 11/24 = 0.458$$

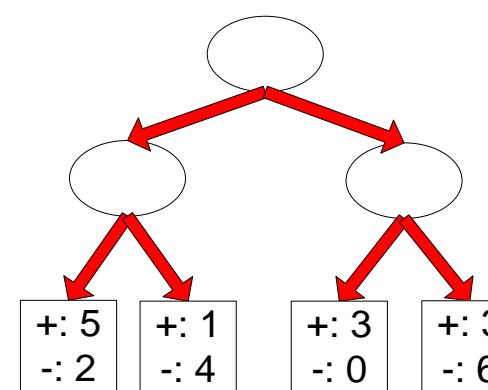
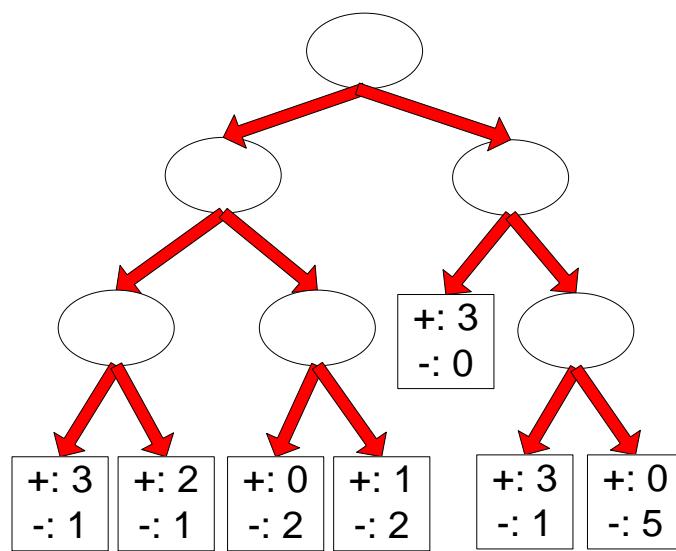
$$err_{\text{gen}}(T) = err(T) + \Omega \times \frac{k}{N_{\text{train}}}$$

$$e_{\text{gen}}(T_R) = 6/24 + 1 * 4/24 = 10/24 = 0.417$$

# Estimation of Generalization Errors

- Resubstitution Estimate:

- Eğitim hatasını genelleme hatasının iyimser bir tahmini olarak kullanma
- Iyimser hata tahmini olarak anılır (optimistic error estimate)



$$e(T_L) = 4/24$$

$$e(T_R) = 6/24$$

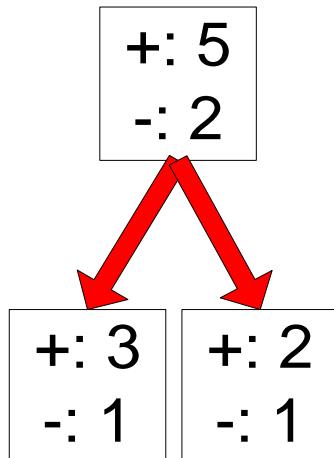
# Estimating Statistical Bounds

By approximating a binomial distribution with a normal distribution, the following upper bound of the error rate  $e$  can be derived:

$\alpha$  is the confidence level

$N$  is the total number of training records used to compute  $e$

$$e'(N, e, \alpha) = \frac{e + \frac{z_{\alpha/2}^2}{2N} + z_{\alpha/2} \sqrt{\frac{e(1-e)}{N} + \frac{z_{\alpha/2}^2}{4N^2}}}{1 + \frac{z_{\alpha/2}^2}{N}}$$



**Before splitting:**  $e = 2/7$ ,  $e'(7, 2/7, 0.25) = 0.503$

$$e'(T) = 7 \times 0.503 = 3.521$$

**After splitting:**

$$e(T_L) = 1/4, \quad e'(4, 1/4, 0.25) = 0.537$$

$$e(T_R) = 1/3, \quad e'(3, 1/3, 0.25) = 0.650$$

$$e'(T) = 4 \times 0.537 + 3 \times 0.650 = 4.098$$

$z_{\alpha/2}$  is the standardized value from a standard normal distribution

**Therefore, do not split**

# Handling Overfitting in Decision Tree Induction

---

- Karar ağacı indükleme (*decision tree induction*) bağlamında modelin ezberlemesini önlemek için iki strateji
  - Pre-Pruning
  - Post-pruning

# Model Selection for Decision Trees

---

- Pre-Pruning (Early Stopping Rule)

- Algoritmayı tamamen büyümüş bir ağaç haline gelmeden durdurun
- Bir düğüm için tipik durma koşulları(stopping conditions)
  - ◆ Tüm örnekler aynı sınıfı (**the same class**) aitse dur
  - ◆ Tüm öznitelik değerleri aynıysa dur
- Daha kısıtlayıcı koşullar :
  - ◆ Örnek sayısı (**number of instances**) kullanıcı tarafından belirlenen bazı eşiklerin altındaysa dur
  - ◆ Örneklerin sınıf dağılımı mevcut özelliklerden bağımsızsa durun (Örn.,  $\chi^2$  testi kullanarak)
  - ◆ **Mevcut düğümün** genişletilmesi **impurity ölçütlerini** (Örn., Gini veya information gain) iyileştirmiyorsa dur
  - ◆ Tahmini genelleme hatası belirli bir eşliğin altına düşerse dur

# Model Selection for Decision Trees

---

- Post-pruning
  - Karar ağacını tamamen büyütün
  - Alt ağaç değişimi (*Subtree replacement*)
    - ◆ Karar ağacının düğümlerini aşağıdan yukarıya (bottom-up) doğru kırpın
    - ◆ Kırmadan sonra genelleme hatası düzelirse, alt ağaç bir **yaprak düğüm (*leaf node*) ile değiştirin**
    - ◆ Yaprak düğümün sınıf etiketi, alt ağaçtaki örneklerin **çoğunluk sınıfından (*majority class*)** belirlenir
  - Subtree raising
    - ◆ Replace subtree with most frequently used branch

# Example of Post-Pruning

Class = Yes	20
Class = No	10
Error = 10/30	

Training Error (Before splitting) = 10/30

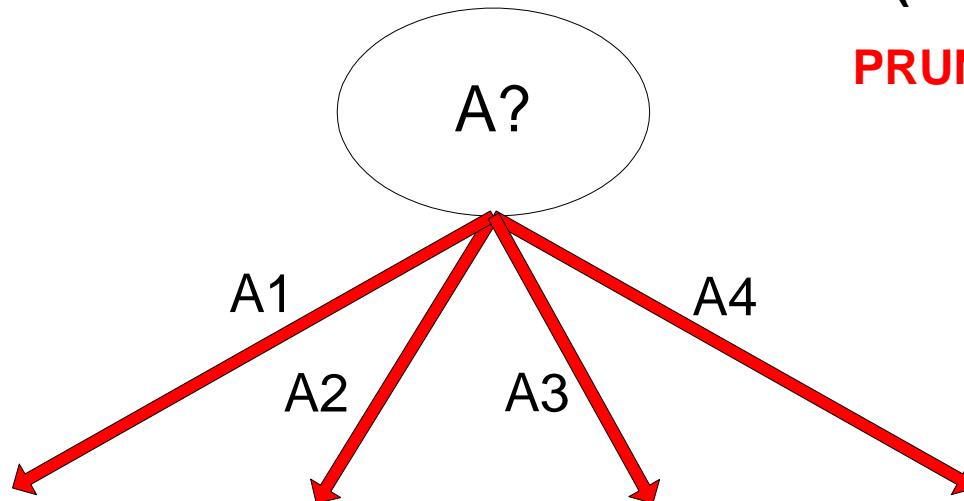
Pessimistic error =  $(10 + 0.5)/30 = 10.5/30$

Training Error (After splitting) = 9/30

Pessimistic error (After splitting)

$$= (9 + 4 \times 0.5)/30 = 11/30$$

**PRUNE!**



Class = Yes	8
Class = No	4

Class = Yes	3
Class = No	4

Class = Yes	4
Class = No	1

Class = Yes	5
Class = No	1

# Examples of Post-pruning

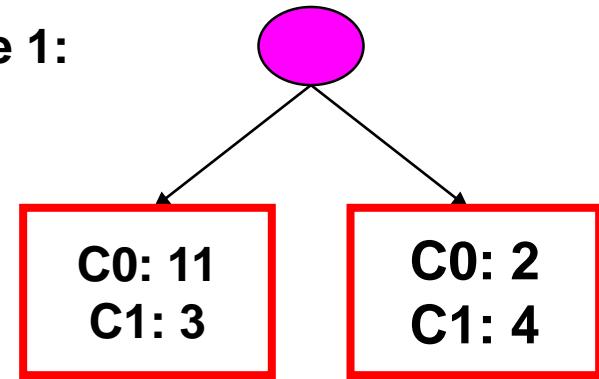
- Optimistic error?

**Don't prune for both cases**

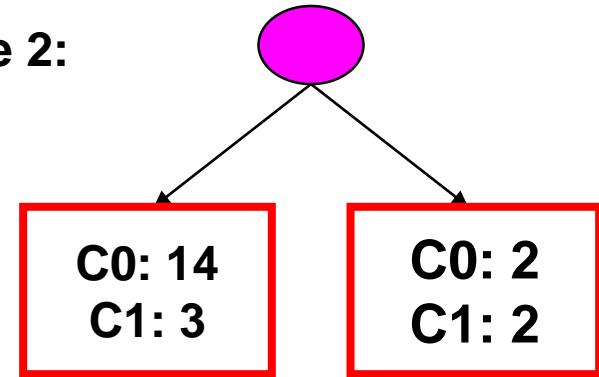
- Pessimistic error?

**Don't prune case 1, prune case 2**

**Case 1:**



**Case 2:**

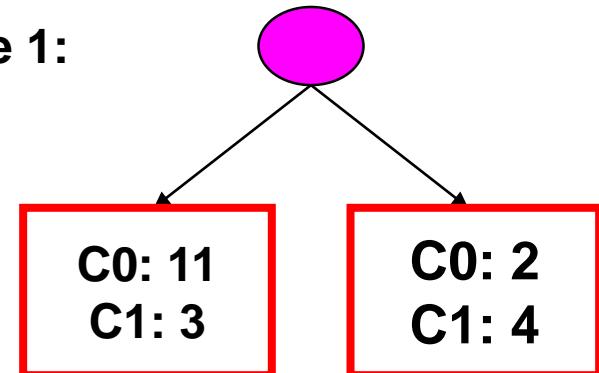


# Examples of Post-pruning

- Optimistic error?

Class = C0	13
Class = C1	7
Error = 7/20	

**Case 1:**

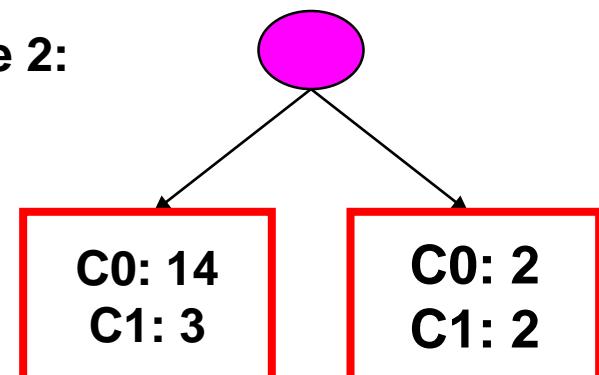


Optimistic error (After splitting) = 5/20

Don't prune for both cases

Class = C0	16
Class = C1	5
Error = 5/21	

**Case 2:**



Optimistic error (After splitting) = 5/21

# Examples of Post-pruning

- Pessimistic error?

Class = C0	13
Class = C1	7
Error = 7/20	

Training Error (Before splitting) = 7/20

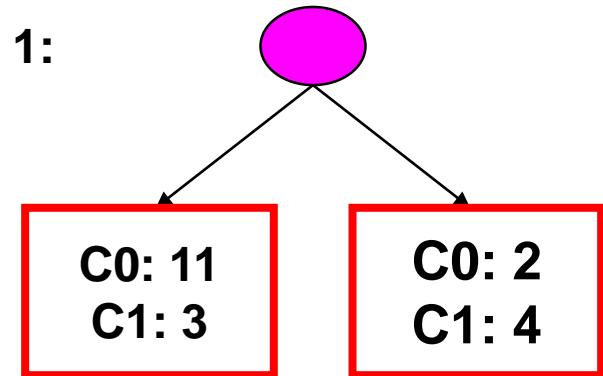
Pessimistic error =  $(7 + 0.5)/30 = 7.5/20$

Training Error (After splitting) = 5/20

Pessimistic error (After splitting)

$$= (5 + 2 \times 0.5)/20 = 6/20$$

Case 1:



DON'T PRUNE FOR CASE 1!

Class = C0	16
Class = C1	5
Error = 5/21	

Training Error (Before splitting) = 5/21

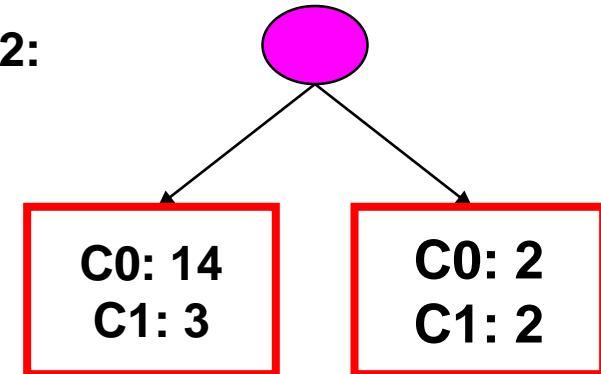
Pessimistic error =  $(5 + 0.5)/30 = 5.5/21$

Training Error (After splitting) = 5/21

Pessimistic error (After splitting)

$$= (5 + 2 \times 0.5)/21 = 6/21$$

Case 2:



PRUNE FOR CASE 2!

# Examples of Post-pruning

## Decision Tree:

```
depth = 1 :  
| breadth > 7 : class 1  
| breadth <= 7 :  
| | breadth <= 3 :  
| | | ImagePages > 0.375 : class 0  
| | | ImagePages <= 0.375 :  
| | | | totalPages <= 6 : class 1  
| | | | totalPages > 6 :  
| | | | | breadth <= 1 : class 1  
| | | | | breadth > 1 : class 0  
| width > 3 :  
| | MultiIP = 0:  
| | | ImagePages <= 0.1333 : class 1  
| | | ImagePages > 0.1333 :  
| | | | breadth <= 6 : class 0  
| | | | breadth > 6 : class 1  
| | MultiIP = 1:  
| | | TotalTime <= 361 : class 0  
| | | TotalTime > 361 : class 1  
depth > 1 :  
| | MultiAgent = 0:  
| | | depth > 2 : class 0  
| | | depth <= 2 :  
| | | | MultiIP = 1: class 0  
| | | | MultiIP = 0:  
| | | | | breadth <= 6 : class 0  
| | | | | breadth > 6 :  
| | | | | | RepeatedAccess <= 0.0322 : class 0  
| | | | | | RepeatedAccess > 0.0322 : class 1  
| | | MultiAgent = 1:  
| | | | totalPages <= 81 : class 0  
| | | | totalPages > 81 : class 1
```

## Simplified Decision Tree:

```
depth = 1 :  
| | ImagePages <= 0.1333 : class 1  
| | ImagePages > 0.1333 :  
| | | breadth <= 6 : class 0  
| | | breadth > 6 : class 1  
depth > 1 :  
| | MultiAgent = 0: class 0  
| | MultiAgent = 1:  
| | | totalPages <= 81 : class 0  
| | | totalPages > 81 : class 1
```

Subtree Raising

Subtree Replacement

# Data Mining Classification: Model Evaluation

---

---

Lecture Notes for Chapter 4

Introduction to Data Mining

by

Tan, Steinbach, Kumar

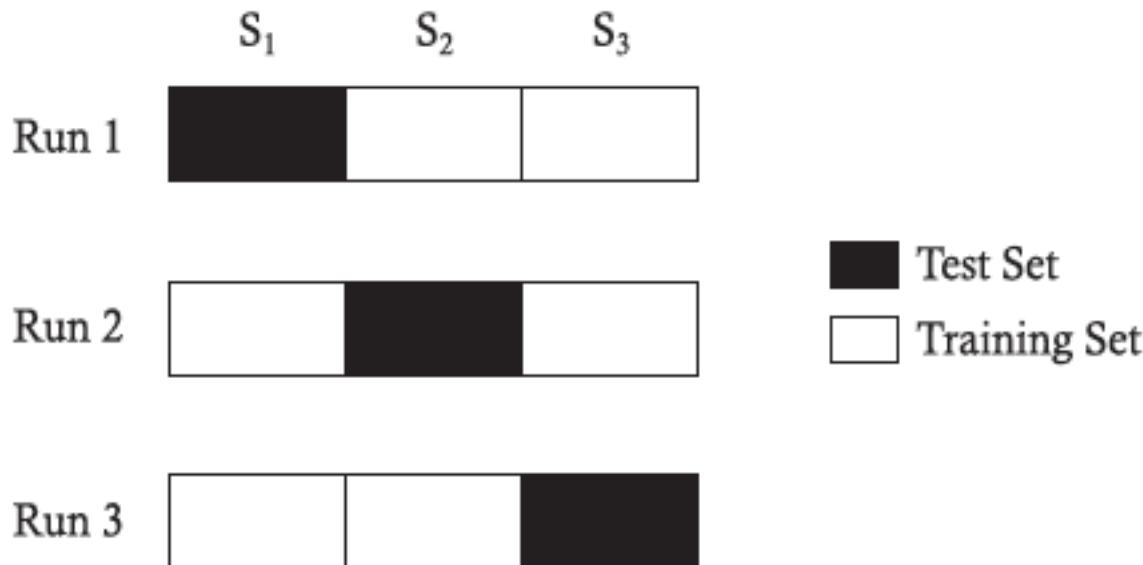
# Model Evaluation

---

- Purpose:
  - To estimate performance of classifier on previously unseen data (test set)
- Holdout
  - Reserve  $k\%$  for training and  $(100-k)\%$  for testing
    - ◆ Proportion Left at the discretion of the analysts (e.g., 50-50 or two thirds for training and one-third for testing).
  - Random subsampling: repeated holdout
- Cross validation
  - Partition data into  $k$  disjoint subsets
  - $k$ -fold: train on  $k-1$  partitions, test on the remaining one
  - Leave-one-out:  $k=n$

# Cross-validation Example

- 3-fold cross-validation



# Model Evaluation

---

- Performans Değerlendirmesi için Metrikler
  - **Bir modelin performansı** nasıl değerlendirilir?
- Performans Değerlendirme Yöntemleri
  - **Güvenilir tahminler** nasıl elde edilir?
- Model Karşılaştırma Yöntemleri
  - Rakip modeller arasında **göreceli performans** nasıl karşılaştırılır?

# Model Evaluation

---

- Performans Değerlendirmesi için Metrikler
  - Bir modelin **performansı** nasıl değerlendirilir?
- Performans Değerlendirme Yöntemleri
  - **Güvenilir tahminler** nasıl elde edilir?
- Model Karşılaştırma Yöntemleri
  - Rakip modeller arasında **göreceli performans** nasıl karşılaştırılır?

# Metrics for Performance Evaluation

- Bir modelin tahmin yeteneğine (**predictive capability**) odaklanır
  - Sınıflandırma hızı, model oluşturma hızı, ölçülebilirlik vb hususlardan ziyade...
- Confusion Matrix:

		PREDICTED CLASS	
		Class=Yes	Class>No
ACTUAL CLASS	Class=Yes	a	b
	Class>No	c	d

a: TP (true positive)  
b: FN (false negative)  
c: FP (false positive)  
d: TN (true negative)

# Metrics for Performance Evaluation...

		PREDICTED CLASS	
		Class=Yes	Class>No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class>No	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

# Metrics for Performance Evaluation...

---

- **True positive** (**TP**) or  $f_{++}$ , sınıflandırma modeli tarafından doğru bir şekilde tahmin edilen pozitif örneklerin sayısına karşılık gelir.
- **False negative** (**FN**) or  $f_{+-}$ , sınıflandırma modeli tarafından yanlış bir şekilde negatif olarak tahmin edilen pozitif örneklerin sayısına karşılık gelir.
- **False positive** (**FP**) or  $f_{-+}$ , sınıflandırma modeli tarafından yanlış bir şekilde pozitif olarak tahmin edilen negatif örneklerin sayısına karşılık gelir.
- **True negative** (**TN**) or  $f_{--}$ , sınıflandırma modeli tarafından doğru bir şekilde tahmin edilen negatif örneklerin sayısına karşılık gelir.

# Metrics for Performance Evaluation...

---

- Confusion matrisindeki sayılar ayrıca yüzde olarak da ifade edilebilir.
- True positive rate (TPR)** veya **sensitivity** (hassasiyet), model tarafından doğru şekilde tahmin edilen pozitif örneklerin oranı olarak tanımlanır, yani  
$$TPR = TP/(TP + FN).$$
- True negative rate (TNR)** veya **specificity**, model tarafından doğru bir şekilde tahmin edilen negatif örneklerin oranı olarak tanımlanır, yani,  
$$TNR = TN/(TN + FP).$$
- False positive rate (FPR)**, pozitif bir sınıf olarak tahmin edilen negatif örneklerin oranıdır, yani  
$$FPR = FP/(TN + FP),$$
- False negative rate (FNR)**, negatif bir sınıf olarak tahmin edilen pozitif örneklerin oranıdır, yani,  
$$FNR = FN/(TP + FN).$$

# Limitation of Accuracy

---

- 2-sınıflı bir problem düşünün
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10
- Model her şeyin *Sınıf-0* olacağını öngörürse, doğruluk  $9990/10000 = \% 99,9$ 'dur.
  - Doğruluk (**Accuracy**) yanlıştır çünkü model herhangi bir *Sınıf-1* örneği tespit etmez

its limitation is obvious for **imbalanced datasets**

# Cost Matrix

		PREDICTED CLASS		
		C(i j)	Class=Yes	Class>No
ACTUAL CLASS	Class=Yes	C(Yes Yes)	C(No Yes)	
	Class>No	C(Yes No)	C(No No)	

$C(i|j)$ : Sınıf jrneğini sınıf i olarak yanlış sınıflandırmanın maliyeti

# Computing Cost of Classification

Cost Matrix		PREDICTED CLASS	
ACTUAL CLASS	C(i  j)	+	-
	+	-1	100
	-	1	0

Model M <sub>1</sub>	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	150	40
	-	60	250

Accuracy = 80%

Cost = 3910

Model M <sub>2</sub>	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	250	45
	-	5	200

Accuracy = 90%

Cost = 4255

# Cost vs Accuracy

Count	PREDICTED CLASS	
ACTUAL CLASS	Class=Yes	Class=No
	Class=Yes	a
	Class=No	d

Accuracy is proportional to cost if

1.  $C(Yes|No)=C(No|Yes) = q$
2.  $C(Yes|Yes)=C(No|No) = p$

$$N = a + b + c + d$$

$$\text{Accuracy} = (a + d)/N$$

Cost	PREDICTED CLASS	
ACTUAL CLASS	Class=Yes	Class=No
	Class=Yes	p
	Class=No	q

$$\text{Cost} = p (a + d) + q (b + c)$$

$$= p (a + d) + q (N - a - d)$$

$$= q N - (q - p)(a + d)$$

$$= N [q - (q-p) \times \text{Accuracy}]$$

# Cost-Sensitive Measures

$$\text{Precision (p)} = \frac{a}{a + c}$$

$$\text{Recall (r)} = \frac{a}{a + b}$$

$$F\text{-measure (F)} = \frac{2rp}{r + p} = \frac{2a}{2a + b + c}$$

In principle, *F-measure (F<sub>1</sub>)* represents a harmonic mean between recall and precision, i.e.

$$F_1 = \frac{2}{\frac{1}{r} + \frac{1}{p}}$$

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	a (TP)	b (FN)
	Class>No	c (FP)	d (TN)

- **Precision** is biased towards C(Yes|Yes) & C(Yes|No)
- **Recall** is biased towards C(Yes|Yes) & C(No|Yes)
- **F-measure** is biased towards all except C(No|No)

$$\text{Weighted Accuracy} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

# Cost-Sensitive Measures

---

- **Precision**, sınıflandırıcının pozitif bir sınıf olarak bildirdiği grupta gerçekten pozitif çıkan kayıtların oranını belirler.
- **Recall**, sınıflandırıcı tarafından doğru bir şekilde tahmin edilen pozitif örneklerin oranını ölçer.

# Model Evaluation

---

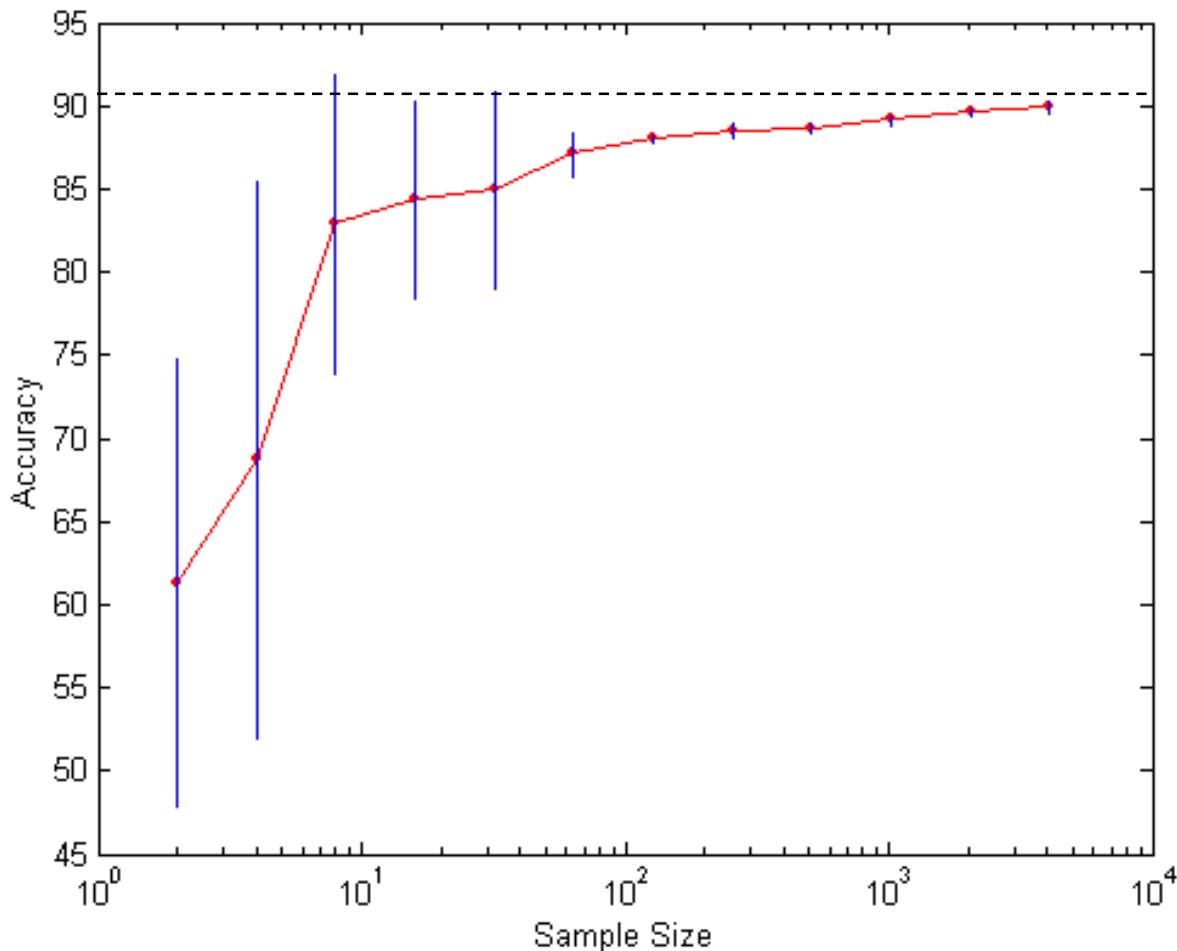
- Performans Değerlendirmesi için Metrikler
  - Bir modelin performansı nasıl değerlendirilir?
- Performans Değerlendirme Yöntemleri
  - Güvenilir tahminler nasıl elde edilir?
- Model Karşılaştırma Yöntemleri
  - Rakip modeller arasında göreceli performans nasıl karşılaştırılır?

# Methods for Performance Evaluation

---

- Güvenilir bir performans tahmini nasıl elde edilir?
- Bir modelin performansı, öğrenme algoritmasının yanı sıra diğer faktörlere de bağlı olabilir:
  - Sınıf dağılımı (*Class distribution*)
  - Yanlış sınıflandırma maliyeti (*Cost of misclassification*)
  - Eğitim ve test setlerinin boyutu

# Learning Curve



- Learning curve shows how accuracy changes with varying sample size
- Requires a sampling schedule for creating learning curve:
  - Arithmetic sampling (Langley, et al)
  - Geometric sampling (Provost et al)

## Effect of small sample size:

- Bias in the estimate
- Variance of estimate

# Methods of Estimation

---

- Holdout
  - Reserve 2/3 for training and 1/3 for testing
- Random subsampling
  - Repeated holdout
- Cross validation
  - Partition data into  $k$  disjoint subsets
  - $k$ -fold: train on  $k-1$  partitions, test on the remaining one
  - Leave-one-out:  $k=n$

# Model Evaluation

---

- Performans Değerlendirmesi için Metrikler
  - Bir modelin performansı nasıl değerlendirilir?
- Performans Değerlendirme Yöntemleri
  - Güvenilir tahminler nasıl elde edilir?
- Model Karşılaştırma Yöntemleri
  - Rakip modeller arasında göreceli performans nasıl karşılaştırılır?

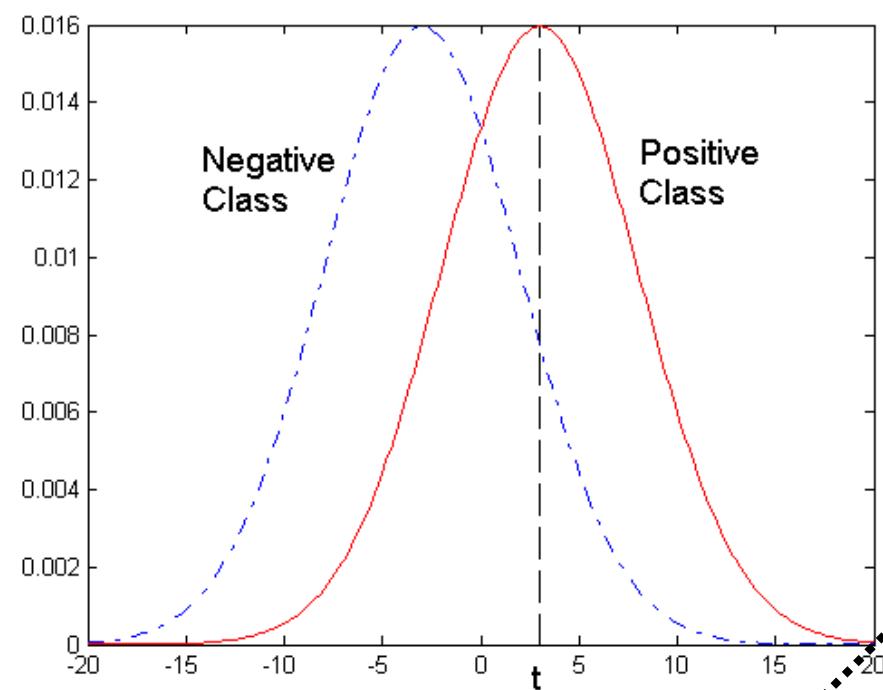
# ROC (Receiver Operating Characteristic)

---

- 1950'lerde gürültülü sinyalleri analiz etmek amacıyla sinyal algılama teorisi için geliştirildi
  - Pozitif işaretler ve yanlış alarmlar arasındaki ödünləşimi karakterize eder (**trade-off between positive hits and false alarms**)
  - ROC eğrisi (curve), TP oranını (y ekseninde) FP oranına (x ekseninde) karşı karakterize eder
- Her sınıflandırıcının performansı ROC eğrisinde bir nokta olarak temsil edilir
  - Algoritmanın eşğini, örneklem dağılımını veya maliyet matrisini değiştirme noktanın konumunu değiştirir.

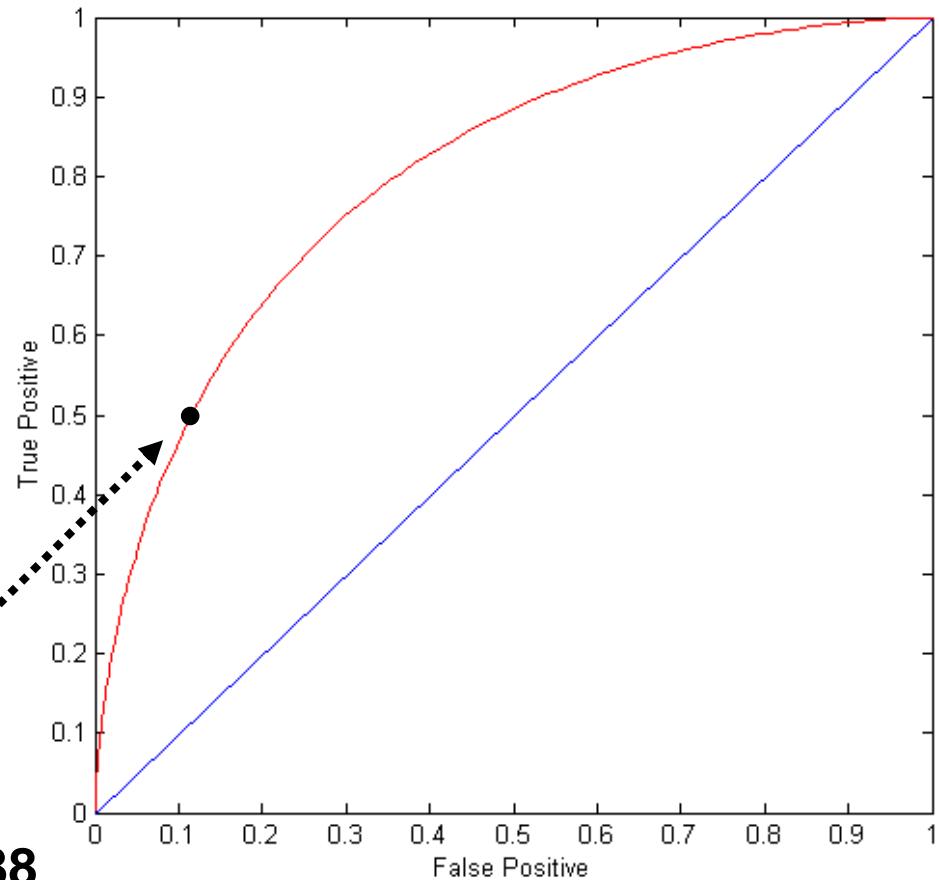
# ROC Curve

- 1-dimensional data set containing 2 classes (positive and negative)
- any points located at  $x > t$  is classified as positive



At threshold  $t$ :

$\text{TP}=0.5, \text{FN}=0.5, \text{FP}=0.12, \text{TN}=0.88$

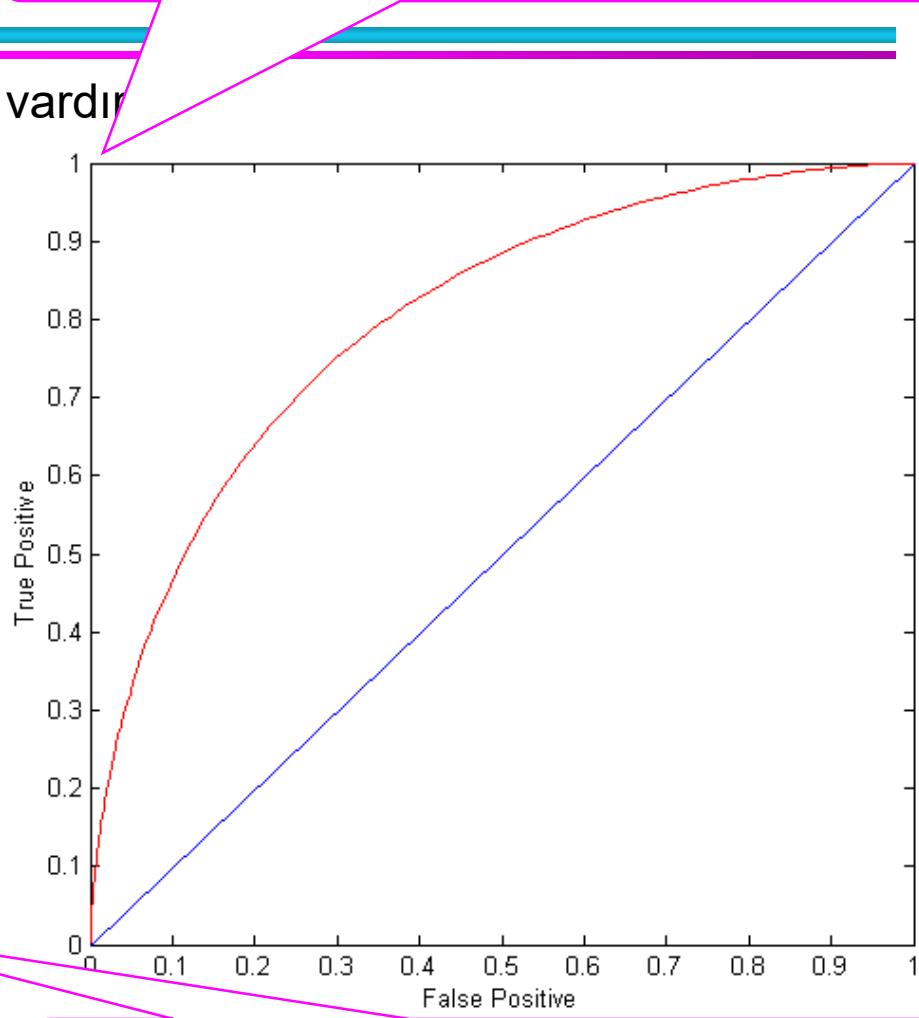


# ROC Curve

İyi bir sınıflandırma modeli, diyagramın sol üst köşesine mümkün olduğunca yakın konumlanmalıdır.

Bir ROC eğrisi boyunca birkaç kritik nokta vardır

- ( $TPR=0, FPR=0$ ): Model, her örneğin bir negatif sınıf olacağını öngörür.
- ( $TPR=1, FPR=1$ ): Model, her örneğin pozitif bir sınıf olduğunu öngörür.
- ( $TPR=1, FPR=0$ ): İdeal model.
- Köşegen (*Diagonal line*):
  - Random guessing
  - Below diagonal line:
    - ◆ prediction is opposite of the true class



Rastgele tahmin (**Random guessing**), bir kaydın, öznitelik kümesine bakılmaksızın, sabit bir olasılık  $p$  ile pozitif bir sınıf olarak sınıflandırılması anlamına gelir.

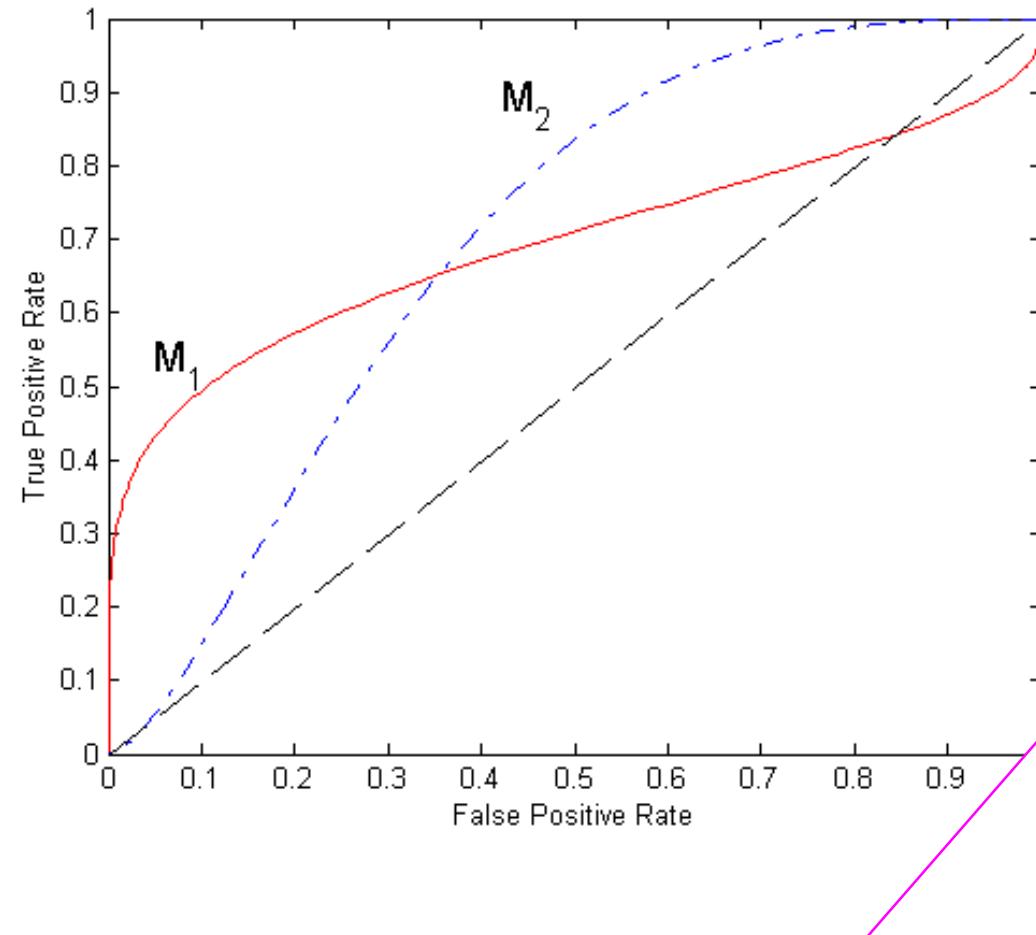
---

---

**Rastgele tahmin**, bir **kaydın**, öznitelik kümese  
bakılmaksızın, sabit bir olasılık  $p$  ile pozitif bir sınıf olarak  
sınıflandırılması anlamına gelir.

- Örneğin,  $n_+$  pozitif örnekler ve  $n_-$  negatif örnekler içeren bir veri kümesi düşünün.
- Rastgele sınıflandırıcının pozitif örneklerin  $pn_+$  ‘sini doğru şekilde sınıflandırması ve negatif örneklerin  $pn_-$  ‘sini yanlış sınıflandırması beklenir.
- Bu nedenle, sınıflandırıcının TPR'si  $(pn_+)/n_+ = p$ ,  
FPR'si  $(pn_-)/n_- = p$ .
- TPR ve FPR aynı olduğundan, **rastgele sınıflandırıcı için ROC eğrisi** her zaman **ana köşegen** boyunca yer alır.

# Using ROC for Model Comparison



- (In this example) No model consistently outperform the other
  - $M_1$  is better for small FPR
  - $M_2$  is better for large FPR
- Area Under the ROC curve
  - Ideal:
    - Area = 1
  - Random guess:
    - Area = 0.5

The area under the ROC curve (AUC) provides another approach for evaluating **which model is better on average**.

# How to Construct an ROC curve

Instance	$P(+ A)$	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

- Use classifier that produces posterior probability for each test instance  $P(+|A)$
- Sort the instances according to  $P(+|A)$  in decreasing order
- Apply threshold at each unique value of  $P(+|A)$
- Count the number of TP, FP, TN, FN at each threshold
- TP rate,  $TPR = TP/(TP+FN)$
- FP rate,  $FPR = FP/(FP + TN)$

# How to Construct an ROC curve

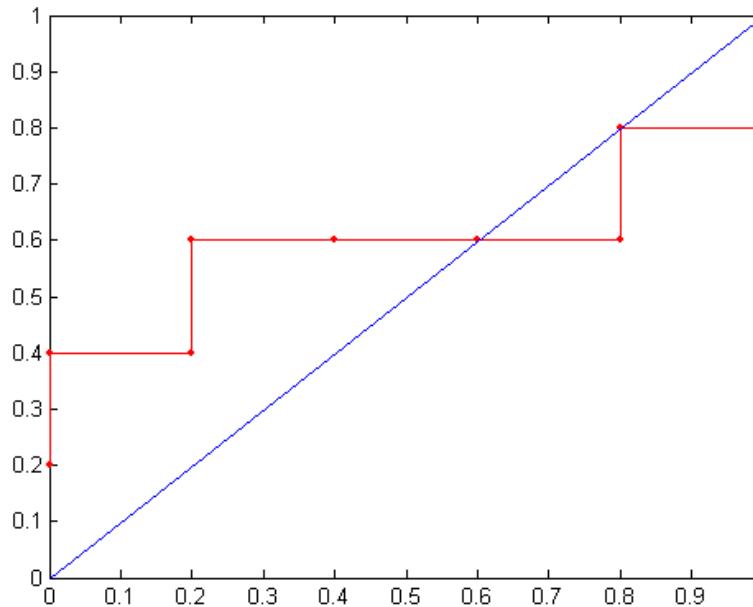
---

1. Sürekli değerli çıktıların pozitif sınıf için tanımlandığını varsayıarak, kayıtları çıktı değerlerinin artan sırasına göre sıralayın.
2. En düşük dereceli test kaydını seçin (yani, en düşük çıktı değerine sahip kaydı). Seçilen kaydı ve üzerinde sıralananları pozitif sınıf'a atayın. Bu yaklaşım, tüm test kayıtlarının pozitif sınıf olarak sınıflandırılmasına eşdeğerdir. Tüm pozitif örnekler doğru şekilde sınıflandırıldığı ve negatif örnekler yanlış sınıflandırıldığı için,  $TPR = FPR = 1$ .
3. Sıralanan listeden sonraki test kaydını seçin. Seçili kaydı ve üzerinde sıralananları pozitif, altında olanları negatif olarak sınıflandırın. Önceden seçilen kaydın gerçek sınıf etiketini inceleyerek TP ve FP sayılarını güncelleyin. Önceden seçilen kayıt pozitif bir sınıfsa, TP sayısı azaltılır ve FP sayısı öncekiyle aynı kalır. Önceden seçilen kayıt negatif bir sınıfsa, FP sayısı azaltılır ve TP sayısı öncekiyle aynı kalır.
4. Adımı tekrarlayın ve en yüksek dereceli test kaydı seçilene kadar TP ve FP sayılarını uygun şekilde güncelleyin.
5. Sınıflandırıcının FPR'sine karşı TPR'yi çizin.

# How to construct an ROC curve

Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
→ TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
→ FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

ROC Curve:



# Test of Significance

---

- Given two models:
  - Model M1: accuracy = 85%, tested on 30 instances
  - Model M2: accuracy = 75%, tested on 5000 instances
- M1'in M2'den daha iyi olduğunu söyleyebilir miyiz?
  - M1 ve M2'nin doğruluğuna ne kadar güvenebiliriz?
  - Performans ölçüsündeki fark, **test setindeki rastgele dalgalanmaların** bir sonucu olarak açıklanabilir mi?

# Confidence Interval for Accuracy

---

- Güven aralığını belirlemek için, doğruluk (accuracy) ölçüsünü yöneten olasılık dağılımını oluşturmamız gereklidir.
- Sınıflandırma görevini binom deneyi olarak modelleyerek güven aralığını türetmek için bir yaklaşımı ihtiyacımız var.
- Aşağıda bir binom deneyinin (Binomial Experiment) özelliklerinin bir listesi verilmiştir:
  1. Deney, her denemenin iki olası sonuca sahip olduğu  $N$  bağımsız denemeden oluşur: başarı (**success**) veya başarısızlık (**failure**).
  2. Her deneme başarı olasılığı,  $p$ , sabittir.

# Confidence Interval for Accuracy

- **Binom deneyine bir örnek**, (bir yazı tura denemsinde) bozuk para  $N$  kez atıldığında ortaya çıkan tura sayısını saymaktır.
- $X$ ,  $N$  denemede gözlemlenen başarı sayısı ise,  **$X$ 'in belirli bir değeri alma olasılığı**, ortalama  $Np$  ve varyans  $Np(1 - p)$  olan bir binom dağılımı ile verilir:

$$P(X = v) = \binom{N}{v} p^v (1 - p)^{N-v}.$$

- Örneğin, bozuk para adil (*fair coin*) ise ( $p = 0.5$ ) ve elli kez atılmışsa, turanın 20 kez ortaya çıkma olasılığı

$$P(X = 20) = \binom{50}{20} 0.5^{20} (1 - 0.5)^{30} = 0.0419.$$

- Deney birçok kez tekrarlanırsa, ortaya çıkması beklenen ortalama tura sayısı  $50 \times 0.5 = 25$  iken varyansı  $50 \times 0.5 \times 0.5 = 12.5$ 'tir.

# Confidence Interval for Accuracy

---

- Tahmin, bir Bernoulli denemesi olarak kabul edilebilir
  - Bernoulli denemesinin 2 olası sonucu vardır
  - Tahmin için olası sonuçlar: doğru veya yanlış
  - Bernoulli denemeleri koleksiyonunun Binom dağılımı vardır:
    - ◆  $x \sim \text{Bin}(N, p)$       $x$ : doğru tahmin sayısı
    - ◆ e.g: Adil bir bozuk parayı 50 kez atarsan, kaç tura çıkar?  
Beklenen tura sayısı =  $N \times p = 50 \times 0.5 = 25$
- $X$  (doğru tahmin sayısı) verildiğinde veya eşdeğer olarak,  $\text{acc} = x / N$  ve  $N$  (test örneği sayısı) verildiğinde,  
 $p$ 'yi (modelin gerçek doğruluğunu) tahmin edebilir miyiz?

true accuracy of model

# Confidence Interval for Accuracy

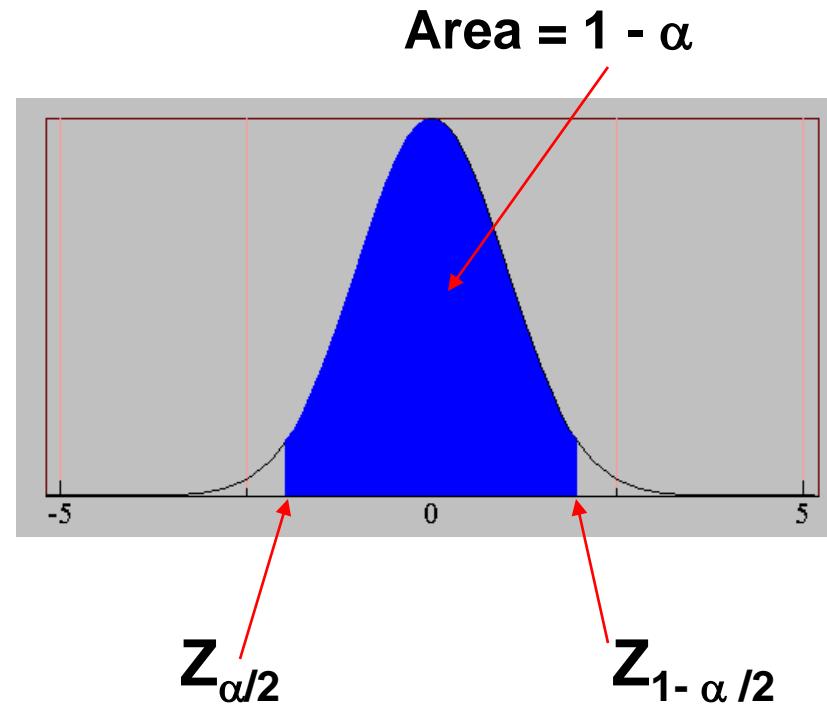
---

- Test kayıtlarının sınıf etiketlerini tahmin etme görevi de bir binom deneyi (**binomial experiment**) olarak düşünülebilir.
- $N$  kayıt içeren bir test seti verildiğinde,  $X$  bir model tarafından **doğru tahmin edilen kayıt sayısı** ve  $p$  modelin gerçek doğruluğu (**the true accuracy**) olsun.
- Tahmin görevini binom deneyi olarak modelleyerek,  $X; Np$  ortalama ve  $Np(1 - p)$  varyans ile bir binom dağılımına sahiptir.
- Deneysel doğruluğun,  $acc = X / N$ , aynı zamanda  $p$  ortalama ve  $p(1 - p) / N$  varyans ile bir binom dağılımına sahip olduğu gösterilebilir (bkz. önceki slaytlar).
- Binom dağılım,  $acc$  için güven aralığını tahmin etmek amacıyla kullanılmasına rağmen,  $N$  **yeterince büyük** olduğunda genellikle normal dağılımla yaklaşık olarak tahmin edilir.

# Confidence Interval for Accuracy

- For large test sets ( $N > 30$ ),
  - acc has a normal distribution with mean  $p$  and variance  $p(1-p)/N$

$$P(Z_{\alpha/2} < \frac{acc - p}{\sqrt{p(1-p)/N}} < Z_{1-\alpha/2}) = 1 - \alpha$$



- Confidence Interval for  $p$ :

$$p = \frac{2 \times N \times acc + Z_{\alpha/2}^2 \pm \sqrt{Z_{\alpha/2}^2 + 4 \times N \times acc - 4 \times N \times acc^2}}{2(N + Z_{\alpha/2}^2)}$$

# Confidence Interval for Accuracy

- Consider a model that produces an accuracy of 80% when evaluated on 100 test instances:

- $N=100$ , acc = 0.8
- Let  $1-\alpha = 0.95$  (95% confidence)
- From probability table,  $Z_{\alpha/2}=1.96$

N	50	100	500	1000	5000
p(lower)	0.670	0.711	0.763	0.774	0.789
p(upper)	0.888	0.866	0.833	0.824	0.811

1- $\alpha$	Z
0.99	2.58
0.98	2.33
0.95	1.96
0.90	1.65

Note that the confidence interval becomes tighter when  $N$  increases

Confidence Interval:  
71.1% and 86.7%

# Comparing Performance of 2 Models

---

- Given two models, say M1 and M2, which is better?
  - M1 is tested on D1 (size=n1), found error rate =  $e_1$
  - M2 is tested on D2 (size=n2), found error rate =  $e_2$
  - Assume D1 and D2 are independent test sets
  - If n1 and n2 are sufficiently large, then

$$e_1 \sim N(\mu_1, \sigma_1)$$

$$e_2 \sim N(\mu_2, \sigma_2)$$

the error rates  $e_1$  and  $e_2$  can be approximated using normal distributions.

- Approximate:  $\hat{\sigma}_i = \frac{e_i(1-e_i)}{n_i}$

# Comparing Performance of 2 Models

- To test if performance difference is **statistically significant**:  $d = e_1 - e_2$

- $d \sim N(d_t, \sigma_d)$  where  $d_t$  is the **true difference**
- Since  $D_1$  and  $D_2$  are independent, their variance adds up:

$$\begin{aligned}\sigma_d^2 &= \sigma_1^2 + \sigma_2^2 \approx \hat{\sigma}_1^2 + \hat{\sigma}_2^2 \\ &= \frac{e_1(1-e_1)}{n_1} + \frac{e_2(1-e_2)}{n_2}\end{aligned}$$

If the observed difference in the error rate is denoted as  $d = e_1 - e_2$ , then  $d$  is also normally distributed with mean  $d_t$ , its true difference, and variance,  $\sigma_d^2$

Our goal is to test whether the observed difference between  $e_1$  and  $e_2$  is statistically significant.

it can be shown that the **confidence interval for the true difference  $d_t$**  is given by this equation

- At  $(1-\alpha)$  confidence level,

$$d_t = d \pm Z_{\alpha/2} \hat{\sigma}_d$$

# An Illustrative Example

- Given: M1:  $n_1 = 30, e_1 = 0.15$   
M2:  $n_2 = 5000, e_2 = 0.25$
- $d = |e_2 - e_1| = 0.1$  (2-sided test)

In this example, we are performing a two-sided test to check whether  $dt = 0$  or  $dt \neq 0$ .

$$\hat{\sigma}_d^2 = \frac{0.15(1-0.15)}{30} + \frac{0.25(1-0.25)}{5000} = 0.0043$$

- At 95% confidence level,  $Z_{\alpha/2}=1.96$

Estimated variance

$$d_t = 0.100 \pm 1.96 \times \sqrt{0.0043} = 0.100 \pm 0.128$$

=> Interval contains 0 => difference may not be statistically significant

As the interval spans the value zero, we can conclude that the observed difference is not statistically significant at a 95% confidence level.

# Comparing Performance of 2 Algorithms (Classifiers)

---

- Each learning algorithm may produce k models:
  - L1 may produce M<sub>11</sub> , M<sub>12</sub>, ..., M<sub>1k</sub>
  - L2 may produce M<sub>21</sub> , M<sub>22</sub>, ..., M<sub>2k</sub>
- If models are generated on the same test sets D<sub>1</sub>,D<sub>2</sub>, ..., D<sub>k</sub> (e.g., via cross-validation)
  - For each set: compute  $d_j = e_{1j} - e_{2j}$
  - $d_j$  has mean  $d_t$  and variance  $\sigma_t$
  - Estimate:

$$\hat{\sigma}_t^2 = \frac{\sum_{j=1}^k (d_j - \bar{d})^2}{k(k-1)}$$

$$d_t = d \pm t_{1-\alpha, k-1} \hat{\sigma}_t$$

# **Data Mining Classification: Alternative Techniques**

---

Lecture Notes for Chapter 4

Rule-Based

Introduction to Data Mining , 2<sup>nd</sup> Edition

by

Tan, Steinbach, Karpatne, Kumar

# Rule-Based Classifier

---

- Kayıtları “if...then...” kurallarından oluşan bir koleksiyon kullanarak sınıflandırır
- Rule:  $(Condition) \rightarrow y$ 
  - burada
    - ◆ *Condition* (Koşul), özniteliklerin birleşimidir
    - ◆  $y$ , sınıf etiketidir
  - *LHS*: rule antecedent (kural öncülü) or condition (koşul)
  - *RHS*: rule consequent (kural sonucu)
  - Sınıflandırma kurallarına örnekler:
    - ◆ (Blood Type=Warm)  $\wedge$  (Lay Eggs=Yes)  $\rightarrow$  Birds
    - ◆ (Taxable Income < 50K)  $\wedge$  (Refund=Yes)  $\rightarrow$  Evade>No

# Rule-based Classifier (Example)

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds
cat	warm	yes	no	no	mammals
leopard shark	cold	yes	no	yes	fishes
turtle	cold	no	no	sometimes	reptiles
penguin	warm	no	no	sometimes	birds
porcupine	warm	yes	no	no	mammals
eel	cold	no	no	yes	fishes
salamander	cold	no	no	sometimes	amphibians
gila monster	cold	no	no	no	reptiles
platypus	warm	no	no	no	mammals
owl	warm	no	yes	no	birds
dolphin	warm	yes	no	yes	mammals
eagle	warm	no	yes	no	birds

R1: (Give Birth = no)  $\wedge$  (Can Fly = yes)  $\rightarrow$  Birds

R2: (Give Birth = no)  $\wedge$  (Live in Water = yes)  $\rightarrow$  Fishes

R3: (Give Birth = yes)  $\wedge$  (Blood Type = warm)  $\rightarrow$  Mammals

R4: (Give Birth = no)  $\wedge$  (Can Fly = no)  $\rightarrow$  Reptiles

R5: (Live in Water = sometimes)  $\rightarrow$  Amphibians

# Application of Rule-Based Classifier

---

- Bir  $r$  kuralı, ilgili örneğin öznitelikleri (attributes of the instance) kuralın koşulunu karşılıyorsa  $x$  örneğini **kapsar (covers)**

R1: (Give Birth = no)  $\wedge$  (Can Fly = yes)  $\rightarrow$  Birds

R2: (Give Birth = no)  $\wedge$  (Live in Water = yes)  $\rightarrow$  Fishes

R3: (Give Birth = yes)  $\wedge$  (Blood Type = warm)  $\rightarrow$  Mammals

R4: (Give Birth = no)  $\wedge$  (Can Fly = no)  $\rightarrow$  Reptiles

R5: (Live in Water = sometimes)  $\rightarrow$  Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
hawk	warm	no	yes	no	?
grizzly bear	warm	yes	no	no	?

The rule R1 covers a hawk => Bird

The rule R3 covers the grizzly bear => Mammal

# Rule Coverage and Accuracy

- Bir sınıflandırma kuralının **kalitesi**, kapsam (**coverage**) ve doğruluk (**accuracy**) gibi ölçütler kullanılarak değerlendirilebilir.
- Bir veri kümesi  $D$  ve bir sınıflandırma kuralı verildiğinde  $r : A \rightarrow y$ 
  - kuralın kapsamı (**coverage**),  $D$  veri kümesinde  $r$  kuralını tetikleyen kayıtların oranı olarak tanımlanır.
  - Öte yandan, kuralın doğruluğu (**accuracy**) veya güven faktörü (**confidence factor**), sınıf etiketleri  $y$ 'ye eşit olan  $r$  tarafından tetiklenen kayıtların oranı olarak tanımlanır.
  - Bu ölçütlerin tanımları

$$\text{Coverage}(r) = \frac{|A|}{|D|} = \frac{|LHS|}{n}$$

$$\text{Accuracy}(r) = \frac{|A \cap y|}{|A|} = \frac{|LHS \cap RHS|}{|LHS|}$$

Burada  $|A|$  kural öncülünü (**rule antecedent**) karşılayan kayıtların sayısı,  $|A \cap y|$  hem öncülü hem de sonucu (both **antecedent and consequent**) karşılayan kayıtların sayısıdır ve  $|D|$  toplam kayıt sayısıdır.

# Rule Coverage and Accuracy

- **Coverage** of a rule:
  - Bir kuralın öncülünü (*antecedent of a rule*) karşılayan kayıtların oranı
- **Accuracy** of a rule:
  - Bir kuralın sonucunu da öncülünü de karşılayan kayıtların oranı

Tid	Refund	Marital Status	Taxable Income	Class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

(Status=Single) → No

Coverage = 40%, Accuracy = 50%

# How does Rule-based Classifier Work?

---

R1: (Give Birth = no)  $\wedge$  (Can Fly = yes)  $\rightarrow$  Birds

R2: (Give Birth = no)  $\wedge$  (Live in Water = yes)  $\rightarrow$  Fishes

R3: (Give Birth = yes)  $\wedge$  (Blood Type = warm)  $\rightarrow$  Mammals

R4: (Give Birth = no)  $\wedge$  (Can Fly = no)  $\rightarrow$  Reptiles

R5: (Live in Water = sometimes)  $\rightarrow$  Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

A lemur triggers rule R3, so it is classified as a mammal

A turtle triggers both R4 and R5

A dogfish shark triggers none of the rules

# Two important properties of the rule set

- Önceki örnek, kurala dayalı bir sınıflandırıcı tarafından oluşturulan kural kümesinin iki önemli özelliğini göstermektedir.
- **Mutually Exclusive Rules (Karşılıklı/Birbirlerini Dışlayan Kurallar)**
  - Eğer aynı kayıt tarafından *R*'deki herhangi iki kural (no two rules) tetiklenmezse, bu *R* kural kümesindeki kurallar birbirini dışlar (**mutually exclusive**).
  - Bu özellik, her kaydın *R*de en fazla bir kural tarafından kapsamasını sağlar. Karşılıklı dışlayıcı bir kural kümesi örneği Tablo 5.3'te gösterilmiştir.
- **Exhaustive Rules (Eksiksiz/Kapsamlı Kurallar)**
  - Her öznitelik değeri kombinasyonu için bir kural varsa, bir *R* kural grubu eksiksiz kapsamaya(**exhaustive coverage**) sahiptir.
  - Bu özellik, her kaydın *R*'deki **en az** bir kural tarafından kapsamasını sağlar. «Body Temperature» ve «Gives Birth» özniteliklerinin ikili değişkenler olduğunu varsayırsak, Tablo 5.3'te gösterilen kural seti eksiksiz bir kapsamaya sahiptir.

Table 5.3. Example of a mutually exclusive and exhaustive rule set.

$r_1$ : (Body Temperature = cold-blooded)  $\longrightarrow$  Non-mammals

$r_2$ : (Body Temperature = warm-blooded)  $\wedge$  (Gives Birth = yes)  $\longrightarrow$  Mammals

$r_3$ : (Body Temperature = warm-blooded)  $\wedge$  (Gives Birth = no)  $\longrightarrow$  Non-mammals

# Characteristics of Rule Sets: Strategy 1

---

- Mutually exclusive rules
  - Kurallar birbirinden bağımsızsa sınıflandırıcı, birbirini dışlayan kurallar içerir
  - Her kayıt en fazla bir kural tarafından kapsanır
- Exhaustive rules
  - Sınıflandırıcı, öznitelik değerlerinin her olası kombinasyonunu hesaba katıyorsa eksiksiz bir kapsamaya sahiptir
  - Her kayıt en az bir kural tarafından kapsamaktadır.

# Characteristics of Rule Sets: Strategy 2

---

- Rules are not mutually exclusive
  - Bir kayıt birden fazla kuralı tetikleyebilir
  - Çözüm?
    - ◆ *Ordered rule set*
    - ◆ *Unordered rule set – use voting schemes*
- Rules are not exhaustive
  - Bir kayıt herhangi bir kuralı tetiklemeyebilir
  - Çözüm?
    - ◆ *Use a default class*

# Ordered Rule Set

Öncelik (**priority**) birçok şekilde tanımlanabilir (ör. doğruluk, kapsam, toplam açıklama uzunluğu veya kuralların oluşturulma sırasına göre).

- Kurallar, önceliğe (**priority**) göre sıralanır
  - Sıralı bir kural kümesi (ordered rule set), karar listesi olarak bilinir
- Sınıflandırıcıya bir test kaydı sunulduğunda
  - Tetiklediği en yüksek dereceli kuralın sınıf etiketine atanır
  - Kurallardan hiçbirini tetiklenmediyse, varsayılan sınıfı (*default class*) atanır

R1: (Give Birth = no)  $\wedge$  (Can Fly = yes)  $\rightarrow$  Birds

R2: (Give Birth = no)  $\wedge$  (Live in Water = yes)  $\rightarrow$  Fishes

R3: (Give Birth = yes)  $\wedge$  (Blood Type = warm)  $\rightarrow$  Mammals

R4: (Give Birth = no)  $\wedge$  (Can Fly = no)  $\rightarrow$  Reptiles

R5: (Live in Water = sometimes)  $\rightarrow$  Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
turtle	cold	no	no	sometimes	?

# Unordered Rule Set

---

- Bu yaklaşım, bir test kaydının **birden çok sınıflandırma kuralını tetiklemesine** izin verir ve her kuralın sonucunu **belirli bir sınıf için bir oy (vote)** olarak değerlendirir.
- Oylar daha sonra test kaydının sınıf etiketini belirlemek için toplanır.
- Kayıt genellikle en çok oyu (**highest number of votes**) alan sınıfına atanır.
- Bazı durumlarda oylama, kuralın doğruluğuna göre ağırlıklandırılabilir.

# Rule Ordering Schemes

- Rule-based ordering
  - Bireysel kurallar kalitelerine göre sıralanır
- Class-based ordering
  - Aynı sınıfı ait kurallar birlikte görünür

## Rule-based Ordering

(Refund=Yes) ==> No

(Refund=No, Marital Status={Single,Divorced},  
Taxable Income<80K) ==> No

(Refund=No, Marital Status={Single,Divorced},  
Taxable Income>80K) ==> Yes

(Refund=No, Marital Status={Married}) ==> No

## Class-based Ordering

(Refund=Yes) ==> No

(Refund=No, Marital Status={Single,Divorced},  
Taxable Income<80K) ==> No

(Refund=No, Marital Status={Married}) ==> No

(Refund=No, Marital Status={Single,Divorced},  
Taxable Income>80K) ==> Yes

# Rule-based ordering scheme

---

- bireysel kuralları bazı kural kalite ölçülerine göre sıralar.
- her test kaydının, kendisini kapsayan "en iyi" kurala göre sınıflandırılmasını sağlar.
- potansiyel bir dezavantajı var
  - Daha düşük dereceli kuralları yorumlamak çok daha zordur çünkü kendilerinden önceki kuralların değilmesini varsayarlar.

# Class-based ordering scheme

---

- Bu yaklaşımda, **aynı sınıfa** ait olan kurallar,  $R$  kural kümesinde **birlikte görünür**.
- Kurallar daha sonra **sınıf bilgilerine** göre **toplu olarak sıralanır**.
- Aynı sınıfındaki kurallar arasında **göreceli sıralama** önemli değildir;
  - kurallardan biri tetiklendiği sürece, sınıf test kaydına atanacaktır.
- Bu, kural yorumlamasını biraz daha kolaylaştırır.
- İyi bilinen kural tabanlı (*rule-based*) sınıflandırıcıların çoğu (**C4.5rules** ve **RIPPER** gibi) sınıf tabanlı sıralama şemasını (**class-based ordering scheme**) kullandığından, bu bölümün geri kalanındaki tartışma esas olarak bu tür sıralama şemasına odaklanmaktadır.

# Building Classification Rules

Sınıflandırma kuralları çıkarımı için iki geniş yöntem sınıfı vardır :

- Direct Method:

- ◆ Kuralları doğrudan verilerden çıkarır
- ◆ Örnekler: RIPPER, CN2, Holte's 1R

- Indirect Method:

- ◆ Kuralları diğer sınıflandırma modellerinden çıkarır (e.g. decision trees, neural networks, etc).
- ◆ Örnekler : C4.5rules

Doğrudan yöntemler (Direct methods), öznitelik uzayını daha küçük altuzaylara böler, böylece bir alt-uzaya ait olan tüm kayıtlar **tek bir sınıflandırma kuralı** kullanılarak **sınıflandırılabilir**.

# Direct Method: Sequential Covering

---

1. Start from an empty rule
2. Grow a rule using the Learn-One-Rule function
3. Remove training records covered by the rule
4. The rule is added to the bottom of the decision list  $R$ .
5. Repeat Step (2) and (3) until stopping criterion is met

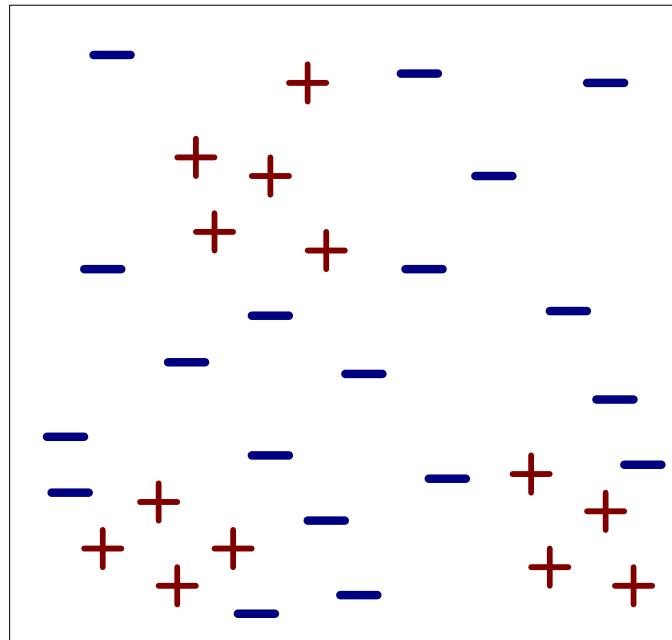
A rule is desirable if it covers most of the positive examples and none (or very few) of the negative examples.

## Learn-One-Rule Function

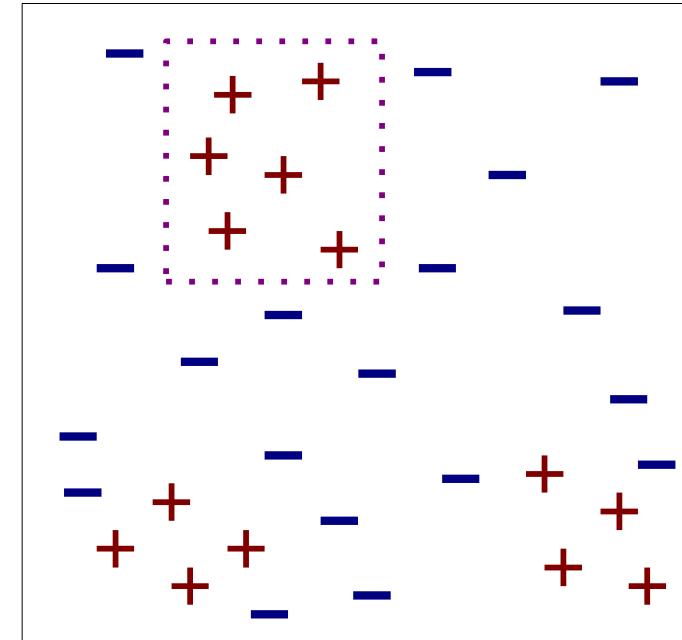
Learn-One-Rule fonksiyonunun amacı, eğitim setindeki pozitif örneklerin çoğunu ve negatif örneklerin hiçbirini (veya çok azını) kapsayan bir sınıflandırma kuralı çıkarmaktır. Bununla birlikte, arama uzayının üstel boyutu göz önüne alındığında, optimal bir kural bulmak hesaplama açısından pahalıdır.

# Example of Sequential Covering

Pozitif örneklerin en büyük kısmını kapsadığı için önce R1 kuralı çıkarılır.



(i) Original Data

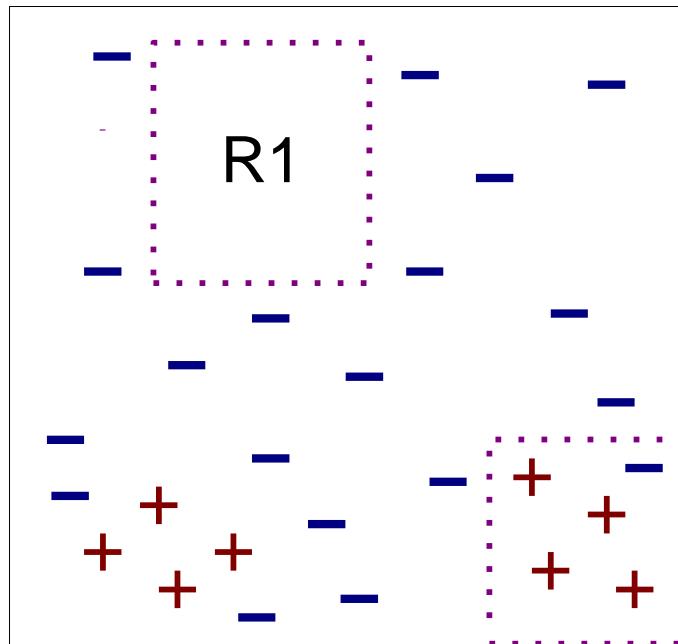


(ii) Step 1

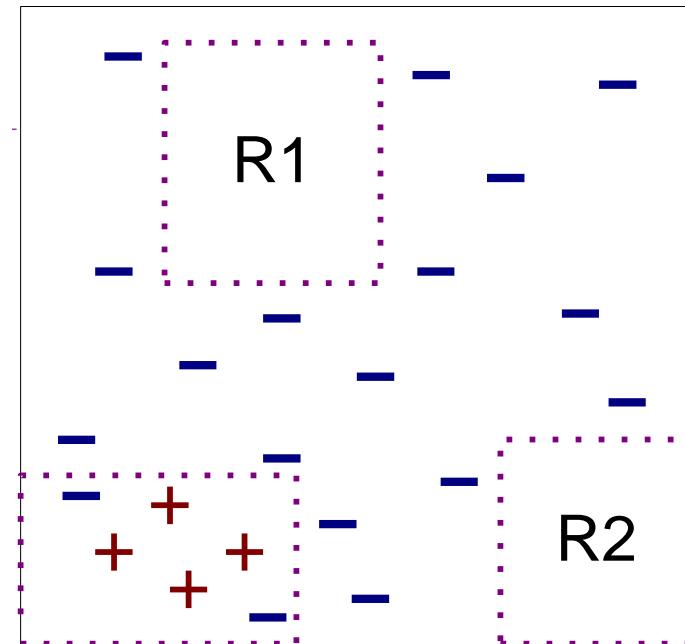
Böyle bir kural bulunduğunda, kuralın kapsadığı eğitim kayıtları elimine edilir.

# Example of Sequential Covering...

*R1* tarafından kapsanan tüm eğitim kayıtları kaldırılır ve daha sonra algoritma bir sonraki en iyi kuralı, yani *R2*'yi aramaya devam eder.



(iii) Step 2

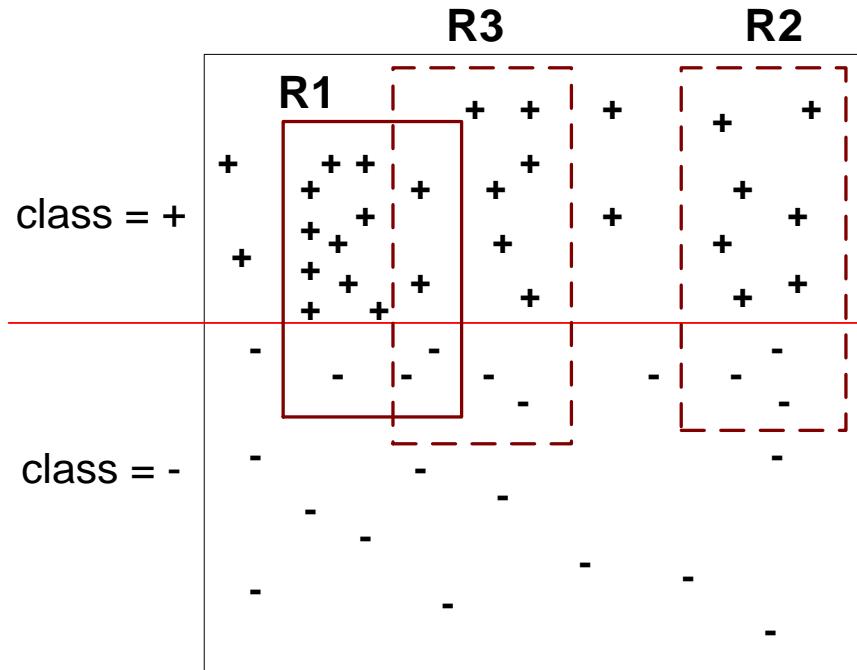


(iv) Step 3

Böyle bir kural bulunduğuunda, kuralın kapsadığı eğitim kayıtları elimine edilir.

# Instance Elimination

- Örnekleri (*instances*) neden ortadan kaldırılmamız gerekiyor?
  - Aksi takdirde, sonraki kural önceki kuralla aynı olur
- Pozitif örnekleri neden kaldırıyoruz?
  - Sonraki kuralın farklı olmasını sağlama alıyor
- Negatif örnekleri neden kaldırıyoruz?
  - Kuralın doğruluğunun eksik değerlemesini öner
  - Diyagramda R2 ve R3 kurallarını karşılaştırın



**Şekil 5.4.** «sequential covering algorithm» ile eğitim kayıtlarının ortadan kaldırılması (elimination). R1, R2 ve R3, üç farklı kural tarafından kapsanan bölgeleri temsil eder

Şekil 5.4, 29 olumlu örnek ve 21 olumsuz örnek içeren bir veri kümesinden çıkarılan üç olası kuralı, R1, R2 ve R3'ü gösterir. **R1**, **R2** ve **R3**'ün doğrulukları sırasıyla 12/15 (%80), 7/10 (%70) ve 8 / 12'dir (%66.7). En yüksek doğruluğa sahip olduğu için ilk önce R1 oluşturulur.

# Instance Elimination

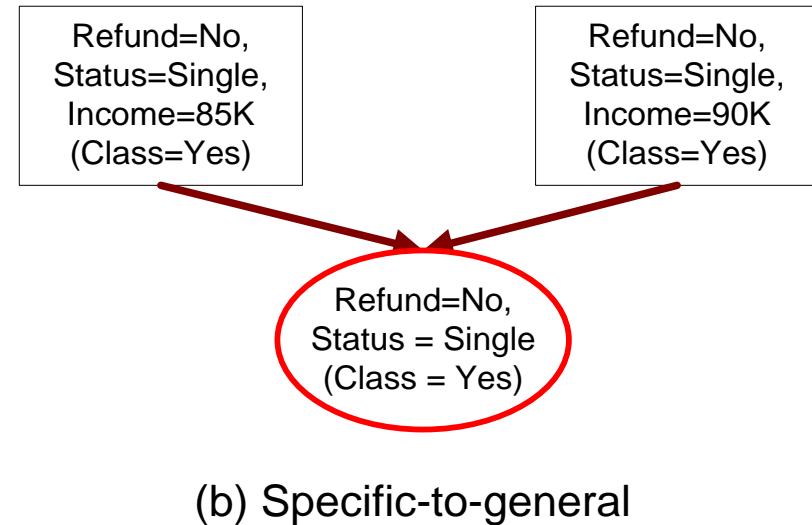
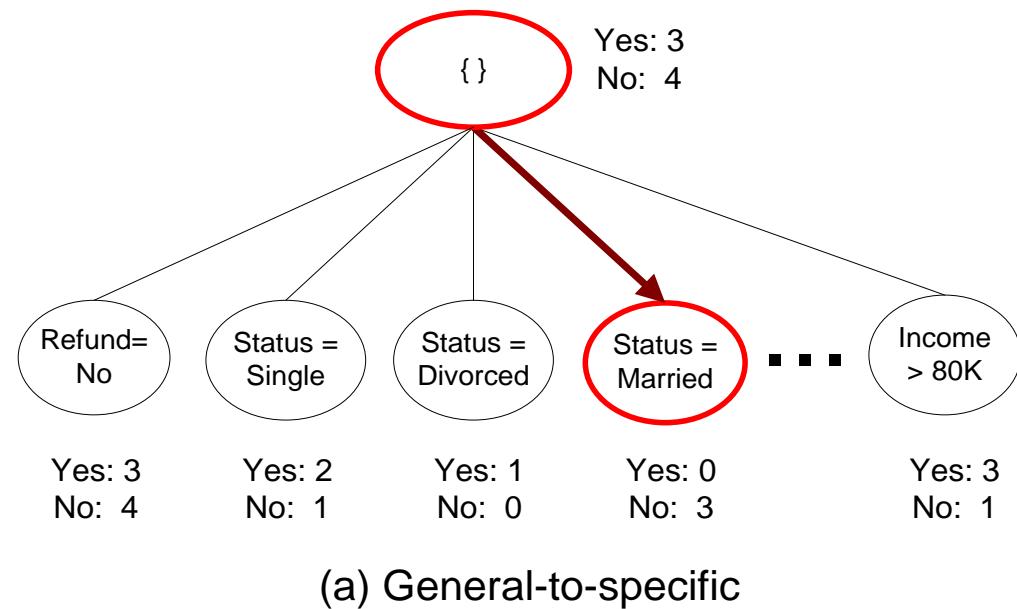
---

$R_1$ ,  $R_2$  ve  $R_3$ 'ün doğrulukları sırasıyla 12/15 (%80), 7/10 (%70) ve 8 / 12'dir (%66.7). En yüksek doğruluğa sahip olduğu için ilk önce  $R_1$  oluşturulur.

En yüksek doğruluğa sahip olduğu için önce  $R_1$  oluşturulur.  $R_1$  oluşturuluduktan sonra, algoritma tarafından oluşturulan bir sonraki kuralın  $R_1$ 'den farklı olması için kuralın kapsadığı pozitif örneklerin kaldırılması gerektiği açıktır. Ardından, algoritmaya  $R_2$  veya  $R_3$  üretme seçeneği verildiğini varsayıyalım.  $R_2$ ,  $R_3$ 'ten daha yüksek doğruluğa sahip olsa da,  $R_1$  ve  $R_3$  birlikte 18 olumlu örneği ve 5 olumsuz örneği kapsar (genel olarak %78,3 doğrulukla sonuçlanır), oysa  $R_1$  ve  $R_2$  birlikte 19 olumlu örneği ve 6 olumsuz örneği kapsar (genel bir sonuçla sonuçlanır). %76 doğruluk).  $R_2$  veya  $R_3$ 'ün doğruluk üzerindeki artan etkisi, doğrulukları hesaplanmadan önce  $R_1$  kapsamındaki olumlu ve olumsuz örnekler kaldırıldığında daha belirgindir. Özellikle,  $R_1$ 'in kapsadığı olumlu örnekler çıkarılmazsa,  $R_3$ 'ün etkin doğruluğunu abartabiliriz ve olumsuz örnekler çıkarılmazsa,  $R_3$ 'ün doğruluğunu küçümseyebiliriz. İkinci durumda,  $R_3$  tarafından işlenen yanlış pozitif hataların yarısı bir önceki kural olan  $R_1$  tarafından zaten açıklanmış olsa bile,  $R_2$ 'yi  $R_3$ 'e tercih edebiliriz.

# Rule Growing

- Two common strategies



# Recall: How to determine the Best Split

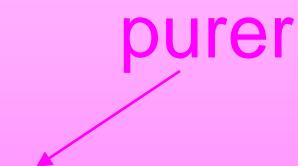
- | Greedy approach:
  - Nodes with **purer** class distribution are preferred
- | Need a measure of node impurity:

C0: 5
C1: 5

High degree of impurity

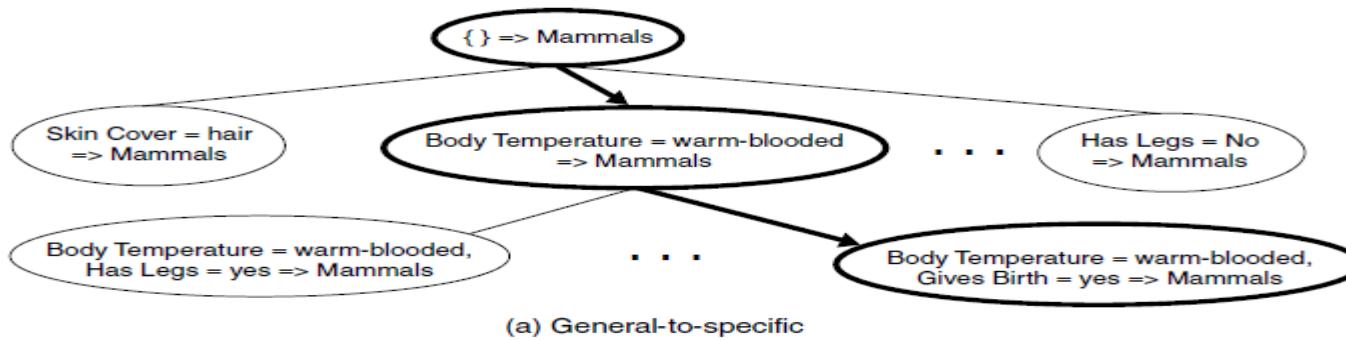
C0: 9
C1: 1

Low degree of impurity



# Rule Growing strategy: general-to-specific

- Bir başlangıç kuralı  $r : \{ \} \rightarrow y$  oluşturulur, burada sol taraf boş bir kümedir ve sağ taraf hedef sınıfı (*target class*) içerir.
- Bu haliyle kuralın kalitesi düşük (**poor quality**) çünkü eğitim setindeki tüm örnekleri kapsar.
- Kuralın kalitesini iyileştirmek için sonradan yeni bağlaçlar (**new conjuncts**) eklenir
- **Body Temperature=warm-blooded** bağlacı kural öncülünü oluşturmak için başlangıçta seçilir.
- Algoritma daha sonra olası tüm adayları araştırır ve greedy mantığıyla bir sonraki bağlacı, **Gives Birth=yes** kural öncülüne eklenecek olarak seçer.

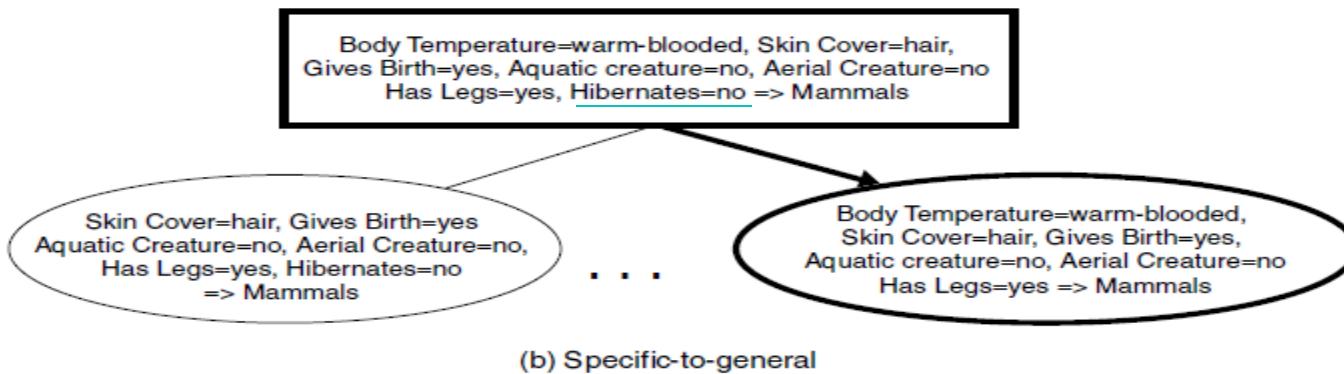


The general-to-specific rule-growing strategy for the vertebrate classification problem.

Bu süreç, durma kriteri karşılanana kadar devam eder (örneğin, eklenen bağlaç **kuralın kalitesini iyileştirmede**ğinde).

# Rule Growing strategy: specific-to-general

- Pozitif örneklerden biri, kural geliştirme süreci için ilk çekirdek (**initial seed**) olarak rastgele seçilir.
- İyileştirme adımı sırasında kural, bağlaçlarından biri kaldırılarak genelleştirilir
  - böylece **daha fazla sayıda pozitif örnekleri** kapsayabilir.
- İlk çekirdek olarak memeliler için pozitif bir örnek seçildiğini varsayıyalım.
  - Başlangıç kuralı, çekirdek öznitelik değerleriyle aynı bağlaçları içerir.
- Kapsamını (coverage) iyileştirmek için, kural **Hibernate=no** bağlacını kaldırılarak genelleştirilir.



Rafine etme adımı, durma kriteri karşılanana kadar tekrar edilir, örneğin, **kural negatif örnekleri kapsamaya başladığında**

the specific-to-general approach for the vertebrate classification problem.

# Rule Evaluation

---

- Kural geliştirme sürecinde hangi bağlacın eklenmesi (veya kaldırılması) gerektiğini belirlemek için bir değerlendirme ölçüyü gereklidir.
- **Accuracy** bariz bir seçimdir çünkü kural tarafından doğru bir şekilde sınıflandırılan eğitim örneklerinin oranını açık bir şekilde ölçer.
- Ancak, **accuracy**'nin **olası handikapı** şudur:
  - kuralın kapsamını (coverage) hesaba katmaz.
- Örneğin, aşağıdakileri içeren bir eğitim seti düşünün:
  - 60 pozitif örnek ve
  - 100 negatif örnek.
- Aşağıdaki iki aday kuralın bize verildiğini varsayıalım:
  - Rule  $r1$ : 50 pozitif örnek ve 5 negatif örnek içerir,
  - Rule  $r2$ : 2 pozitif örneği içerir ve hiç negatif örnek içermez

# Rule Evaluation

---

- Rule  $r_1$  : covers 50 positive examples and 5 negative examples,
  - Rule  $r_2$  : covers 2 positive examples and no negative examples.
- 
- $r_1$  ve  $r_2$  için doğruluklar (accuracies) sırasıyla % 90,9 ve % 100'dür.
  - Ancak, doğruluğu düşük olmasına rağmen  $r_1$  daha iyi bir kuraldır.
  - $r_2$  için yüksek doğruluk potansiyel olarak sahtedir (**spurious**) çünkü **kuralın kapsamı (coverage)** çok düşüktür.
  - Bunu halletmek için aşağıdaki yaklaşım kullanılabilir

# Rule Evaluation

- Foil's Information Gain

**FOIL: First Order Inductive Learner – an early rule-based learning algorithm**

- R0: {} => class (initial rule)
  - R1: {A} => class (rule after adding conjunct)
  - $\text{Gain}(R0, R1) = t \left( \log_2 \frac{p1}{p1+n1} - \log_2 \frac{p0}{p0+n0} \right)$
  - where t: number of positive instances covered by both R0 and R1  
p0: number of positive instances covered by R0  
n0: number of negative instances covered by R0  
p1: number of positive instances covered by R1  
n1: number of negative instances covered by R1
- Since the measure is proportional to  $p_1$  and  $p_1/(p_1+n_1)$ , it prefers rules that have high support count and accuracy.

# Rule Evaluation example by FOIL's info. gain

$$\text{FOIL's information gain} = p_1 \times \left( \log_2 \frac{p_1}{p_1 + n_1} - \log_2 \frac{p_0}{p_0 + n_0} \right)$$

p0=60, n0=100

**Rule  $r_1$**  : covers 50 positive examples and 5 negative examples, i.e.

**p1=50, n1=5**

**FOIL's info gain=**

$$50 \left( \log_2 \frac{50}{55} - \log_2 \frac{60}{160} \right) = 63,87$$

p0=60, n0=100

**Rule  $r_2$**  : covers 2 positive examples and no negative examples, i.e.

**p1=2, n1=0**

**FOIL's info gain=**

$$2 \left( \log_2 \frac{2}{2} - \log_2 \frac{60}{160} \right) = 2,83$$

Therefore,  $r_1$  is a better rule than  $r_2$ .

# Stopping Criterion and Rule Pruning

---

- Stopping criterion
  - Kazancı (*gain*) hesaplayın
  - Kazanç ihmal edilebilecek kadar küçük ise, yeni kuralı atın
- Rule Pruning
  - Karar ağaçlarının sonradan budamasına (*post-pruning*) benzer
  - Reduced Error Pruning:
    - ◆ Kuraldaki bağlaçlardan birini kaldırın
    - ◆ Budamadan önce ve sonra doğrulama (*on validation set before and after pruning*) setindeki hata oranını karşılaştırın
    - ◆ Hata iyileşirse, bağlacı budayın

# Summary of Direct Method

---

---

- Grow a single rule
- Remove Instances from rule
- Prune the rule (if necessary)
- Add rule to Current Rule Set
- Repeat

# Direct Method: RIPPER

---

- RIPPER, yaygın olarak kullanılan bir kural indükleme (*induction*) algoritmasıdır.
- Bu algoritma, eğitim örneklerinin sayısıyla neredeyse doğrusal olarak ölçeklenir ve
  - özellikle dengesiz sınıf dağılımlarına (**imbalanced class distributions**) sahip veri kümelerinden modeller oluşturmak için uygundur.
- RIPPER, modelin ezberlemesini önlemek için bir doğrulama seti kullandığından gürültülü veri kümeleriyle (**noisy data sets**) de iyi çalışır.

# Direct Method: RIPPER

---

- 2 sınıfı problem için, sınıflardan birini pozitif sınıf olarak seçin ve diğerini negatif sınıf olarak
  - Pozitif sınıf için kuralları öğrenin (**the minority**)
  - Negatif sınıf (**the majority**) varsayılan sınıf olacaktır
- Çok sınıfı (*multi-class*) problem için
  - Sınıfları artan sınıf yaygınlığına göre sıralayın (belirli bir sınıf'a ait örneklerin oranı, yani **frekans**)
  - Önce en küçük sınıf için kural kümesini öğrenin, gerisini negatif sınıf olarak değerlendirin
  - Pozitif sınıf olarak sonraki en küçük sınıfı tekrarlayın

# Direct Method: RIPPER

---

- Growing a rule:
  - Boş kuraldan başlayın (RIPPER, genelden özele (*general-to-specific strategy*) bir strateji kullanır)
  - FOIL'ın bilgi kazancını iyileştirdikleri sürece bağlaçlar ekleyin
  - Kural artık negatif örnekleri kapsamadığında eklemeyi durdurun
  - «incremental reduced error pruning» kullanarak kuralı hemen budayın (Yeni kural daha sonra doğrulama setindeki performansına göre budanır.)
  - Measure for pruning:  $v = (p-n)/(p+n)$ 
    - ◆ p: validation setinde kuralın kapsadığı pozitif örneklerin sayısı
    - ◆ n: validation setinde kuralın kapsadığı negatif örneklerin sayısı
  - Pruning method: v'yi en maksimize eden son koşullar dizisini silin (Budama, kurala eklenen son bağlaçtan başlayarak yapılır.)

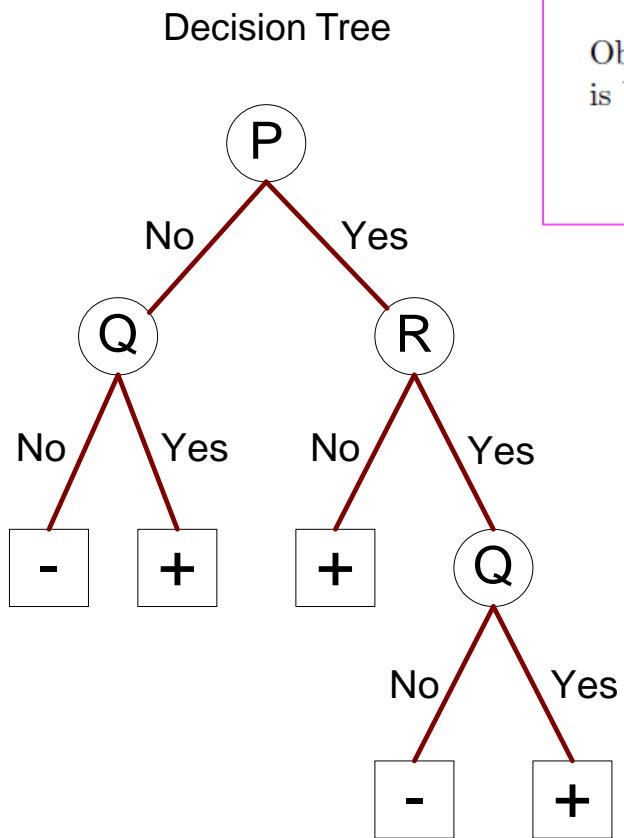
# Direct Method: RIPPER

---

- Building a Rule Set:
  - Use sequential covering algorithm
    - ◆ Mevcut pozitif örnekler kümesini kapsayan en iyi kuralı bulur
    - ◆ Kuralın kapsadığı hem pozitif hem de negatif örnekleri ortadan kaldırır
  - Kural kümesine her kural eklendiğinde, yeni açıklama uzunluğunu (description length) hesaplanır
    - ◆ Yeni açıklama uzunluğu şimdije kadar elde edilen en küçük açıklama uzunluğundan  $d$  bit daha uzun olduğunda yeni kural eklemeyi durdurur

Yeni kural, kural kümesinin toplam açıklama uzunluğunu en az  $d$  bit arttırırsa, RIPPER kural kümesine kural eklemeyi durdurur (varsayılan olarak,  $d$  64 bit seçilir).

# Indirect Methods



Example 5.2. Consider the following three rules from Figure 5.5:

$$r2 : (P = \text{No}) \wedge (Q = \text{Yes}) \rightarrow +$$

$$r3 : (P = \text{Yes}) \wedge (R = \text{No}) \rightarrow +$$

$$r5 : (P = \text{Yes}) \wedge (R = \text{Yes}) \wedge (Q = \text{Yes}) \rightarrow +$$

Observe that the rule set always predicts a positive class when the value of  $Q$  is Yes. Therefore, we may simplify the rules as follows:

$$r2' : (Q = \text{Yes}) \rightarrow +$$

$$r3 : (P = \text{Yes}) \wedge (R = \text{No}) \rightarrow +.$$

## Rule Set

$$r1 : (P=\text{No}, Q=\text{No}) \implies -$$

$$r2 : (P=\text{No}, Q=\text{Yes}) \implies +$$

$$r3 : (P=\text{Yes}, R=\text{No}) \implies +$$

$$r4 : (P=\text{Yes}, R=\text{Yes}, Q=\text{No}) \implies -$$

$$r5 : (P=\text{Yes}, R=\text{Yes}, Q=\text{Yes}) \implies +$$

Converting a decision tree into classification rules.

# Indirect Method: C4.5rules

---

- Budanmamış bir karar ağacından kuralları çıkarır
- Her kural için,  $r: A \rightarrow y$ ,
  - alternatif bir kural düşünün  $r': A' \rightarrow y$ . Burada  $A'$  başlangıçtaki kuraldaki bağlaçlardan birinin kaldırılmasıyla elde edilir.
  - $r'$ nin karamsar hata oranını (*pessimistic error rate*) tüm  $r'$ lere karşılaştırın
  - Alternatif kurallardan biri daha düşük karamsar hata oranına sahipse budayın
  - Genelleme hatasını (*generalization error*) artık iyileştiremeye kadar tekrarlayın

# Indirect Method: C4.5rules

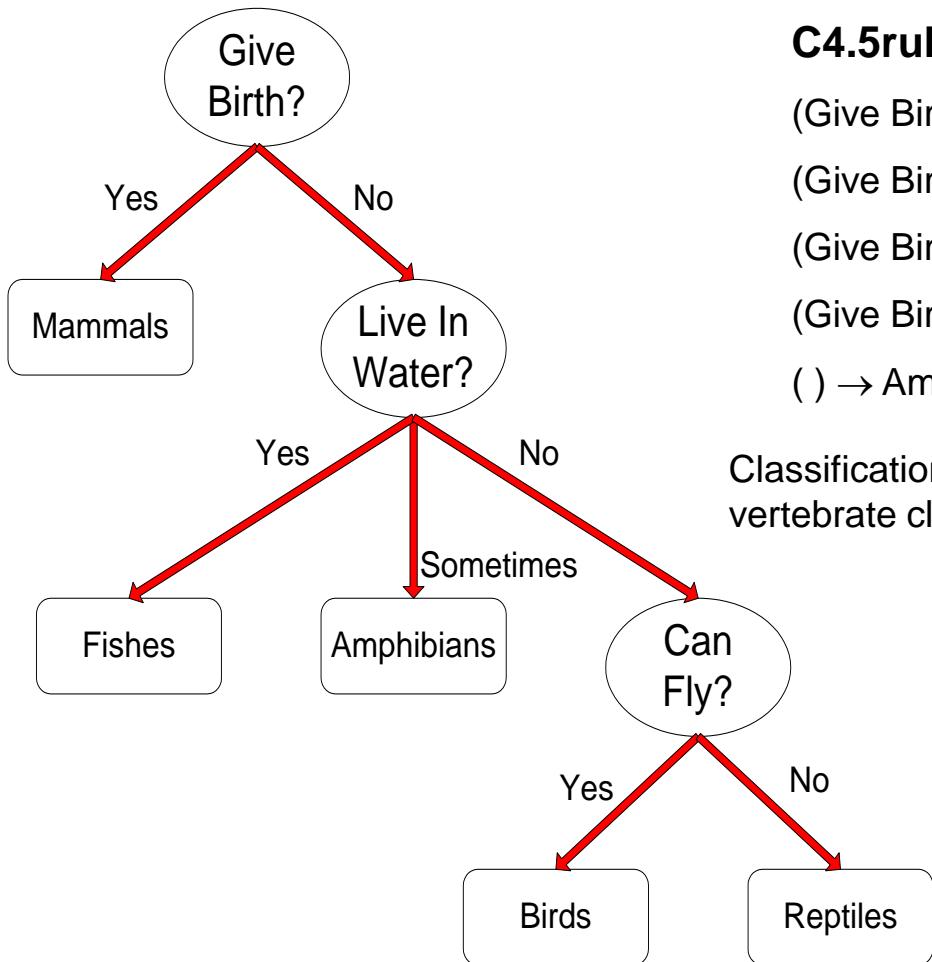
---

- Kuralları sıralamak yerine, kuralların alt kümelerini sıralayın (**class ordering**)
  - Her alt küme, aynı kural sonucunu (class) içeren bir kurallar koleksiyonudur.
  - Her alt kümenin açıklama uzunluğunu hesaplayın( description length)
    - ◆ Description length =  $L(\text{error}) + g L(\text{model})$
    - ◆  $g$ , bir kural kümesindeki gereksiz özniteliklerin varlığını hesaba katan bir parametredir (varsayılan değer = 0,5)

# Example

Name	Give Birth	Lay Eggs	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	no	yes	mammals
python	no	yes	no	no	no	reptiles
salmon	no	yes	no	yes	no	fishes
whale	yes	no	no	yes	no	mammals
frog	no	yes	no	sometimes	yes	amphibians
komodo	no	yes	no	no	yes	reptiles
bat	yes	no	yes	no	yes	mammals
pigeon	no	yes	yes	no	yes	birds
cat	yes	no	no	no	yes	mammals
leopard shark	yes	no	no	yes	no	fishes
turtle	no	yes	no	sometimes	yes	reptiles
penguin	no	yes	no	sometimes	yes	birds
porcupine	yes	no	no	no	yes	mammals
eel	no	yes	no	yes	no	fishes
salamander	no	yes	no	sometimes	yes	amphibians
gila monster	no	yes	no	no	yes	reptiles
platypus	no	yes	no	no	yes	mammals
owl	no	yes	yes	no	yes	birds
dolphin	yes	no	no	yes	no	mammals
eagle	no	yes	yes	no	yes	birds

# C4.5rules versus RIPPER



## C4.5rules:

- (Give Birth=No, Can Fly=Yes) → Birds
- (Give Birth=No, Live in Water=Yes) → Fishes
- (Give Birth=Yes) → Mammals
- (Give Birth=No, Can Fly>No, Live in Water>No) → Reptiles
- ( ) → Amphibians

Classification rules extracted from a decision tree for the vertebrate classification problem

## RIPPER:

- (Live in Water=Yes) → Fishes
- (Have Legs>No) → Reptiles
- (Give Birth=No, Can Fly>No, Live In Water>No) → Reptiles
- (Can Fly=Yes, Give Birth=No) → Birds
- ( ) → Mammals

# C4.5 versus C4.5rules versus RIPPER

C4.5 and C4.5rules:

		PREDICTED CLASS				
		Amphibians	Fishes	Reptiles	Birds	Mammals
ACTUAL CLASS	Amphibians	2	0	0	0	0
	Fishes	0	2	0	0	1
	Reptiles	1	0	3	0	0
	Birds	1	0	0	3	0
	Mammals	0	0	1	0	6

RIPPER:

		PREDICTED CLASS				
		Amphibians	Fishes	Reptiles	Birds	Mammals
ACTUAL CLASS	Amphibians	0	0	0	0	2
	Fishes	0	3	0	0	0
	Reptiles	0	0	3	0	1
	Birds	0	0	1	2	1
	Mammals	0	2	1	0	4

# Advantages of Rule-Based Classifiers

---

- Karar ağaçlarına oldukça benzer özelliklere sahiptir
  - Karar ağaçları kadar ifade gücü yüksek
  - Yorumlaması kolay
  - Yeni örnekleri hızla sınıflandırabilir
  - (fakat) performansı Karar ağaçlarıyla kıyaslanabilir
  - Gereksiz özniteliklerle başa çıkabilir
- Dengesiz sınıfların üstesinden gelmek için daha uygun (imbalanced classes)
- Test setindeki eksik değerler ile başa çıkmak daha zor