504553 - VERİ MADENCİLİĞİ

Yrd.Doç.Dr.Abdullah BAYKAL

Konular

- Veri madenciliğine giriş
- Veri hazırlama ve temizleme
- Sınıflandırma teknikleri
- Demetleme teknikleri
- Bağıntı kuralları
- Sıralı diziler
- Veri madenciliği uygulamaları

Giriş- Problem Tanımı

- Teknolojinin gelişimiyle bilgisayar ortamında ve veritabanlarında tutulan veri miktarının da artması
  - bu veriyi nasıl kullanacağımızı bilmiyoruz
  - saklanan veriden bilgi elde etmek için bu veriyi yorumlamamız gerekiyor
- Kullanıcıların beklentilerinin artması
  - basit veritabanı sorgulama yöntemlerinin yeterli olmaması
- Veri madenciliği yöntemleri fazla miktardaki veri içinden yararlı bilgiyi bulmak için kullanılır.

Bilgi Keşfi

- Büyük veri kaynaklarından yararlı ve ilginç bilgiyi bulmak
- Bulunan bilgi
  - **Gizli**
  - **Önemli**
  - **Önceden bilinmeyen**
  - **Yararlı**

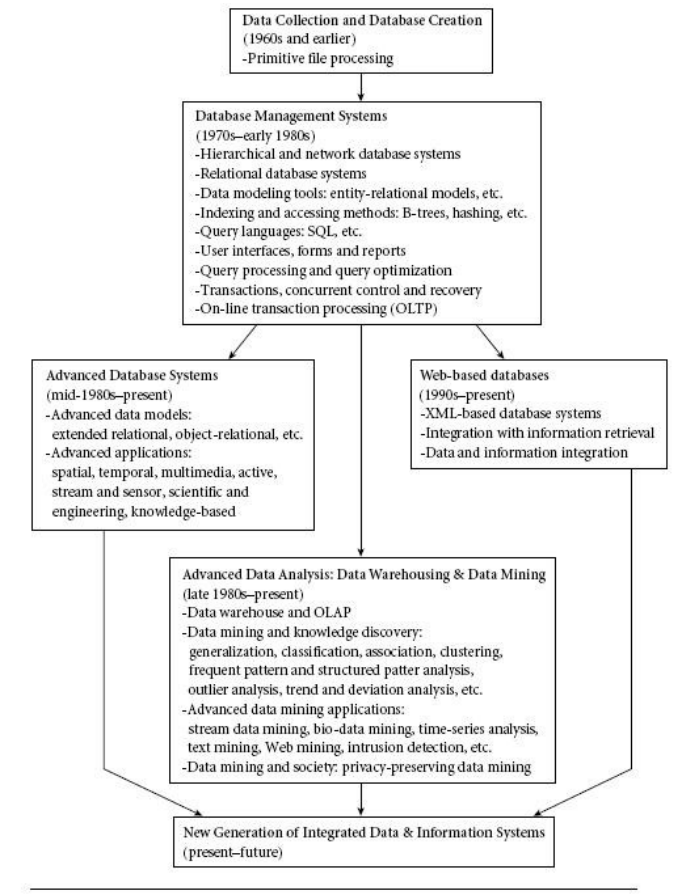
Neden Veri Madenciliği?

- Neden veri madenciliği?
  - Artan veri miktarı –terabyte to petabyte Teknolojinin gelişimiyle bilgisayar ortamında ve veritabanlarında tutulan veri miktarının da artması
    - Yeni veri toplama yolları Otomatik veri toplama aletleri, veritabanı sistemleri, bilgisayar kullanımının artması
    - Büyük veri kaynakları İş dünyası: Web, e-ticaret, alışveriş, hisse senetleri, ...
    - Bilim dünyası: Uzaktan algılama ve izleme, bioinformatik, simülasyonlar..
    - Toplum: haberler,digital kameralar, YouTube, Facebook...
- Veri içinde boğuluyoruz, ancak bilgi elde edemiyoruz!!!

VERİ MADENCİLİĞİ NEDİR?

- Veri madenciliği basit ve açık olmayan, önceden bilinmeyen ve yararlı olan desenlerin ya da bilginin çok büyük miktarlardaki veriden çıkarılması
- Veri madenciliği teknikleri, veri içindeki örüntüleri bulur
  - örüntü: veri içindeki herhangi bir yapı
- Sorgulama ya da basit istatistik yöntemler veri madenciliği değildir
- Veri madenciliği teknikleri veriyi belli bir modele uydurur.

Veri Madenciliği - Tarihçe



Veri Madenciliğinin Amacı

- **Ne yapmak istemiyoruz?**
  - Büyük miktardaki veri içinde arama yapmak (Veritabanı yönetim sistemleri bu işi yapıyor)
- **Veri madenciliğinin amacı:**
  - Aradığımız veri mevcutsa bu veriden çıkarabileceğimiz sonuçlarını anlamak

Veri Tabanı & Veri Madenciliği İşlemleri

Veritabanı		Veri Madenciliği
Sorgulama	&	Sorgulama
Tanımlı		Tam tanımlı değil
SQL		yaygın sorgulama dili yok
Veri	&	Veri
Canlı veri		Üzerinde işlem yapılmayan veri
Çıkış	&	Çıkış
Belirli		Belirli değil
verinin bir alt kümesi		verinin bir alt kümesi değil

Sorgulama Örnekleri

- Veritabanı uygulaması:
  - Soyadı Gündüz olan kredi kartı sahiplerini bul.
  - Bir ayda 2000 YTL'den fazla harcama yapan kredi kartı sahiplerini bul.
  - DVD satın alan tüm müşterileri bul.

- Veri madenciliği uygulaması
  - Riski az olan tüm kredi kartı başvurularını bul (sınıflandırma)
  - Harcama alışkanlığı benzer olan kredi kartı sahiplerini bul (demetleme)
  - DVD birlikte sıkça satın alınan ürünü bul (bağlantı kuralları)

Veri Madenciliği Uygulama Alanları

- Veri tabanı analizi ve karar verme desteği
  - Pazar araştırması
  - Hedef Pazar, müşteriler arası benzerliklerin saptanması, sepet
  - analizi, çapraz pazar incelemesi
- Risk analizi
  - Kalite kontrolu, rekabet analizi, öngörü
  - Sahtekarlıkların saptanması
- Diğer Uygulamalar
  - Belgeler arası benzerlik (haber kümeleri, e-posta)
  - Sorgulama Sonuçları

Pazar Araştırması - 1

- Veri madenciliği uygulamaları için veri kaynağı
  - Kredi kartı hareketleri, üyelik kartları, ucuzluk kuponları,pazar anketleri
- Hedef pazarlar bulma
  - Benzer özellikler gösteren müşterilerin bulunması:

    benzer gelir grupları, ilgi alanları, harcama alışkanlıkları
- Müşterilerin davranışlarında zaman içindeki değişiklik
  - Tek kişilik banka hesabının ortak hesaba çevrilmesi:Evlilik
- Çapraz pazar incelemesi:
  - Ürün satışları arasındaki bağlantıyı bulma

Pazar Araştırması - 2

- Müşteri profili
  - Hangi özellikteki müşterilerin hangi ürünleri aldıkları (demetleme veya sınıflandırma)
- Müşterilerin ihtiyaçlarını belirleme
  - Farklı müşterilerin o anki ilgisine yönelik ürünü bulma
  - Yeni müşterileri hangi faktörlerin etkilediğini bulma

Sahtekarlık İncelemesi

- Sigorta, bankacılık, telekomünikasyon alanlarında
- Geçmiş veri kullanılarak sahtekarlık yapanlar için bir model oluşturma ve benzer davranış gösterenleri belirleme

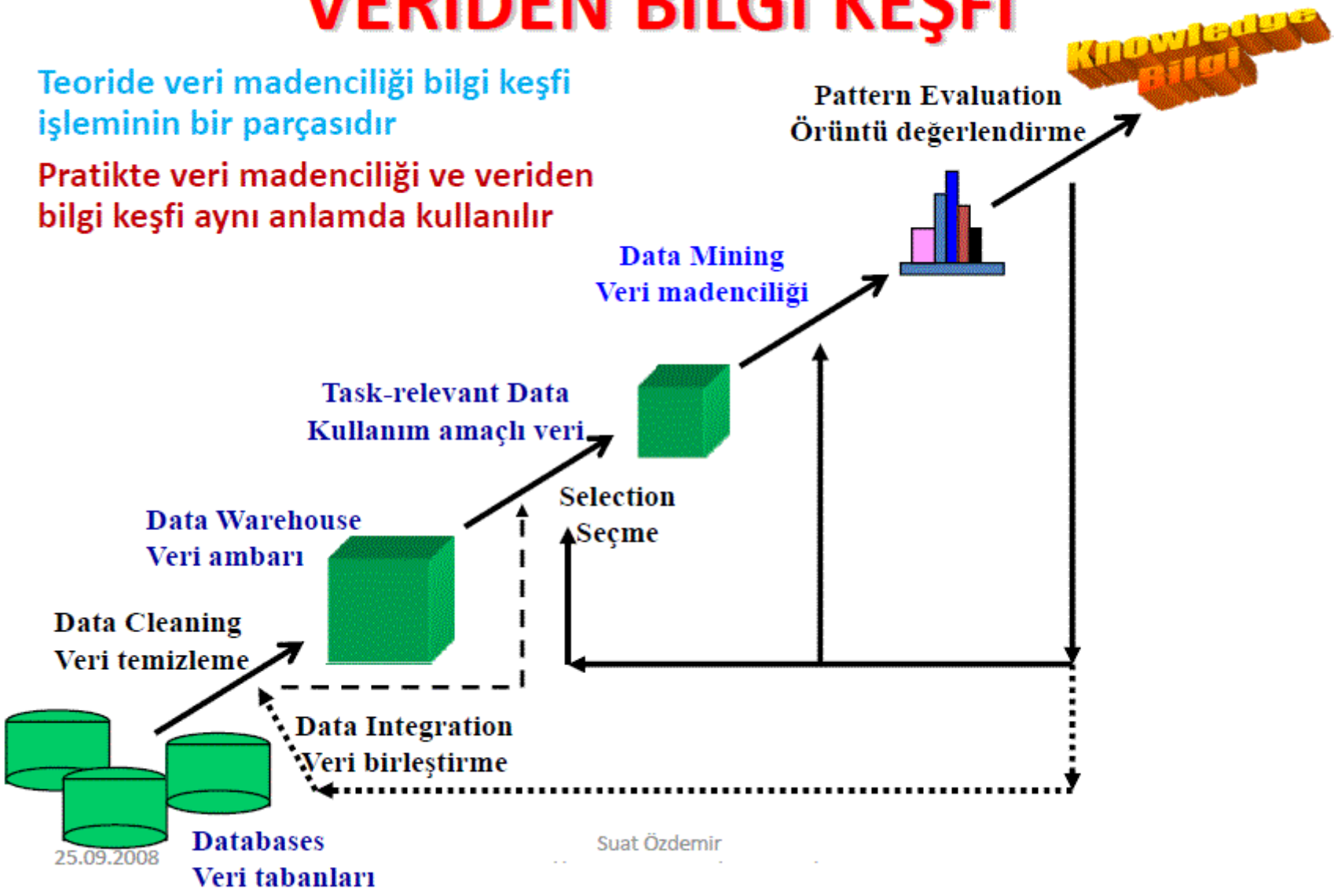
- Örnek:
  - Araba sigortası
  - Sağlık Sigortası
  - Kredi kartı başvurusu

Risk Analizi

- Finans planlaması ve bilanço değerlendirmesi
  - nakit para akışı incelemesi ve kestirimi
  - talep incelemesi
  - zaman serileri incelemesi
- Kaynak planlaması
  - kaynakların incelenmesi ve uygun olarak dağıtılması
- Rekabet
  - rakipleri ve pazar eğilimlerini takip etme
  - müşterileri sınıflara ayırma ve fiyat politikası belirleme

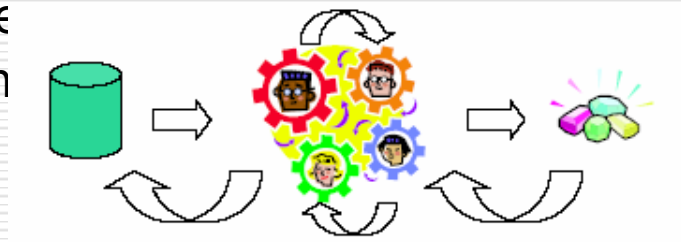
VERİDEN BİLGİ KEŞFİ

- Teoride veri madenciliği bilgi keşfi işleminin bir parçasıdır
- Pratikte veri madenciliği ve veriden bilgi keşfi aynı anlamda kullanılır



Bilgi Keşfinin Aşamaları

- Uygulama alanını inceleme
  - Konuyla ilgili bilgi ve uygulama amaçlarını belirleme
- Amaca uygun veri kümesi oluşturma: Veri seçme
- Veri ayıklama ve önişleme (işlemin %70'lik bölümünü oluşturur)
- Veri azaltma ve veri dönüşümü
  - incelemede gerekli boyutları (özellikleri) seçme, boyutlar arası ilişkiyi belirleme, boyut azaltma,
- Veri madenciliği tekniği seçme
  - Sınıflandırma, eğri uydurma, bağıntı kuralları, demetleme
- Veri madenciliği algoritmasını seçme
- Model değerlendirme ve bilgi sunum
- Bulunan bilginin yorumlanması



Bilgi Keşfi Örnek: Web Kayıtları

- web sitesinin yapısını inceleme
- verileri seçme: tarih aralığını belirleme
- veri ayıklama, önişleme: gereksiz kayıtları silme
- veri azaltma, veri dönüşümü: kullanıcı oturumları belirleme
- veri madenciliği tekniği seçme: demetleme
- veri madenciliği algoritması seçme: k-ortalama, EM,DBSCAN,…
- Model değerlendirme/yorumlama: değişik kullanıcı grupları için sıkça izlenen yolu bulma
- Uygulama alanları: öneri modelleri, kişiselleştirme, ön belleğe alma

Veri Kaynakları

- Veri dosyaları
- İlişkisel veritabanı
- Veri ambarları
- Gelişmiş veritabanları
  - nesneye dayalı veritabanları
  - WWW

Veri Madenciliği Algoritmaları

- amaç: veriyi belli bir modele uydurmak
  - tanımlayıcı
    - En iyi müşterilerim kimler?
    - Hangi ürünler birlikte satılıyor?
    - Hangi müşteri gruplarının alışveriş alışkanlıkları benzer?
  - kestirime dayalı
    - Kredi başvuralarını risk gruplarına ayırma
    - Şirketle çalışmayı bırakacak müşterileri öngörme
    - Borsa tahmini
- seçim: veriye uyan en iyi modeli seçmek için kullanılan kriter
- arama: veri üzerinde arama yapmak için kullanılan teknik

Veri Madenciliği Modelleri

- veri madenciliği
  - kestirime dayalı
    - sınıflandırma
    - Eğri uydurma
    - Zaman serileri
  - Tanımlayıcı
    - demetleme
    - özetleme
    - Bağıntı kuralları
    - Sıralı dizi

Veri Madenciliği Sınıflandırması

- Veri madenciliğinde veriyi belli bir modele uydurmak istiyoruz.
- İki ana grup
  - **Tanımlayıcı veri madenciliği**
    - Veriler arasındaki gizli kalmış ilişkiyi ortaya çıkarırlar
    - En iyi müşterilerim kimler?
    - Hangi ürünler birlikte satılıyor?
    - Hangi müşteri gruplarının alışveriş alışkanlıkları benzer?
  - **Kestirime dayalı veri madenciliği**
    - Kredi başvurularını risk gruplarına ayırma
      - Bu işlemde dolandırıcılık var mıdır?
    - Şirketle çalışmayı bırakacak müşterileri öngörme
    - Borsa Tahmini

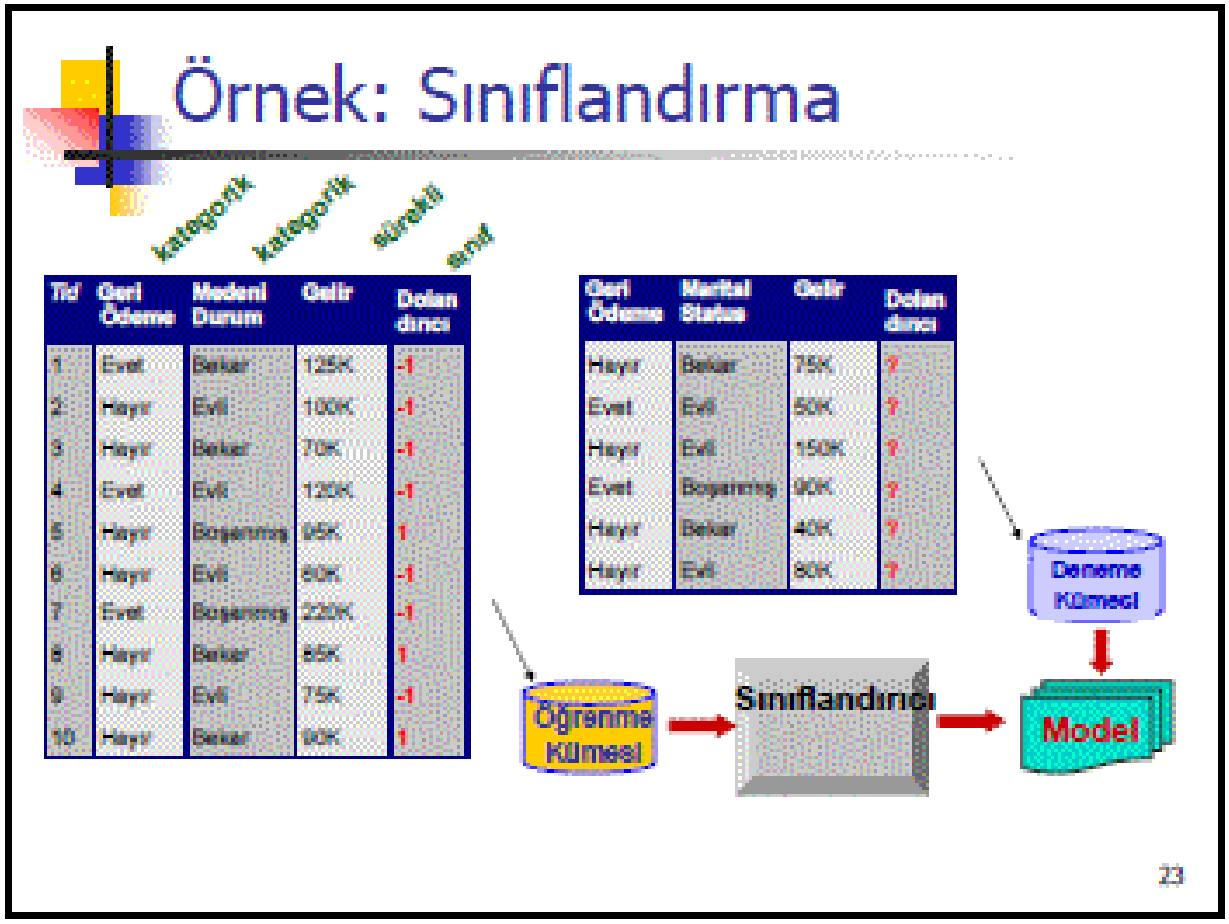
Veri Madenciliği İşlevleri (Kestirime Dayalı)

- Sınıflandırma: Veriyi önceden belirlenmiş sınıflardan birine dahil eder.
  - Gözetimli öğrenme
  - Örüntü tanıma
  - Kestirim
- Eğri uydurma: Veriyi gerçel değerli bir fonksiyona dönüştürür.
- Zaman serileri inceleme: Zaman içinde değişen verinin değerini öngörür

Veri Madenciliği İşlevleri (Tanımlayıcı)

- Demetleme: Benzer verileri aynı grupta toplama
  - Gözetimsiz öğrenme
- Özetleme: Veriyi alt gruplara ayırır. Her alt grubu temsil edecek özellikler bulur.
  - Genelleştirme
  - Nitelendirme
- Bağıntı kuralları
  - Veriler arasındaki ilişkiyi belirler
- Sıralı diziler: Veri içinde sıralı örüntüler bulmak için kullanılır

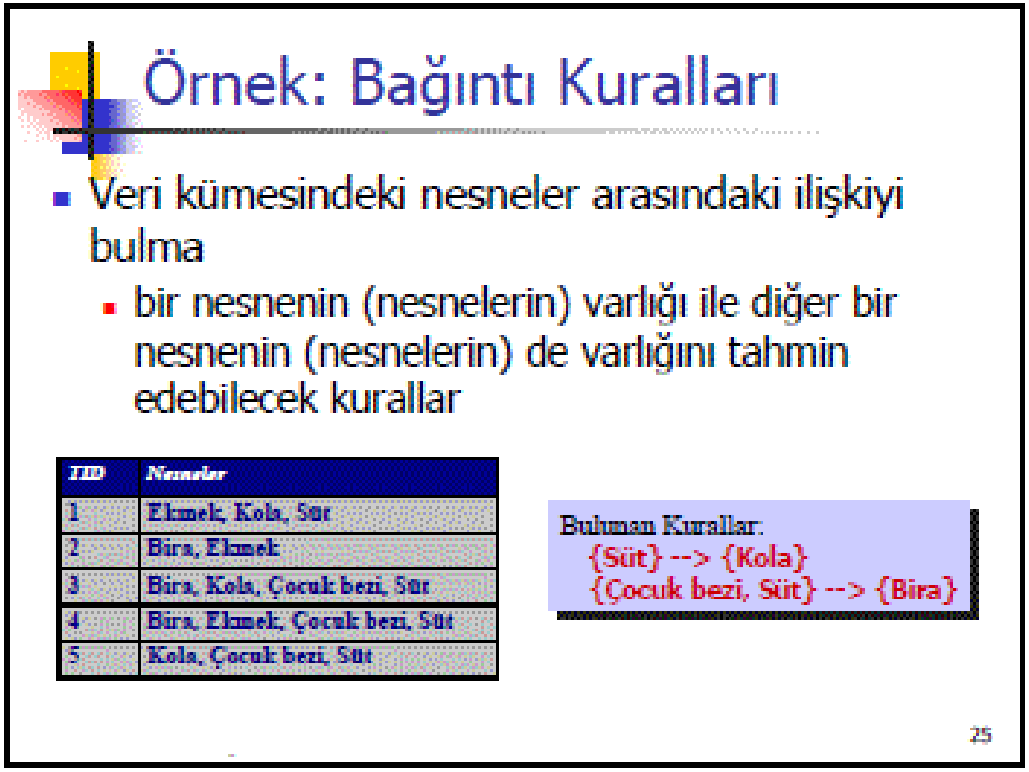
Örnek : Sınıflandırma



Örnek : Demetleme

- Doküman Demetleme:
- Amaç:
  - Döküman içinde geçen terimlere göre aynı konudaki dokümanları gruplama
- Yaklaşım:
  - Her doküman içinde sık geçen terimleri bul. Bu terimlerden ve ağırlıklarından yararlanarak bir benzerlik ölçütü geliştir. Bu ölçüte göre demetleme yap
- Kullanımı:
  - Yeni bir dokümanın hangi dokümanlarla benzer olduğunu bulma
  - terimlere göre arama yapıldığında bu terimleri içeren dokümanları bulma

Örnek : Bağıntı Kuralları



Bulunan Örtüntüler Dikkate Değer mi ?

- Hepsi dikkate alınması gerekmeyen binlerce örüntü
- Bulunan örüntünün dikkate alınması için:
  - insanlar tarafından kolayca anlaşılabilir
  - test verisi veya yeni veriler üzerinde belli oranda geçerli
  - yararlı ve kullanılabilir
  - yeni
- nesnel / öznel metrikler
  - nesnel: örüntünün yapısına bağlı
  - öznel: kullanıcının yaklaşımına bağlı

İlgili Konular : Veri Ambarları

- Çok fazla miktarda üzerinde işlem yapılan veri var
- Çoğunlukla farklı veritabanlarında ve farklı ortamlarda
- Veri farklı formatlarda ve yerlerde (heterojen ve dağıtık)
- Karar destek birimleri veriye sanal olarak tek bir yerden ulaşabilmeli
- Ulaşım hızlı olmalı

Veri Ambarı

- Amaca yönelik
- Birleştirilmiş
- Zaman değişkenli
- Değişken

Veri Ambarları : Amaca Yönelik

- Müşteri, ürün, satış gibi belli konular için düzenlenebilir
- Verinin incelenmesi ve modellenmesi için oluşturulur
- Konuyla ilgili karar vermek için gerekli olmayan veriyi kullanmayarak konuya basit, özet bakış sağlar

Veri Ambarları : Birleştirilmiş

- Veri kaynaklarının birleştirilmesiyle oluşturulur
  - Canlı veri tabanları, dosyalar
- Veri temizleme ve birleştirme teknikleri kullanılır
  - Değişik veri kaynakları arasındaki tutarlılık sağlanır

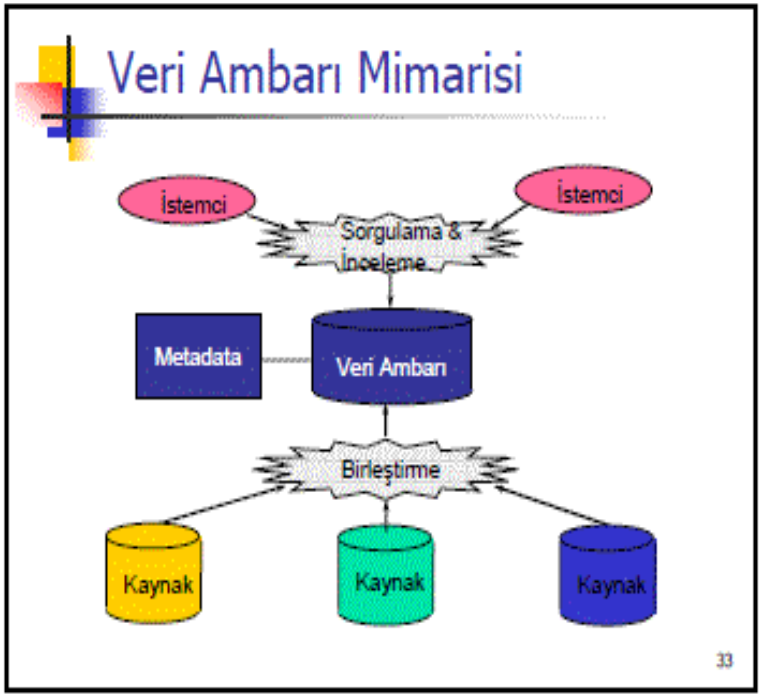
Veri Ambarları : Zaman Değişenli

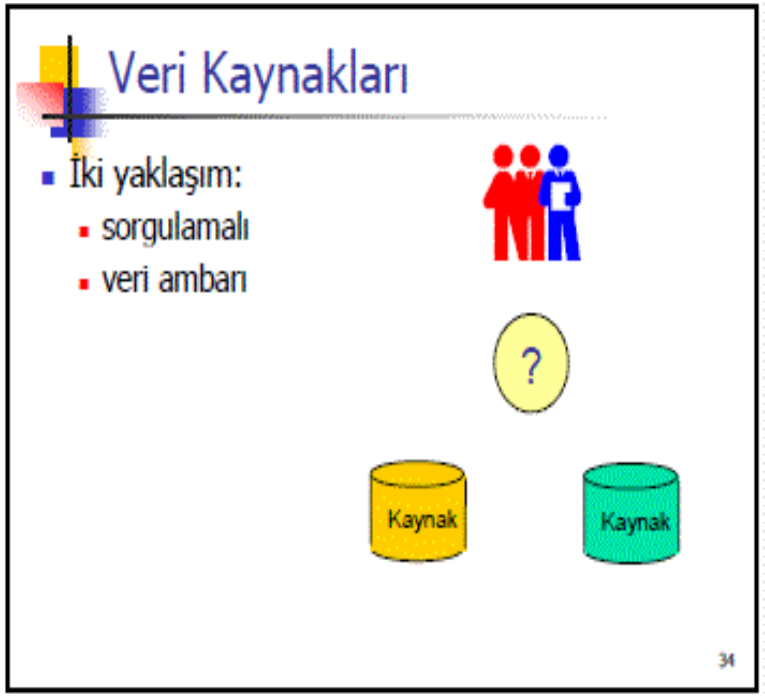
- Zaman değişkeni canlı veri tabanlarına göre daha uzundur
  - Canlı veri tabanları: Güncel veriler bulunur (en çok geçmiş 1 yıl)
  - Veri ambarları: Geçmiş hakkında bilgi verir (geçmiş 5-10 yıl)

Veri Ambarları : Değişen Değil

- Canlı veritabanlarından alınmış verinin fiziksel olarak başka bir ortamda saklanması
- Canlı veritabanlarındaki değişimin veri ambarlarını etkilememesi

Veri Ambarı

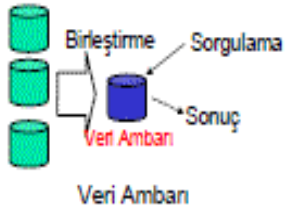

Veri Ambarı Mimarisi
İstemci
İstemci
Sorgulama & İnceleme
Metadata
Veri Ambarı
Birleştirme
Kaynak
Kaynak
Kaynak
33


Veri Kaynakları
İki yaklaşım:
sorgulamalı
veri ambarı
?
Kaynak
Kaynak
34

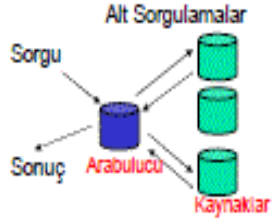
Veri Ambarları

Veri Ambarı & Birleşmiş Veritabanları

- Veritabanlarının birleştirilmesi:
  - Farklı veritabanları arasında bir arabulucu katman
  - Sorgulamalı
    - Bir sorgulamayı her veritabanı için alt sorgulamalara ayır
    - Sonucu birleştir
- Veri ambarı:
  - Veri daha sonra kullanılmak üzere birleştirilip veri ambarında saklanıyor



Veri Ambarı



Birleştirilmiş veritabanları

35

Veri Madenciliği & OLAP

- OLAP (On-Line Analytical Processing)
  - Veri ambarlarının işlevi
  - Veriyi inceleme ve karar verme
  - OLTP (On-Line Transaction Processing) saatler sürebilen işlemler
- OLAP avantajları
  - Daha geniş kapsamlı sonuçlar
  - Daha kısa süreli işlem
- OLAP dezavantajları
  - Kullanıcı neyi nasıl soracağını bilmesi gerekiyor
  - Genelde veriden istatistiksel inceleme yapmak için kullanılır.

OLAP NE sorusuna cevap veriri veri madenciliği NEDEN sorusuna cevap verir.

36

Veri Madenciliğinde Sorunlar(1)

- İnsan etkileşimi
- Aşırı öğrenme
- Aykırılıklar
- Yorumlama
- Görüntüleme
- Büyük veri kümeleri
- Çok boyut

Veri Madenciliğinde Sorunlar(2)

- Farklı veri tipleri
- Eksik veri
- İlgisiz veri
- Gürültülü veri
- Değişen veri
- Birleştirme
- Uygulama

Sosyal Sorunlar

- Gizlilik
- İzlenme
- İzinsiz kullanma

Konular

- Veri
- Veri Önişleme
- Benzerlik ve farklılık

Veri Nedir?

- nesneler ve nesnelerin niteliklerinden oluşan küme
  - kayıt (record), varlık (entity), örnek (sample, instance) nesne için kullanılabilir.
- Nitelik (attribute) bir nesnenin (object) bir özelliğidir
  - bir insanın yaşı, ortamın sıcaklığı..
  - boyut (dimension), özellik (feature, characteristic) olarak da kullanılır
- Niteliklerve bu niteliklere ait değerler bir nesneyi oluşturur.

nitelikler

nesneler

Tid	Geri Ödeme	Medeni Durum	Gelir	Dolandırıcı
1	Evet	Bekar	125K	-1
2	Hayır	Evli	100K	-1
3	Hayır	Bekar	70K	-1
4	Evet	Evli	120K	-1
5	Hayır	Boşanmış	95K	1
6	Hayır	Evli	60K	-1
7	Evet	Boşanmış	220K	-1
8	Hayır	Bekar	85K	1
9	Hayır	Evli	75K	-1
10	Hayır	Bekar	90K	1

Değer Kümeleri

- Nitelik için saptanmış sayılar veya semboller
- Nitelik & Değer Kümeleri
  - aynı nitelik farklı değer kümelerinden değer alabilir
    - ağırlık: kg, lb
  - farklı nitelikler aynı değer kümesinden değer alabilirler
    - ID, yaş: her ikisi de sayısal

Nitelik Türleri

- Belli aralıkta yeralan değişkenler (interval)
  - sıcaklık, tarih
- İkili değişkenler (binary)
  - cinsiyet
- Ayrık ve sıralı değişkenler (nominal, ordinal, ratio scaled)
  - göz rengi, posta kodu
- Karma tipte değişkenler

Konular

- Veri
- Veri Önişleme
- Benzerlik ve farklılık

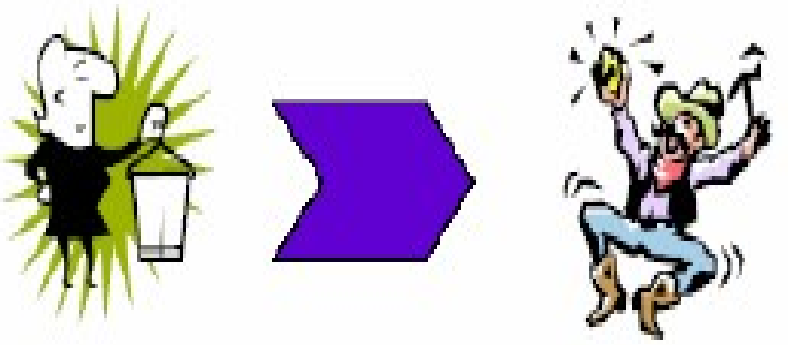
Problem

- Uygulamalarda toplanan veri yetersiz, tutarsız ya da gürültülü olabilir.
  - Hata sebepleri:
    - hatalı veri toplama gereçleri
    - veri giriş problemleri
    - veri girişi sırasında kullanıcıların hatalı yorumları
    - veri iletim hataları
    - teknolojik sınırlamalar
    - veri isimlendirmede uyumsuzluk
  - Sonuçları:
    - tekrarlanan kayıtlar
    - çelişkili veriler
    - yetersiz veriler

Problem

- Veri güvenilmez
  - Veri madenciliği sonuçlarına güvenilebilir mi?

- Veri kaliteli ise veri madenciliği uygulamaları ile yararlı bilgi bulma şansı daha fazla.



Veri Önişleme

- Veri temizleme
- Veri birleştirme
- Veri dönüşümü
- Veri azaltma

Konular

- Veri
- Veri Önişleme
  - Veri temizleme
    - eksik veri
    - gürültülü veri
  - Veri birleştirme
  - Veri dönüşümü
  - Veri azaltma
- Benzerlik ve farklılık

Veri Temizleme

- Gerçek uygulamalarda veri eksik, gürültülü veya tutarsız olabilir.
  - Bazı nitelik değerleri girilmemiş
  - Bazı nitelik değerleri hatalı
  - Bazı nitelik değerlerinin kaydedilmesine gerek görülmemiş
- Veri temizleme işlemleri
  - Eksik nitelik değerlerini tamamlama
  - Aykırılıkların bulunması ve gürültülü verinin düzeltilmesi
  - Tutarsızlıkların giderilmesi

Eksik Veri

- Veri için bazı niteliklerin değerleri her zaman bilinemeyebilir.
- Eksik veri
  - diğer veri kayıtlarıyla tutarsızlığı nedeniyle silinmesi
  - yanlış anlama sonucu kaydedilmeme
  - veri girişi sırasında bazı nitelikleri önemsiz görme

Eksik Veriler nasıl Tamamlanır?

- Eksik nitelik değerleri olan veri kayıtlarını kullanma
- Eksik nitelik değerlerini doldur
- Eksik nitelik değerleri için global bir değişken kullan (Null, bilinmiyor,...)
- Eksik nitelik değerlerini o niteliğin ortalama değeri ile doldur
- Aynı sınıfa ait kayıtların nitelik değerlerinin ortalaması ile doldur
- Olasılığı en fazla olan nitelik değeriyle doldur

Gürültülü Veri

- Ölçülen bir değerdeki hata
- Yanlış nitelik değerleri
  - hatalı veri toplama gereçleri
  - veri girişi problemleri
  - veri iletimi problemleri
  - teknolojik kısıtlar
  - nitelik isimlerinde tutarsızlık

Gürültülü Veri nasıl Düzeltilir?

- **Gürültüyü yok etme**
  - **Bölmeleme**
    - önce veriyi sırala, sonra eşit aralıklarla bölmele
    - her bölme ortalamayla ya da bölmenin en alt ve üst sınırlarıyla temsil edilir
  - **Demetleme**
    - aykırılıkları belirler
  - **Eğri uydurma**
    - veriyi eğriye uydurarak gürültüyü düzeltir

Bölmeleme

- Veri sıralanır: 4, 8, 15, 21, 21, 24, 25, 28, 34

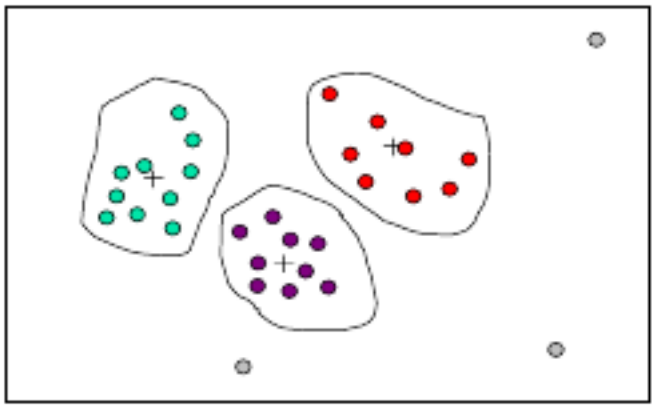
Bölme genişliği:3
1. Bölme: 4, 8, 15
2. Bölme: 21, 21, 24
3. Bölme: 25, 28, 34

Ortalamayla düzeltme:
1. Bölme: 9, 9, 9
2. Bölme: 22, 22, 22
3. Bölme: 29, 29, 29

Alt-üst sınırla düzeltme:
1. Bölme: 4, 4, 15
2. Bölme: 21, 21, 24
3. Bölme: 25, 25, 34

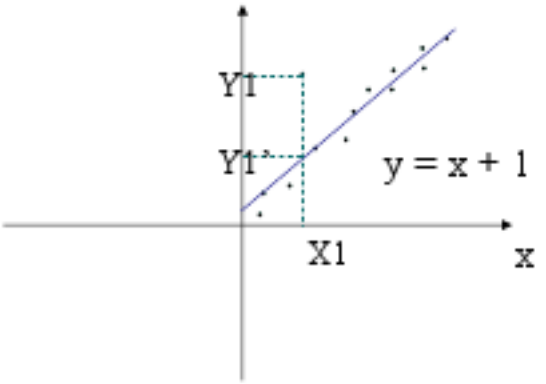
Demetleme

- Benzer veriler aynı demette olacak şekilde gruplanır
- Bu demetlerin dışında kalan veriler aykırılık olarak belirlenir ve silinir



Eğri Uydurma

- Veri bir fonksiyona uydurulur. Doğrusal eğri uydurmada, bir değişkenin değeri diğer bir değişken kullanılarak bulunabilir.



Konular

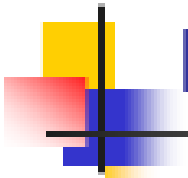
- Veri
- Veri Önişleme
  - Veri temizleme
  - Veri birleştirme
  - Veri dönüşümü
  - Veri azaltma
- Benzerlik ve farklılık

Veri Birleştirme

- Farklı kaynaklardan verilerin tutarlı olarak birleştirilmesi
- Şema birleştirilmesi
  - Aynı varlıkların saptanması: A.cust_id=B.cust_num
  - meta veri kullanılır
- Nitelik değerlerinin tutarsızlığının saptanması
  - Aynı nitelik için farklı kaynaklarda farklı değerler olması
  - Farklı metrikler kullanılması

Gereksiz Veri

- Farklı veri kaynaklarından veriler birleştirilince gereksiz (fazla) veri oluşabilir
  - aynı nitelik farklı kaynaklarda farklı isimle
  - bir niteliğin değeri başka bir nitelik kullanılarak hesaplanabilir
    - korelasyon hesaplaması



Konular

- Veri
- Veri Önişleme
  - Veri temizleme
  - Veri birleştirme
  - Veri dönüşümü
    - normalizasyon
    - nitelik oluşturma
  - Veri azaltma
- Benzerlik ve farklılık

Veri Dönüşümü

- Veri, veri madenciliği uygulamaları için uygun olmayabilir
  - Seçilen algoritmaya uygun olmayabilir
  - Veri belirleyici değil
- Çözüm
  - normalizasyon
  - nitelik oluşturma

Normalizasyon

- min-max normalizasyon

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

- z-score normalizasyon

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

- ondalık normalizasyon

$$v' = \frac{v}{10^j}$$ j: Max(| v' |)<1 olacak şekildeki en küçük tam sayı

Nitelik Oluşturma

- Yeni nitelikler yarat
  - orjinal niteliklerden daha önemli bilgi içersin
  - veri madenciliği algoritmalarının başarımı daha iyi olsun

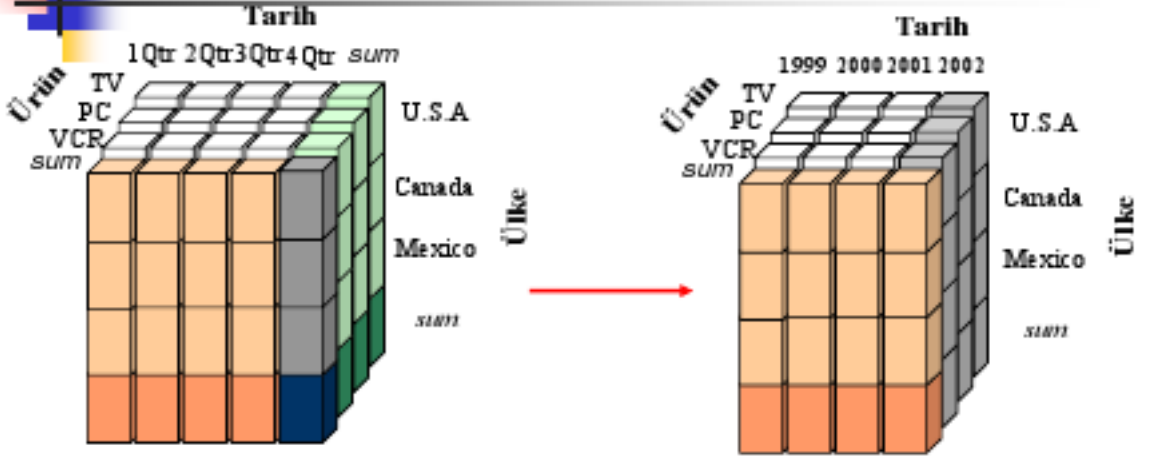
Konular

- Veri
- Veri Önişleme
  - Veri temizleme
  - Veri birleştirme
  - Veri dönüşümü
  - Veri azaltma
    - nitelik birleştirme
    - nitelik azaltma
    - veri sıkıştırma
    - veri ayrıştırma ve kavram oluşturma
    - değer azaltma
- Benzerlik ve farklılık

Veri Azaltma

- Veri miktarı çok fazla olduğu zaman veri madenciliği algoritmalarının çalışması ve sonuç üretmesi çok uzun sürebilir
  - veriyi azaltma başarımı artırır
  - sonucun (nerdeyse) hiç değişmemesi gerekir
- Veri azaltma
  - nitelik birleştirme
  - nitelik azaltma
  - veri sıkıştırma
  - veri ayrıştırma ve kavram oluşturma
  - veri küçültme
    - eğri uydurma
    - demetleme
    - histogram
    - örnekleme

Nitelik Birleştirme



- Sorgulama için gerekli olan boyutlar kullanılıyor.

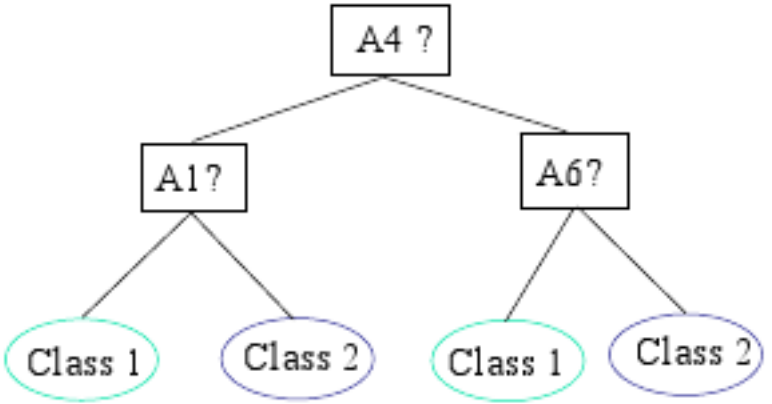
Nitelik Azaltma

- Nitelik seçme
  - Veri madenciliği uygulaması için gerekli olan nitelikleri seç
  - Nitelikler altkümesi kullanılarak elde edilen sınıfların dağılımları gerçek dağılıma eşit ya da çok yakın olmalı
- Sezgisel yöntemler kullanılarak nitelikler azaltılabilir.
  - istatistiksel anlamlılık testi (statistical significance)
  - bilgi kazancı (information gain)
  - karar ağaçları

Örnek

Başlangıç nitelikler kümesi:

{A1, A2, A3, A4, A5, A6}



Seçilen nitelik kümesi: {A1, A4, A6}

Veri Sıkıştırma

- Verinin boyutunu azaltır
  - daha az saklama ortamı
  - veriye ulaşmak daha çabuk
- Kayıplı ve kayıpsız veri sıkıştırma
  - bazı yöntemler bazı veri tiplerine uygun
  - her veri tipi için kullanılan yöntemler de var
- Eğer veri madenciliği yöntemi sıkıştırılmış veri üzerinde doğrudan çalışabiliyorsa elverişli

Veri Ayrıştırma

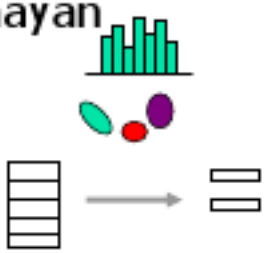
- Bazı veri madenciliği algoritmaları sadece ayrık veriler ile çalışır
- Sürekli bir nitelik değerini bölerek her aralığı etiketler
- Verinin değeri, bulunduğu aralağın etiketi ile değişir
- Veri boyutu küçülür
- Kavram oluşturmak için kullanılır

Kavram Oluşturma

- Sayısal veriler
  - çok geniş aralıkta olabilir
  - değerleri çok sık değişebilir
- Sayısal veriler için kavram oluşturma
  - bölmeleme
  - histogram
  - demetleme
  - entropi
  - kesmeleme

Veri Küçültme

- Veriyi farklı şekillerde gösterme
  - parametrik
    - eğri uydurma
  - parametrik olmayan
    - histogram
    - demetleme
    - örnekleme



Histogram ile Veri Küçültme

- Verinin dağılımı
- Veriyi bölerek her bölüm için veri değerini gösterir (toplam, ortalama)
  - eşit genişlik (equi-width): bölmelerin genişliği eşit
  - eşit yükseklik (equi-height): her bölmedeki veri sayısı eşit
  - v-optimal: en az varyansı olan histogram $\Sigma(count_b * value_b)$
  - MaxDiff: bölme genişliğini kullanıcı belirler

Demetleme ile Veri Küçültme

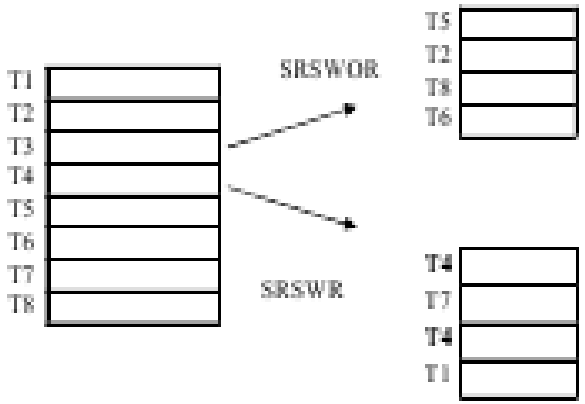
- Veri demetlere ayrılır
- Veri yerine demetler ve aykırılıklar temsil edilir
- Etkisi verinin dağılımına bağlı
- Hiyerarşik demetleme yöntemleri kullanılabilir.

Örnekleme ile Veri Küçültme

- Büyük veri kümesini daha küçük bir alt küme ile temsil etme
- Alt küme nasıl seçiliyor?
  - yerine koymadan örnekleme (SRSWOR)
  - yerine koyarak örnekleme (SRSWR)
  - demet örnekleme (yerine koymadan veya koyarak)
  - katman örnekleme (katman: nitelik değerine göre grup)

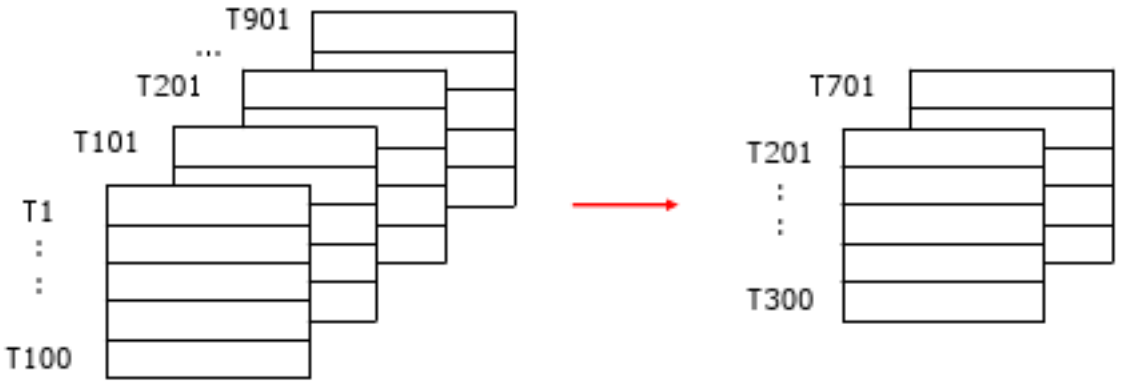
Örnek

- Örnekleme



Örnek

- Demetleme



Örnek

- Katman Örnekleme

T1	genç
T2	genç
T3	genç
T4	genç
T5	orta yaşlı
T6	orta yaşlı
T7	orta yaşlı
T8	orta yaşlı
T9	orta yaşlı
T10	orta yaşlı
T11	orta yaşlı
T12	orta yaşlı
T13	yaşlı
T14	yaşlı

T1	genç
T4	genç
T6	orta yaşlı
T7	orta yaşlı
T9	orta yaşlı
T11	orta yaşlı
T13	yaşlı

Konular

- Veri
- Veri Önişleme
- Benzerlik ve farklılık

Benzerlik ve Farklılık

- Benzerlik
  - iki nesnenin benzerliğini ölçen sayısal değer
  - nesneler birbirine daha benzer ise daha büyük
  - genelde 0-1 aralığında değer alır
- Farklılık
  - iki nesnenin birbirinden ne kadar farklı olduğunu gösteren sayısal değer
  - nesneler birbirine daha benzer ise daha küçük
  - en küçük farklılık genelde 0
  - üst sınır değişebilir

Öklid Uzaklığı

- Veri kümesi

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix}$$

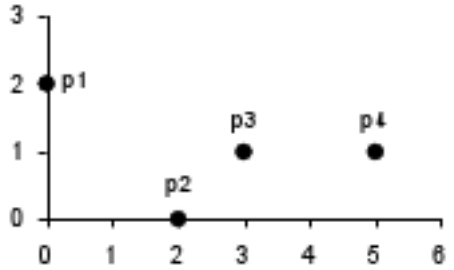
- Uzaklık matrisi

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

- Öklid uzaklığı (Euclidean Distance) nesneler arası farklılığı bulmak için kullanılır.
  - p adet niteliği (boyutu) olan i ve j nesneleri arasındaki uzaklık

$$d(i, j) = \sqrt{\left|x_{i1} - x_{j1}\right|^2 + \left|x_{i2} - x_{j2}\right|^2 + \dots + \left|x_{ip} - x_{jp}\right|^2}$$

Örnek: Öklid Uzaklığı



nesne	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

Uzaklık Matrisi

Minkowski Uzaklığı

- Öklid uzaklığının genelleştirilmiş hali

$$d(i,j) = \sqrt[q]{(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \ldots + |x_{ip} - x_{jp}|^q)}$$

q: positif tam sayı

- q=1 → Manhattan uzaklığı

Uzaklık Özellikleri

- q=1⇒ Manhattan Uzaklığı
- q=2 ⇒ Euclidean Uzaklığı
- Özellikleri:
  1. d(i,j)≥0
  2. d(i,i)=0
  3. d(i,j)=d(j,i)
  4. d(i,j)≤d(i,h)+d(h,j)
- Bu özellikleri sağlayan uzaklık ölçüttür
- Uzaklıklar ağırlıklı olarak da hesaplanabilir:

$$d(i,j) = \sqrt{w_1|x_{i1} - x_{j1}|^2 + w_2|x_{i2} - x_{j2}|^2 + \ldots + w_p|x_{ip} - x_{jp}|^2}$$

Benzerlik Özellikleri

- İki nesne arası benzerlik özellikleri
  1. sim(i,j)<=1
  2. sim(i,i)=1
  3. sim(i,j)=sim(j,i)

İkili Değişkenler Arası Benzerlik

- İkili bir değişkenin 0 veya 1 olarak iki değeri olabilir.
- Bir olasılık tablosu oluşturulur:

		Nesne j	
		0	1
Nesne i	0	M_{00}	M_{01}
	1	M_{10}	M_{11}

M_{00}: i nesnesinin 0, j nesnesinin 0 olduğu niteliklerin sayısı
M_{10}: i nesnesinin 1, j nesnesinin 0 olduğu niteliklerin sayısı
M_{01}: i nesnesinin 0, j nesnesinin 1 olduğu niteliklerin sayısı
M_{11}: i nesnesinin 1, j nesnesinin 1 olduğu niteliklerin sayısı

- Yalın uyum katsayısı (simple matching coefficient): ikili değişkenin simetrik olduğu durumlarda

$$sim(i, j) = \frac{M_{11} + M_{00}}{M_{00} + M_{01} + M_{10} + M_{11}}$$

- Jaccard katsayısı (İkili değişkenin asimetrik olduğu durumlar):

$$d(i, j) = \frac{M_{11}}{M_{01} + M_{10} + M_{11}}$$

Örnek

p = 1 0 0 0 0 0 0 0 0 0
q = 0 0 0 0 0 0 1 0 0 1

$M_{01} = 2$
$M_{10} = 1$
$M_{00} = 7$
$M_{11} = 0$

Yalın Uyum Katsayısı:

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

Jaccard Katsayısı:

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

Örnek

Ad	Cinsiyet	Ateş	Öksürük	Test-1	Test-2	Test-3	Test-4
Ali	Erkek	E	H	P	N	N	N
Ayşe	Bayan	E	H	P	N	P	N
Mehmet	Erkek	E	E	N	N	N	N
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

- Cinsiyet simetrik ikili bir değişken.
- Diğer değişkenler asimetrik ikili değişkenler.
- E (Var), P (pozitif) değerleri 1'e, N (yok veya negatif) değerleri 0'a eşitlenirse Jaccard Katsayısı şu şekilde hesaplanabilir:

$$d(Ali, Ayse) = \frac{0+1}{2+0+1} = 0{,}33$$

$$d(Ali, Mehmet) = \frac{1+1}{1+1+1} = 0{,}67$$

$$d(Mehmet, Ayse) = \frac{1+2}{1+1+2} = 0{,}75$$

Cosine Benzerliği

- d_1 ve d_2 iki doküman. Cosine benzerliği

  $\cos(d_1,d_2)=d_1 \bullet d_2 / ||d_1|| \, ||d_2||$

  $d_i \bullet d_j$: iki dokümanın vektör çarpımı

  $||d_i||$: d_i dokümanının uzunluğu

- **Örnek**

  $d_1 = 3\,2\,0\,5\,0\,0\,0\,2\,0\,0$

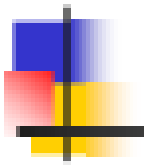
  $d_2 = 1\,0\,0\,0\,0\,0\,0\,1\,0\,2$

  $d_1 \bullet d_2 = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$

  $||d_1|| = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.481$

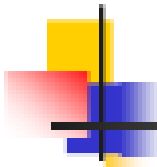
  $||d_2|| = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)^{0.5} = (6)^{0.5} = 2.245$

  $\cos(d_1, d_2) = .3150$



VERİ MADENCİLİĞİ

Temel Sınıflandırma Yöntemleri

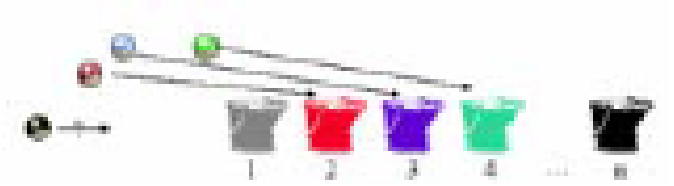


Sınıflandırma

- Sınıflandırma (classification) problemi:
  - nesnelerden olışan veri seti (öğrenme kümesi): $D=\{t_1,t_2,\ldots,t_n\}$
  - her nesne niteliklereden oluşuyor, bir nitelik sınıf bilgisi
- Sınıf niteliğini belirlemek için diğer nitelikleri kullanarak bir model bulma
- Öğrenme kümesinde yer almayan nesneleri (sınama kümesi) mümkün olan en iyi şekilde doğru sınıflara atamak
- sınıflandırma=ayrık değişkenler için öngörüde (prediction) bulunma

Sınıflandırma

- Amaç: Bir niteliğin değerini diğer nitelikleri kullanarak belirlemek
  - verinin dağılımına göre bir model bulunur
  - bulunan model, başarımı belirlendikten sonra niteliğin gelecekteki ya da bilinmeyen değerini tahmin etmek için kullanılır
    - model başarımı: doğru sınıflandırılmış sınama kümesi örneklerinin oranı
- Veri madenciliği uygulamasında:
  - ayrık nitelik değerlerini tahmin etmek: sınıflandırma
  - sürekli nitelik değerlerini tahmin etmek: öngörü



- Sınıflandırma: hangi topun hangi sepete koyulabileceği
- Öngörü: Topun ağırlığı

Sınıflandırma İşlemi

- **Sınıflandırma İşlemi:**
- **Model Oluşturma**
  - Her nesnenin sınıf etiketi olarak tanımlanan niteliğinin belirlediği bir sınıfta olduğu varsayılır
  - Model oluşturmak için kullanılan nesnelerin oluşturduğu veri kümesi **öğrenme kümesi** olarak tanımlanır
- Model farklı biçimlerde ifade edilebilir
  - IF – THEN – ELSE kuralları ile
  - Karar ağaçları ile
  - Matematiksel formüller ile

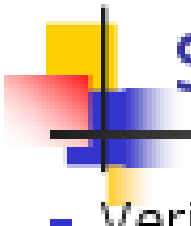
Gözetimli & Gözetimsiz Sınıflandırma

- Gözetimli (Supervised) sınıflandırma = sınıflandırma
  - Sınıfların sayısı ve hangi nesnenin hangi sınıfta olduğu biliniyor.

- Gözetimsiz (Unsupervised) sınıflandırma = clustering
  - Hangi nesnenin hangi sınıfta olduğu bilinmiyor. Genelde sınıf sayısı bilinmiyor.

Sınıflandırma Uygulamaları

- Kredi başvurusu değerlendirme
- Kredi kartı harcamasının sahtekarlık olup olmadığına karar verme
- Hastalık teşhisi
- Ses tanıma
- Karakter tanıma
- Gazete haberlerini konularına göre ayırma
- Kullanıcı davranışları belirleme



Sınıflandırma için Veri Hazırlama

- Veri dönüşümü:
  - Sürekli nitelik değeri ayrık hale getirilir
  - Normalizasyon ([-1,…,1], [0,…,1])
- Veri temizleme:
  - gürültüyü azaltma
  - gereksiz nitelikleri silme

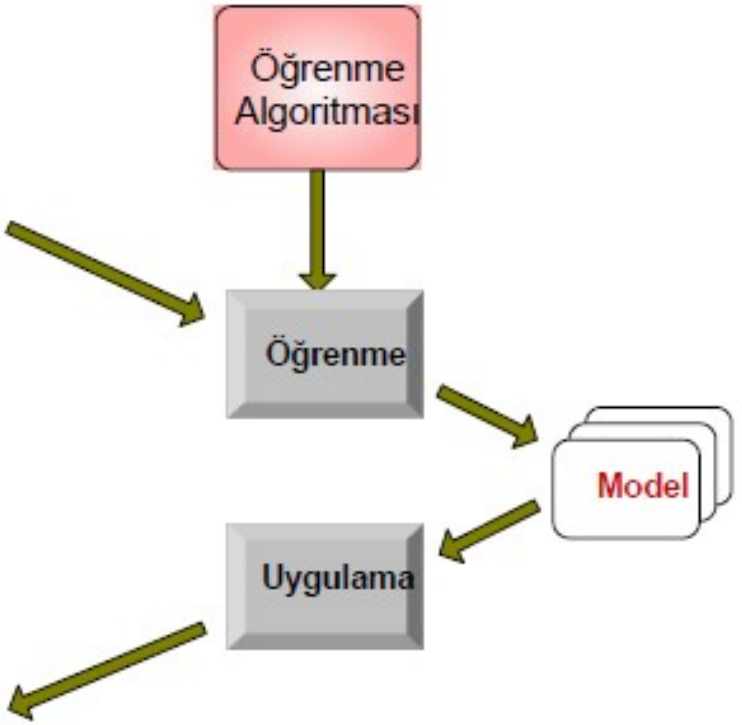
Örnek

Tid	Nit1	Nit2	Nit3	Sınıf
1	1	Büyük	125K	0
2	0	Orta	100K	0
3	0	Küçük	70K	0
4	1	Orta	120K	0
5	0	Büyük	95K	1
6	0	Orta	60K	0
7	1	Büyük	220K	0
8	0	Küçük	85K	1
9	0	Orta	75K	0
10	0	Küçük	90K	1

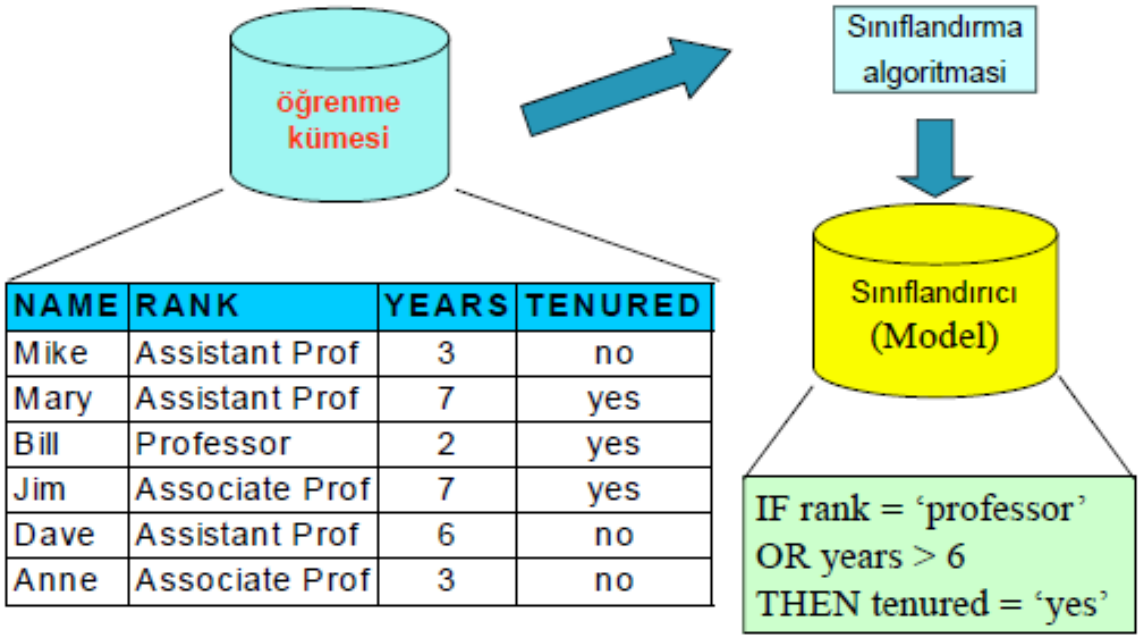
Öğrenme Kümesi

Tid	Nit1	Nit2	Nit3	Sınıf
11	0	Küçük	55K	?
12	1	Orta	80K	?
13	1	Büyük	110K	?
14	0	Küçük	95K	?
15	0	Büyük	67K	?

Sınama Kümesi



Model Oluşturma



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

Sınıflandırma İşlemi

Model Değerlendirme:

- Modelin başarımı (doğruluğu) sınama kümesi örnekleri kullanılarak belirlenir
- Sınıf etiketi bilinen bir sınama kümesi örneği model kullanılarak belirlenen sınıf etiketiyle karşılaştırılır
- Modelin doğruluğu, doğru sınıflandırılmış sınama kümesi örneklerinin toplam sınama kümesi örneklerine oranı olarak belirlenir

- Sınama kümesi model öğrenirken kullanılmaz

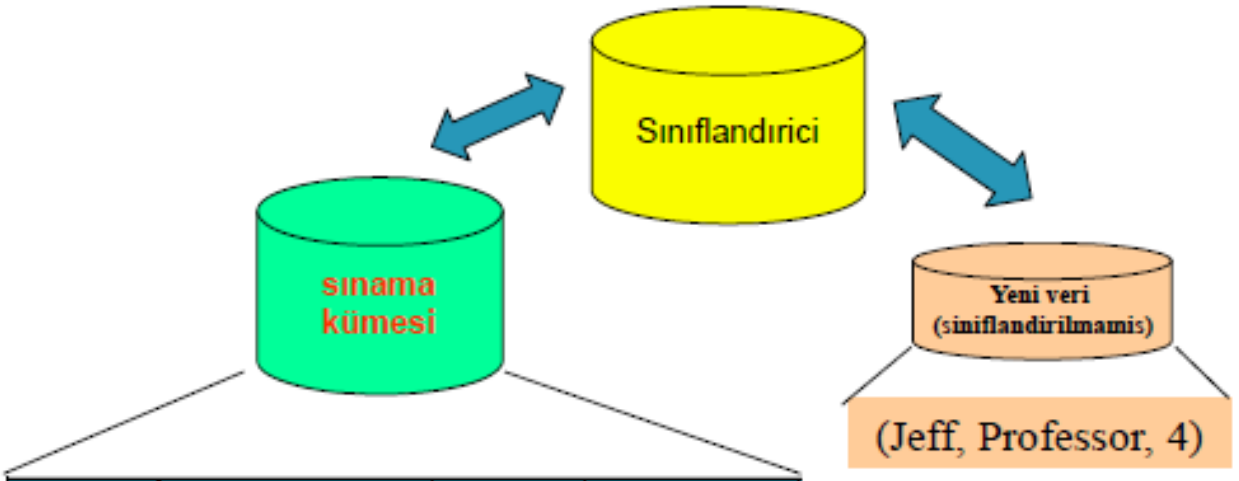
Sınıflandırma İşlemi:

Modeli kullanma:

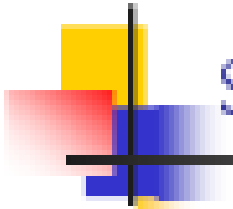
– Model daha önce görülmemiş örnekleri sınıflandırmak için kullanılır

- Örneklerin sınıf etiketlerini tahmin etme
- Bir niteliğin değerini tahmin etme

Modeli kullanma



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



Sınıflandırıcı Başarımını Değerlendirme

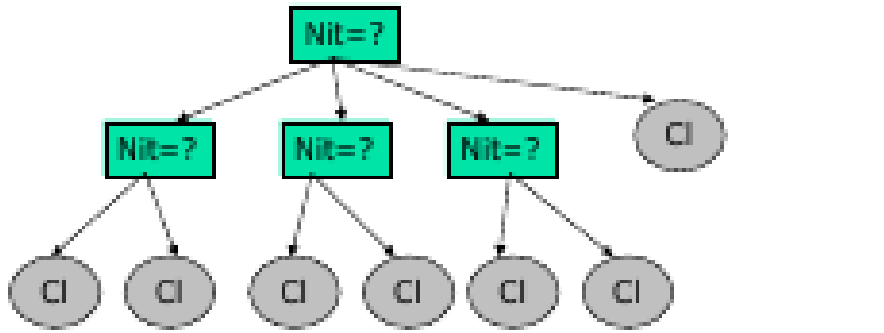
- Doğru sınıflandırma başarısı
- Hız
  - modeli oluşturmak için gerekli süre
  - sınıflandırma yapmak için gerekli süre
- Kararlı olması
  - gürülltülü ve eksik veri olduğu durumlarda da iyi sonuç vermesi
- Ölçeklenebilirlik
  - büyük miktarda veri ile çalışabilmesi
- Anlaşılabilir olması
  - kullanıcı tarafından yorumlanabilir olması
- Kuralların yapısı
  - birbiriyle örtüşmeyen kurallar

Sınıflandırma Yöntemleri

- Karar ağaçları (decision trees)
- Yapay sinir ağları (artificial neural networks)
- Bayes sınıflandırıcılar (Bayes classifier)
- Bağıntı tabanlı sınıflandırıcılar (association-based classifier)
- k-en yakın komşu yöntemi (k- nearest neighbor method)
- Destek vektör makineleri (support vector machines)
- Genetik algoritmalar (genetic algorithms)
- ...

Karar Ağaçları

- Akış diyagramı şeklinde ağaç yapısı
  - Ara düğümler bir nitelik sınaması
  - Dallar sınama sonucu
  - Yapraklar sınıflar



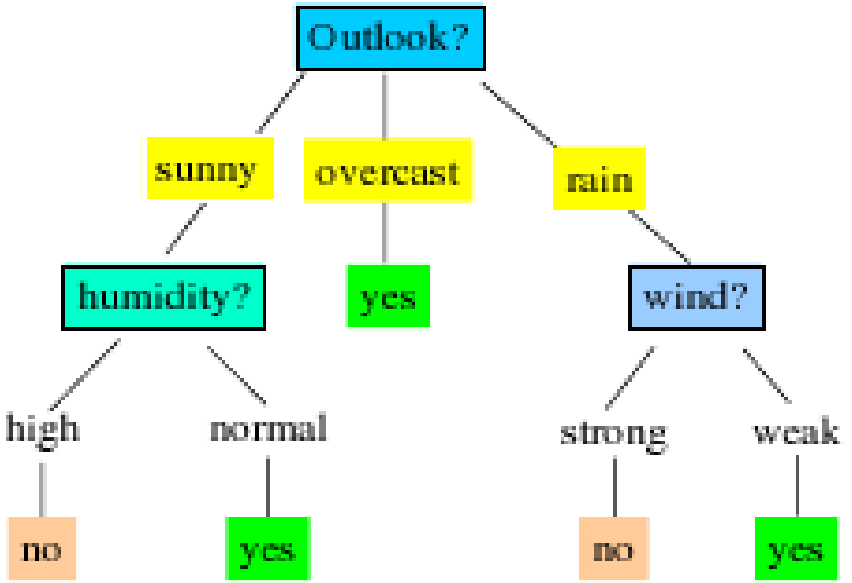
Örnek: Karar Ağacı

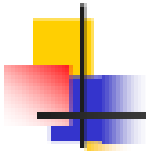
- J. Ross Quinlan'ın geliştirdiği ID3 modeline uyarlanmış:
  - hava tenis oynamaya uygun mu?

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

Hava durumu Verisi

Örnek: Karar Ağacı





Karar Ağacı Yöntemleri

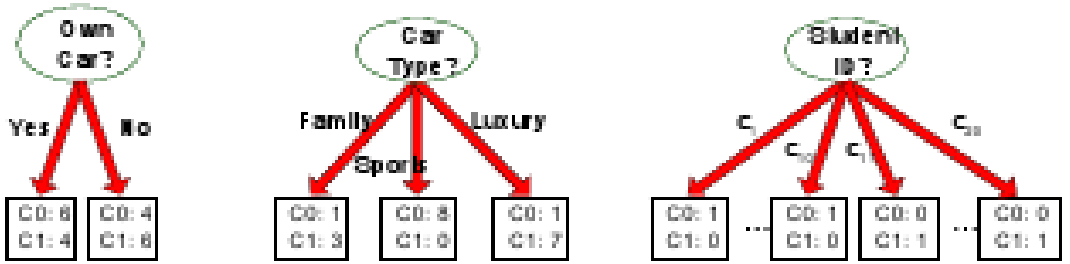
- Karar ağacı oluşturma yöntemleri genel olarak iki aşamadan oluşur:
  - ağaç oluşturma
    - en başta bütün deneme kümesi örnekleri kökte
    - seçilen niteliklere bağlı olarak örnek yinelemeli olarak bölünüyor
  - ağaç budama
    - deneme kümesindeki gürültülü verilerden oluşan ve sınama kümesinde hataya neden olan dalları silme (sınıflandırma başarımını artırır)

Karar Ağacı Oluşturma

- Yinelemeli işlem
  - ağaç bütün verinin oluşturduğu tek bir düğümle başlıyor
  - eğer örnekleri hepsi aynı sınıfa aitse düğüm yaprak olarak sonlanıyor ve sınıf etiketini alıyor
  - eğer değilse örnekleri sınıflara en iyi bölecek olan nitelik seçiliyor
  - işlem sona eriyor
    - örneklerin hepsi (çoğunluğu) aynı sınıfa ait
    - örnekleri bölecek nitelik kalmamış
    - kalan niteliklerin değerini taşıyan örnek yok

Örnekleri En İyi Bölen Nitelik Hangisi?

- Bölmeden önce:
  - 10 örnek C0 sınıfında
  - 10 örnek C1 sınıfında



Hangisi daha iyi?

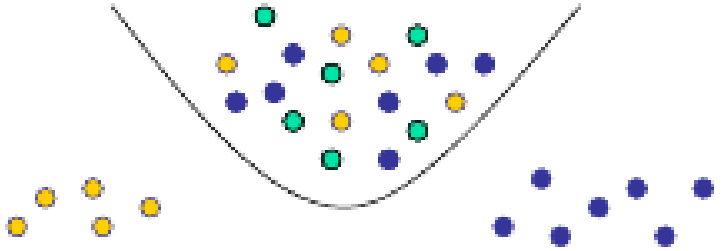
En iyi Bölme Nasıl Belirlenir?

- "Greedy" yaklaşım
  - çoğunlukla aynı sınıfa ait örneklerin bulunduğu (homojen) düğümler tercih edilir
- Düğümün kalitesini ölçmek için bir yöntem

En İyi Bölen Nitelik Nasıl Belirlenir?

- İyilik Fonksiyonu (Goodness Function)
- Farklı algoritmalar farklı iyilik fonksiyonları kullanabilir:
  - bilgi kazancı (information gain): ID3, C4.5
    - bütün niteliklerin ayrık değerler aldığı varsayılıyor
    - sürekli değişkenlere uygulamak için değişiklik yapılabilir
  - gini index (IBM IntelligentMiner)
    - bütün niteliklerin sürekli değişken değerleri aldığı varsayılıyor
    - her nitelik için farklı bölme değerleri olduğu varsayılıyor
    - bölme değerlerini belirlemek için başka yöntemlere (demetleme gibi) ihtiyaç var
    - ayrık değişkenlere uygulamak için değişiklik yapılabilir

Bilgi Kazancı



sepetteki toplar farklı renklerde ise az bilgi var

topların hepsi aynı renkte ise daha fazla bilgi var

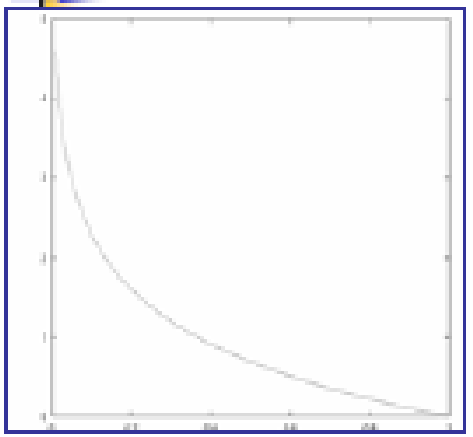
Bilgi / Entropi

- $p_1, p_2, .., p_s$ toplamları 1 olan olasılıklar. Entropi (Entropy)

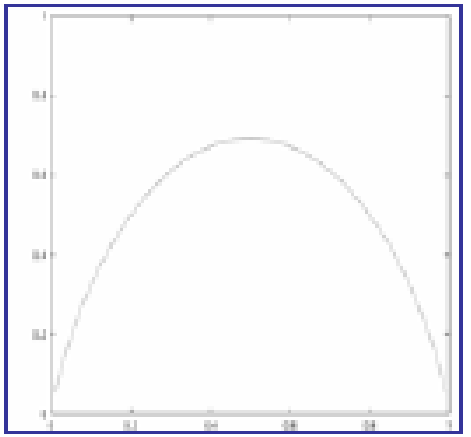
$$H(p_1, p_2, ..., p_s) = -\sum_{i=1}^{s} p_i \log(p_i)$$

- Entropi rastgeleliği, belirsizliği ve beklenmeyen durumun ortaya çıkma olasılığını verir
- Sınıflandırmada
  - olayın olması beklenen bir durum
  - entropi=0

Entropi



log (p)



H(p,1-p)

- örnekler aynı sınıfa aitse entropi=0
- örnekler sınıflar arasında eşit dağılmışsa entropi=1
- örnekler sınıflar arasında rastgele dağılmışsa 0<entropi<1

Örnek

- S veri kümesinde 14 örnek: C0 sınıfına ait 9, C1 sınıfına ait 5 örnek.
- Entropi

$$H(p_1, p_2, \ldots, p_s) = -\sum_{i=1}^{s} p_i \log(p_i)$$

- $H(p_1,p_2) = -(9/14)\,\mathrm{Log}_2(9/14) - (5/14)\,\mathrm{Log}_2(5/14)$
  $= 0.940$

Bilgi Kazancı (ID3 / C4.5)

- Bilgi kuramı kavramlarını kullanarak karar ağacı oluşturulur. Sınıflandırma sonucu için en az sayıda karşılaştırma yapmayı hedefler.
- Ağaç bir niteliğe göre dallandığında entropi ne kadar düşer?
- *A* niteliğinin *S* veri kümesindeki bilgi kazancı

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

Values(A), *A* niteliğinin alabileceği değerler, S_v, $A=v$ olduğu durumda S'nin altkümesi.

Örnek

- Bilgi kazancına göre nitelik seçme

toplam örnek sayısı s=14, iki sınıfa ayrılmış

s1=9(yes), s2=5(no)

Entropy(S) = - (9/14) Log_2 (9/14) - (5/14) Log_2 (5/14) = 0.940

wind için: weak=8, strong=6

weak: no=2, yes=6

strong: no=3, yes=3

Entropy(S_{weak}) = - (6/8)*log2(6/8) - (2/8)*log2(2/8) = 0.811

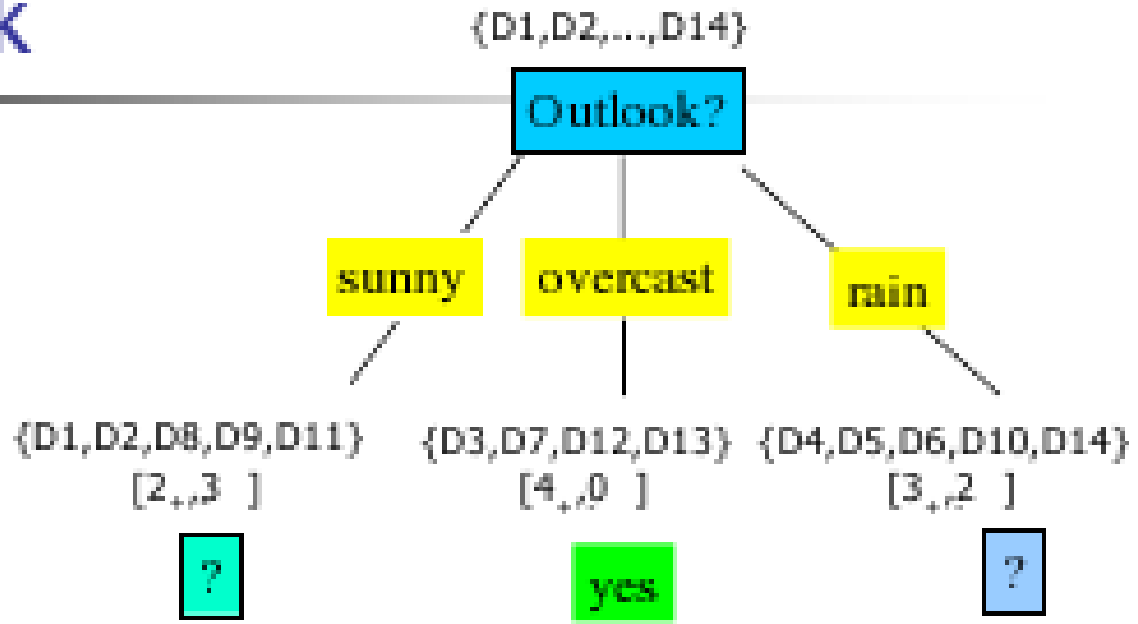
Entropy(S_{strong}) = - (3/6)*log2(3/6) - (3/6)*log2(3/6) = 1.00

Gain(wind)=0.940 - (8/14)*0.811 - (6/14)*1.00

Gain(Outlook) = 0.246
Gain(Humidity) = 0.151
Gain(wind)=0.048
Gain(Temperature) = 0.029

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

Örnek



S_{sunny} = {D1,D2,D8,D9,D11}, Entropy(S_{sunny})=0.970
humidity için: high=3, normal=2
high: no=3, yes=0
normal: no=0, yes=2
Entropy(S_{high}) = 0
Entropy(S_{normal}) = 0
Gain(S_{sunny},Humidity)=0.970-(3/5)0.0- (3/5)0.0=0.970

Gini Index (IBM InteligentMiner)

- Veri kümesi S içinde n sınıf varsa ve p_j C_j sınıfının olasılığı ise

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

- Eğer veri kümesi S_1 ve S_2 altkümelere bölünüyorsa ve her altkümede sırasıyla N_1 ve N_2 örnek varsa:

$$gini_{split}(S) = \frac{N_1}{N} gini(S_1) + \frac{N_2}{N} gini(S_2)$$

- Gini Index değeri en küçük olan nitelik seçilir.

Örnek

$$GINI(S) = 1 - \sum_j [p_j]^2$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Gini = 1 – P(C1)² – P(C2)² = 1 – 0 – 1 = 0

C1	**1**
C2	**5**

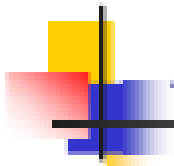
P(C1) = 1/6 P(C2) = 5/6

Gini = 1 – (1/6)² – (5/6)² = 0.278

C1	**2**
C2	**4**

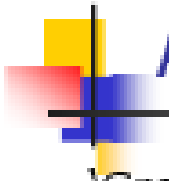
P(C1) = 2/6 P(C2) = 4/6

Gini = 1 – (2/6)² – (4/6)² = 0.444



Ağaç Oluşturmada Temel Yaklaşımlar

- Bölme kriteri:
  - ağacın bir düğümünde karşılaştırma yapılacak niteliğin seçilmesi
  - farklı algoritmalar farklı iyilik fonksiyonları kullanabilir: bilgi kazancı, gini index,...
- Dallanma kriteri:
  - bir örneğin hangi dala ait olduğunu belirleme
  - ikiye dallanma (gini index), çoklu dallanma (bilgi kazancı)
- Durma kararı:
  - dallanma işleminin devam edip etmeyeceğine karar verme
- Etiketleme kuralı:
  - yaprak düğüm en çok örneği olan sınıfla etiketleniyor



Ağaç Oluşturma:

- 'Greedy' algoritma
  - her adımda en iyi çözümü bul: her düğümde dallanmak için en iyi niteliği bul
- Parçala ve çöz (divide and conquer)
  - kökten yapraklara
  - düğümü dallara ayır
  - her dal için algoritmayı uygula

Örnek Algoritma: ID3

- Bütün nitelikler ayrık
- Bir düğüm oluştur N:
  - Eğer örneklerin hepsi C sınıfına ait ise, N düğümü C etiketli yaprak
  - Eğer karşılaştırma yapılacak nitelik yoksa N düğümü en çok örneği olan sınıf
- En büyük bilgi kazancı olan niteliği bölmek için seç
  - N'yi seçilen nitelik ile etiketle
  - niteliğin her A_i değeri için bir dal oluştur
  - S_i, örneklerin hepsinin A_i değeri aldığı dal
    - S_i boş → bir yaprak oluşturup en çok örneği olan sınıfla etiketle
    - Si boş değil → algoritmayı S_i düğümü üzerinde yinele
- Yaprak düğümlere kadar

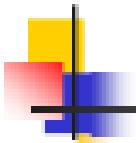
Karar Ağacı Kullanarak Sınıflandırma

- Doğrudan
  - sınıflandırmak istenilen örneğin nitelikleri ağaç boyunca sınanır
  - ulaşılan yaprağın etiketi sınıf bilgisini verir
- Dolaylı
  - karar ağacı sınıflandırma kurallarına dönüştürülür
  - kökten yaprakların herbirine giden yollar için ayrı bir kural oluşturulur.
  - IF-THEN şeklinde kuralları insanlar daha kolay anlıyor
    - Örnek: IF Outlook="sunny" AND humidity="normal" THEN play tennis

Karar Ağacı Kullanarak Sınıflandırma – Değerlendirme

Avantajları:

- Dezavantajları:

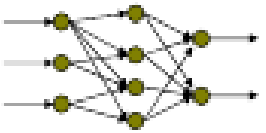


Karar Ağacı Boyutunu Belirleme

- Veri seti öğrenme ve sınama kümesi olarak ayrılır
- Çapraz geçerleme kullanılır.
- Veri kümesinin tümü ağacı oluşturmak için kullanılır
  - istatistiksel bir test ile (chi-square) düğüm eklemenin ya da ağacı küçültmenin katkısı sınanır

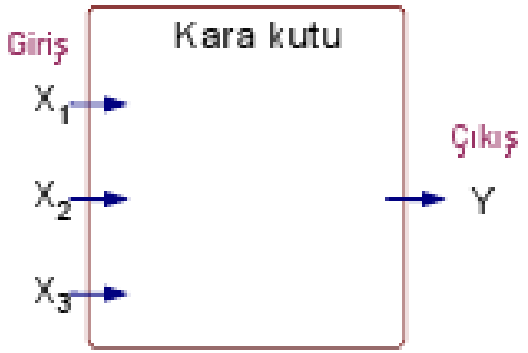
Yapay Sinir Ağları ile Sınıflandırma

- İnsan beynindeki sinir hücrelerinin işlevini modelleyen bir yapı
- Birbiri ile bağlantılı katmanlardan oluşur.
  - katmanlar hücrelerden oluşur
- Katmanlar arasında iletim
- İleti katmanlar arasındaki bağın ağırlığına ve her hücrenin değerine bağlı olarak değişebilir



Örnek:

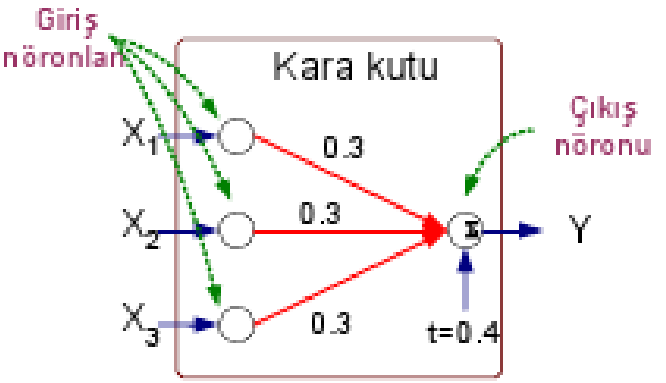
X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0



- En az iki giriş 1 ise çıkış 1, diğer durumlarda çıkış 0

Örnek

X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0

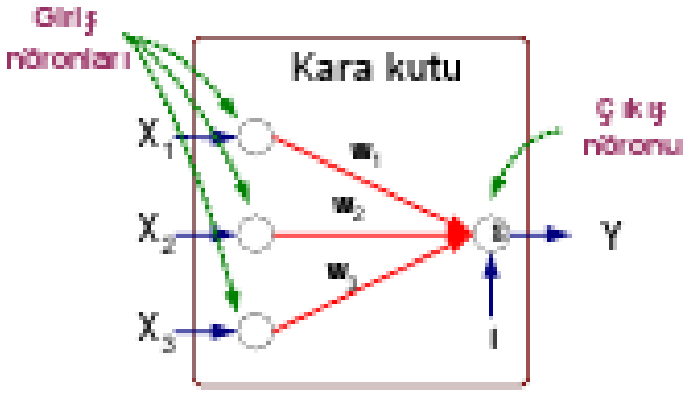


$$Y = I(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4 > 0)$$

$$I(z) = \begin{cases} 1 \text{ eğer } z>0 \\ 0 \text{ diğer} \end{cases}$$

Yapay Sinir Ağları

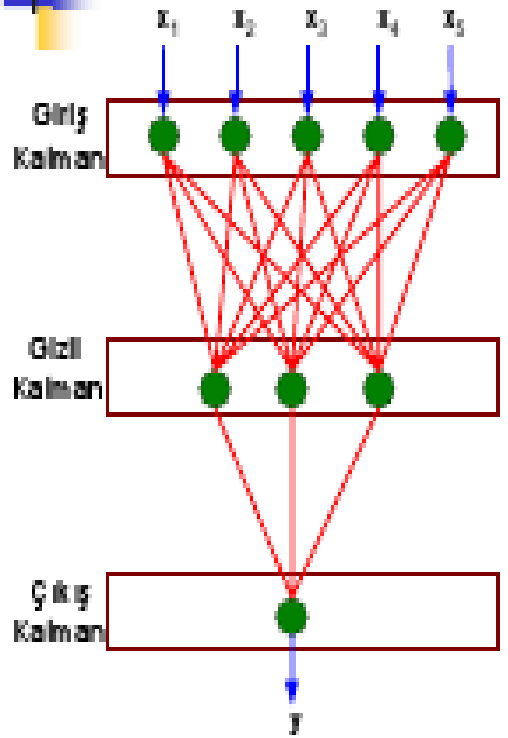
- Birbiri ile bağlantılı nöronlar ve ağırlıklar
- Çıkış nöronu kendisine gelen girişleri ağırlıklı olarak topluyor
- Çıkış nöronu bir eşik değeri ile karşılaştırılıyor

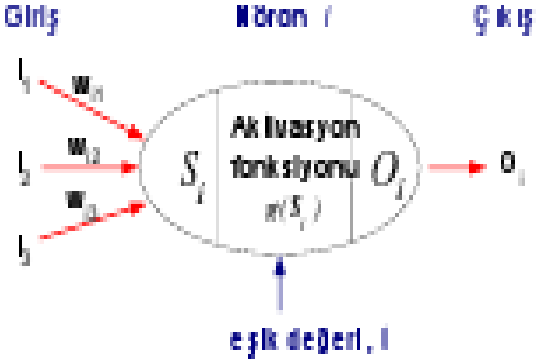


$$Y = I(\sum_i w_i X_i - t)$$

$$Y = sign(\sum_i w_i X_i - t)$$

Çok Katmanlı

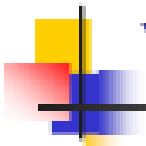




- Yapay sinir ağını öğrenme: ağırlıkları öğrenme

Yapay Sinir Ağı ile Öğrenme

- Yapay sinir ağı oluşturma
  - giriş verisini modelleme
  - gizli katman sayısını, gizli katmanlardaki nöron sayısını belirleme
- Yapay sinir ağını eğitme
- Sinir ağını küçültme
- Sonucu yorumlama



Yapay Sinir Ağını Oluşturma

- Giriş nöron sayısı
  - Öğrenme kimesindeki verilerin nitelik sayısı
- Gizli nöron sayısı
  - öğrenme sırasında ayarlanır
- Çıkış nöron sayısı
  - sınıf sayısı

Yapay Sinir Ağını Eğitme

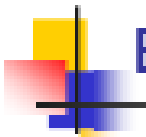
- Amaç: Veri kümesindeki örneklerin hepsini doğru sınıflandıracak ağırlıkları belirlemek
  - ağırlıklara rastgele değerler ata
  - öğrenme kümesindeki giriş değerlerini teker teker sinir ağına uygula
  - çıkışı hesapla
  - hata değerini hesapla $E = \sum_i [Y_i - f(w_i, X_i)]^2$
  - ağırlıkları hata fonksiyonunu enküçültecek şekilde düzelt

Yapay Sinir Ağını Küçültme

- Tam bağlı ağın anlaşılması çok güç
- n giriş nöron, h gizli nöron, m çıkış nöronu
  h(m+n) ağırlık
- Küçültme: ağırlıklardan bazıları sınıflandırma sonucunu etilemeyecek şekilde silinir

Yapay Sinir Ağları

- Yararları
  - doğru sınıflandırma oranı genelde yüksek
  - sağlam – öğrenme kümesinde hata olduğu durumda da çalışıyor
  - çıkış ayrık, sürekli ya da ayrık veya sürekli değişkenlerden oluşan bir vektör olabilir
- Olumsuz yönleri
  - öğrenme süresi uzun
  - öğrenilen fonksiyonun anlaşılması zor



Bayes (İstatistiksel) Modelleme

- Bayes teoremini kullanan istatistiksel sınıflandırıcı
- Örneklerin hangi sınıfa hangi olasılıkla ait oldukları
- Naïve Bayes sınıflandırıcı
  - niteliklerin hepsi aynı derecede önemli
  - nitelikler birbirinden bağımsız
    - bir niteliğin değeri başka bir nitelik değeri hakkında bilgi içermiyor
  - sınıflandırma ve öğrenme problemleri

Bayes Teoremi

- X sınıflandırılacak örnek. Hipotez h, X örneğinin C sınıfına ait olduğu
- h hipotezinin sonrasal olasılığı (*posteriori probability*)

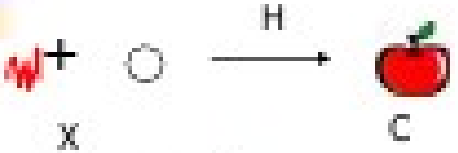
$$P(h|X) = \frac{P(X|h)P(h)}{P(X)}$$

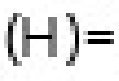
- MAP (maximum posteriori) hipotez

$$h_{MAP} \equiv \arg\max_{h \in H} P(h|D) = \arg\max_{h \in H} P(D|h)P(h)$$

- Çok sayıda olasılığı önceden kestirmek gerekiyor

Örnek

- 

- P(H)=P() P(X)=P()
- P(X|H)= P() eğer )

$$P(h|X) = \frac{P(X|h)P(h)}{P(X)}$$

Naïve Bayes Sınıflandırıcı

- $X=(x_1,x_2,\ldots x_n)$ örneğinin C sınıfında olma olasılığı ($P(C|X)$) nedir?
- $\frac{P(X|C_i)P(C_i)}{P(X)}$ değerini enbüyütme

  → $P(X|C_i)P(C_i)$ değerini enbüyütme

- $P(C_i)=|S_i| / |S|$, S_i: C_i sınıfına ait örneklerin sayısı
- $P(X|C_i)=\prod_{k=1}^{n} P(x_k|C_i)$, $P(x_k|C_i)=s_{ik} / s_k$
- Hesaplama maliyetini azaltıyor, sadece sınıf dağılımları hesaplanıyor
- Naïve: nitelikler bağımsız

Hava Durumu Verisi için Olasılıklar

Outlook			Temperature			Humidity			Windy			Play	
	Yes	*No*		*Yes*	*No*		*Yes*	*No*		*Yes*	*No*	*Yes*	*No*
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes

Hava Durumu Verisi için Olasılıklar

Outlook			Temperature			Humidity			Windy			Play	
	Yes	*No*		*Yes*	*No*		*Yes*	*No*		*Yes*	*No*	*Yes*	*No*
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

- Yeni veri

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

İki Sınıf için olasılık:

P("yes"|X) = 2/9 × 3/9 × 3/9 × 3/9 × 9/14 = 0.0053

P("no"|X) = 3/5 × 1/5 × 4/5 × 3/5 × 5/14 = 0.0206

Normalize edilmiş olasılıklar:

P("yes") = 0.0053 / (0.0053 + 0.0206) = 0.205

P("no") = 0.0206 / (0.0053 + 0.0206) = 0.795

Sürekli Veriler için Olasılık

- Verinin normal dağılımdan geldiği varsayılıyor.
  - Her sınıf-nitelik çifti için bir olasılık hesaplanıyor.

$$P(A_i \mid c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(A_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

  - Gelir için sınıf=-1
    - ortalama=110
    - varyans=2975

$$P(Gelir = 120 \mid -1) = \frac{1}{\sqrt{2\pi}(54.54)} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

Tid	Geri Ödeme	Medeni Durum	Gelir	Dolandırıcı
1	Evet	Bekar	125K	-1
2	Hayır	Evli	100K	-1
3	Hayır	Bekar	70K	-1
4	Evet	Evli	120K	-1
5	Hayır	Boşanmış	95K	1
6	Hayır	Evli	60K	-1
7	Evet	Boşanmış	220K	-1
8	Hayır	Bekar	85K	1
9	Hayır	Evli	75K	-1
10	Hayır	Bekar	90K	1

Örnek

X=(Geri Ödeme="Hayır",Medeni Durum="Evli",Gelir=120k)

P(Geri Ödeme="Evet"|-1)=3/7
P(Geri Ödeme="Hayır"|-1)=4/7
P(Geri Ödeme="Evet"|1)=0
P(Geri Ödeme="Hayır"|1)=1
P(Medeni Durum="Evli"|-1)=4/7
P(Medeni Durum="Bekar"|-1)=2/7
P(Medeni Durum="Boşanmış"|-1)=1/7
P(Medeni Durum="Evli"|1)=0
P(Medeni Durum="Bekar"|1)=2/3
P(Medeni Durum="Boşanmış"|1)=1/3

Gelir:
- Sınıf=-1
  - ortalama=110
  - varyans=2975
- Sınıf=1
  - ortalama=90
  - varyans=25

- P(X|Sınıf=-1) =
  P(Geri Ödeme="Hayır"|Sınıf=-1)
  × P(Medeni Durum="Evli"|Sınıf=-1)
  × P(Gelir=120K| Sınıf=-1)
  = 4/7 × 4/7 × 0.0072 = 0.0024

- P(X|Sınıf=1) =
  P(Geri Ödeme="Hayır"|Sınıf=1)
  × P(Medeni Durum="Evli"| Sınıf=1)
  × P(Gelir=120K| Sınıf=1)
  = 1 × 0 × 1.2 × 10^{-9} = 0

P(X|-1)P(-1) > P(X|1)P(1)
P(-1|X) > P(1|X)
=> Sınıf = -1

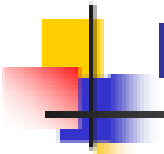
Olasılığın Sıfır Olması

- Her sınıfta bir niteliğin her değeri olmazsa
  - koşullu olasılıklardan biri 0
  - o sınıfa ait olma olasılığı 0
- Olasılıklar

$$\text{Original}: P(A_i \mid C) = \frac{N_{ic}}{N_c}$$

$$\text{Laplace}: P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c}$$

c: sınıf sayısı
Toplamları 1 olmak zorunda

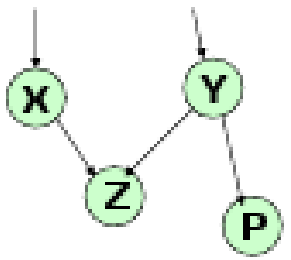


Bayes Sınıflandırıcılar

- Avantajları:
  - gerçeklemesi kolay
  - çoğu durumda iyi sonuçlar
- Dezavantajları
  - varsayım: sınıf bilgisi verildiğinde nitelikler bağımsız
  - gerçek hayatta değişkenler birbirine bağımlı
  - değişkenler arası ilişki modellenemiyor
- Çözüm:
  - Bayes ağları

Bayes Ağları

- Niteliklerin altkümesinin birbiri ile bağımsız olduğunu varsayıyor
- Yönlü çevrimsiz çizge (directed acyclic graph) ve koşullu olasılık tablolarından oluşur
- Her değişken A için bir tablo var
  - niteliğin ebeveynlerine olan koşullu olasılıkları



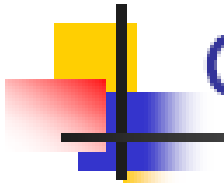
- düğümler: rastgele değişkenler
- ayrıtlar: olasılıklı bağlılık
- *X* ve *Y* , *Z* değişkeninin ebeveyni
- *Y*, *P* değişkeninin ebeveyni
- *Z* ve *P* arasında bağ yok

VERİ MADENCİLİĞİ

Farklı Sınıflandırma Yöntemleri

Konular

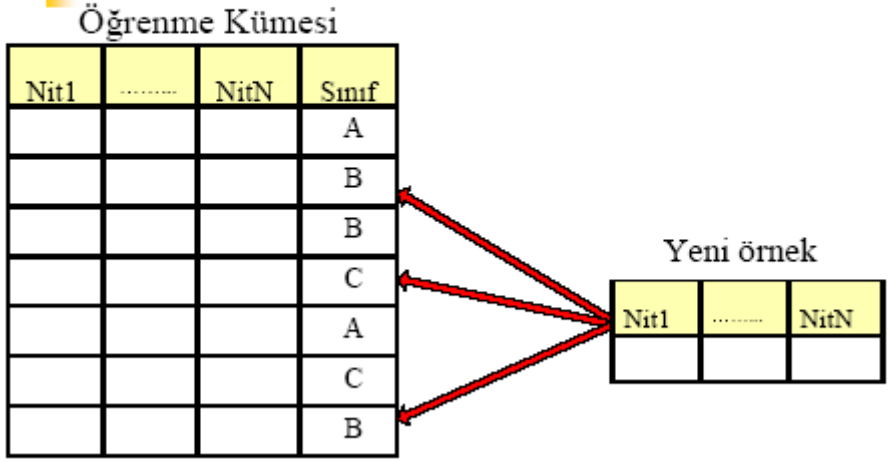
- Sınıflandırma yöntemleri
  - Örnek tabanlı yöntemler
    - k-En Yakın Komşu Yöntemi
  - Genetik Algoritmalar
  - Karar Destek Makinaları
  - Bulanık Küme Sınıflandırıcılar
  - Öngörü
    - Eğri Uydurma
- Model Değerlendirme
- Öğrenme, sınama, geçerleme kümelerini oluşturma
- Sınıflandırıcıları birleştirme



Örnek Tabanlı Yöntemler

- Örnek tabanlı sınıflandırma:
  - Öğrenme kümesi saklanır
  - Sınıflandırılacak yeni bir örnek geldiğinde öğrenme kümesi sınıf etiketini öngörmek için kullanılır (tembel (lazy) yöntemler)
- Yöntemler
  - k-en yakın komşu yöntemi

Örnek Tabanlı Yöntemler

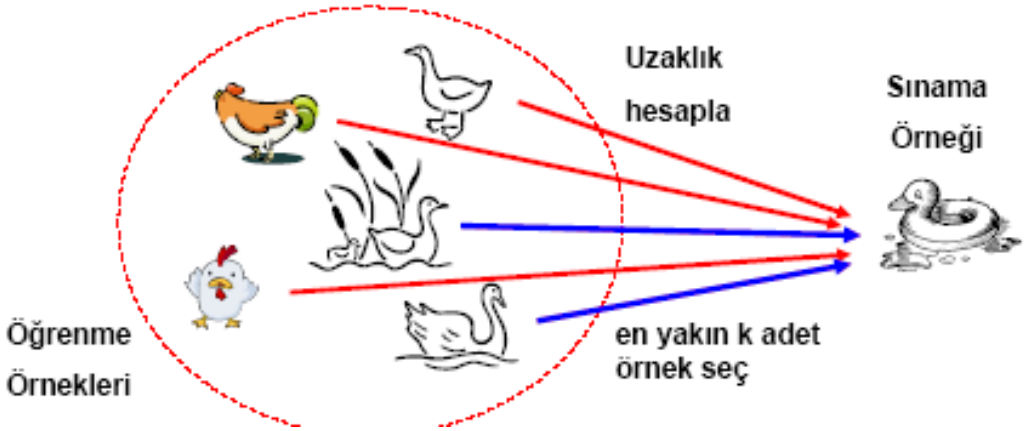


En Yakın Komşu Yöntemi

- Temel yaklaşım: Sınıflandırılmak istenen örneğe en yakın örnekleri bul.

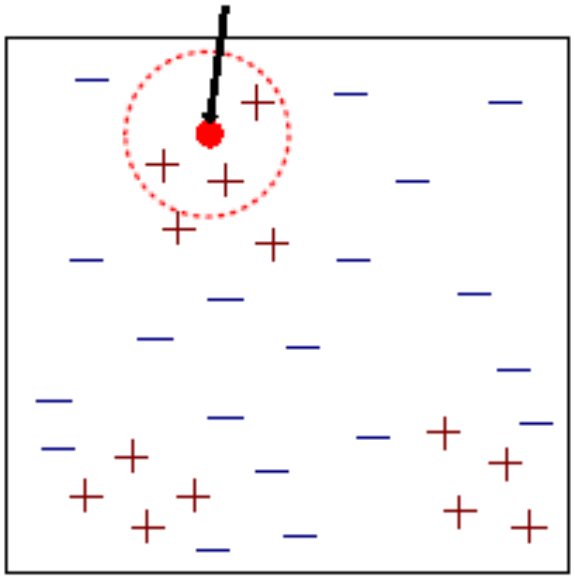
  Örnek: ördek gibi yürüyor, ördek gibi bağırıyor

  => büyük olasılıkla ördek

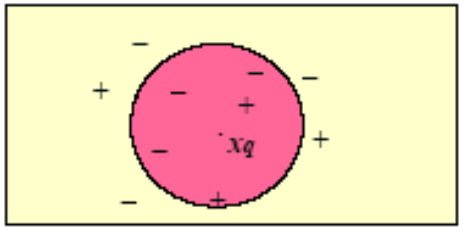


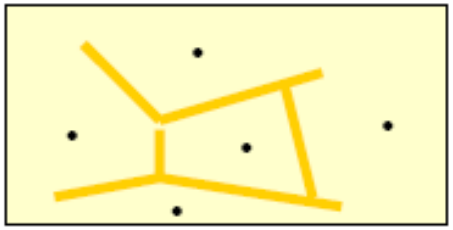
En Yakın Komşu Sınıflandırıcı

- Bütün örnekler n-boyutlu uzayda bir noktaya karşı düşürülür
- Nesneler arasındaki uzaklık (Öklid uzaklığı)
- Öğrenilen fonksiyon ayrık değerli veya gerçel değerli olabilir
- Ayrık değerli fonksiyonlarda k-komşu algoritması Xq örneğine en yakın k öğrenme örneğinde en sık görülen sınıf değerini verir
- Sürekli değerli fonksiyonlarda en yakın k öğrenme örneğinin ortalaması alınır



K-En Yakın Komşu Yöntemi





- *Xq* örneği 1-en yakın komşuya göre pozitif olarak, 5-en yakın komşuya göre negatif olarak sınıflandırılır

- Voronoi diyagramları: Her öğrenme örneğini çevreleyen dışbükey çokgenlerden oluşan karar yüzeyi

Konular

- Sınıflandırma yöntemleri
- Model Değerlendirme
  - Hata oranı
  - Kesinlik
  - Duyarlılık
  - F-ölçütü
- Öğrenme, sınama, geçerleme kümelerini oluşturma
- Sınıflandırıcıları birleştirme

Sınıflandırma Modelini Değerlendirme

- Model başarımını değerlendirme ölçütleri nelerdir?
  - Hata oranı
  - Kesinlik
  - Duyarlılık
  - F-ölçütü
- Model başarımını değerlendirme yöntemleri nelerdir?
- Farklı modellerin başarımı nasıl karşılaştırılır?

Sınıflandırma Hatası

- Sınıflandırma yöntemlerinin hatalarını ölçme
  - başarı: örnek doğru sınıfa atandı
  - hata: örnek yanlış sınıfa atandı
  - hata oranı: hata sayısının toplam örnek sayısına bölünmesi
- Hata oranı sınama kümesi kullanılarak hesaplanır

Model Başarımını Değerlendirme

- Model başarımını değerlendirme ölçütleri
  - modelin ne kadar doğru sınıflandırma yaptığını ölçer
    - hız, ölçeklenebilirlik gibi özellikleri değerlendirmez
- Karışıklık matrisi:

		ÖNGÖRÜLEN SINIF	
		Sınıf=1	Sınıf=-1
DOĞRU SINIF	Sınıf =1	a	b
	Sınıf =-1	c	d

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

Model Başarımını Değerlendirme: Doğruluk

		ÖNGÖRÜLEN SINIF	
		+1	-1
DOĞRU SINIF	+1	a (TP)	b (FN)
	-1	c (FP)	d (TN)

- Modelin başarımı:

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

$$Hata\ \ Orani = \frac{b+c}{a+b+c+d} = \frac{FN+FP}{TP+TN+FP+FN}$$

Örnek

Sınıflandırıcı A	
TP=25	FN=25
FP=25	TN=25

Doğruluk=%25

Sınıflandırıcı B	
TP=50	FN=0
FP=25	TN=25

Doğruluk=%75

Sınıflandırıcı C	
TP=25	FN=25
FP=0	TN=50

Doğruluk=%75

- Hangi sınıflandırıcı daha iyi?
  - B ve C, A'dan daha iyi bir sınıflandırıcı
  - B, C'den daha iyi bir sınıflandırıcı mı?

Model Başarımını Değerlendirme: Kesinlik

		ÖNGÖRÜLEN SINIF	
		+1	-1
DOĞRU SINIF	+1	a (TP)	b (FN)
	-1	c (FP)	d (TN)

$$\text{Kesinlik} = \frac{\text{Doğru sınıflandırılmış pozitif örnek sayısı}}{\text{Pozitif sınıflandırılmış örneklerin sayısı}} = \frac{TP}{TP + FP}$$

Model Başarımını Değerlendirme: Duyarlılık

		ÖNGÖRÜLEN SINIF	
		+1	-1
DOĞRU SINIF	+1	a (TP)	b (FN)
	-1	c (FP)	d (TN)

$$\underset{\text{Doğru pozitif oranı}}{\text{Duyarlılık}} = \frac{\text{Doğru sınıflandırılmış pozitif örnek sayısı}}{\text{Pozitif örneklerin sayısı}}$$

$$= \frac{TP}{TP + FN}$$

Kesinlik / Duyarlılık

- A modeli B modelinden daha iyi kesinlik ve duyarlılık değerine sahipse A modeli daha iyi bir sınıflandırıcıdır.
- Duyarlılık ve kesinlik arasında ters orantı var.

Sınıflandırıcıları Karşılaştırma

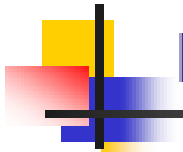
- Doğruluk en basit ölçüt

- Duyarlılık ve kesinlik daha iyi ölçme sağlıyor
  - Model A'nın duyarlılığı model B'den daha iyi ancak model B'nin kesinliği model A'dan daha iyi olabilir.

Model Başarımını Değerlendirme: F-ölçütü

- F-ölçütü: Kesinlik ve duyarlılığın harmonik ortalamasını alır.

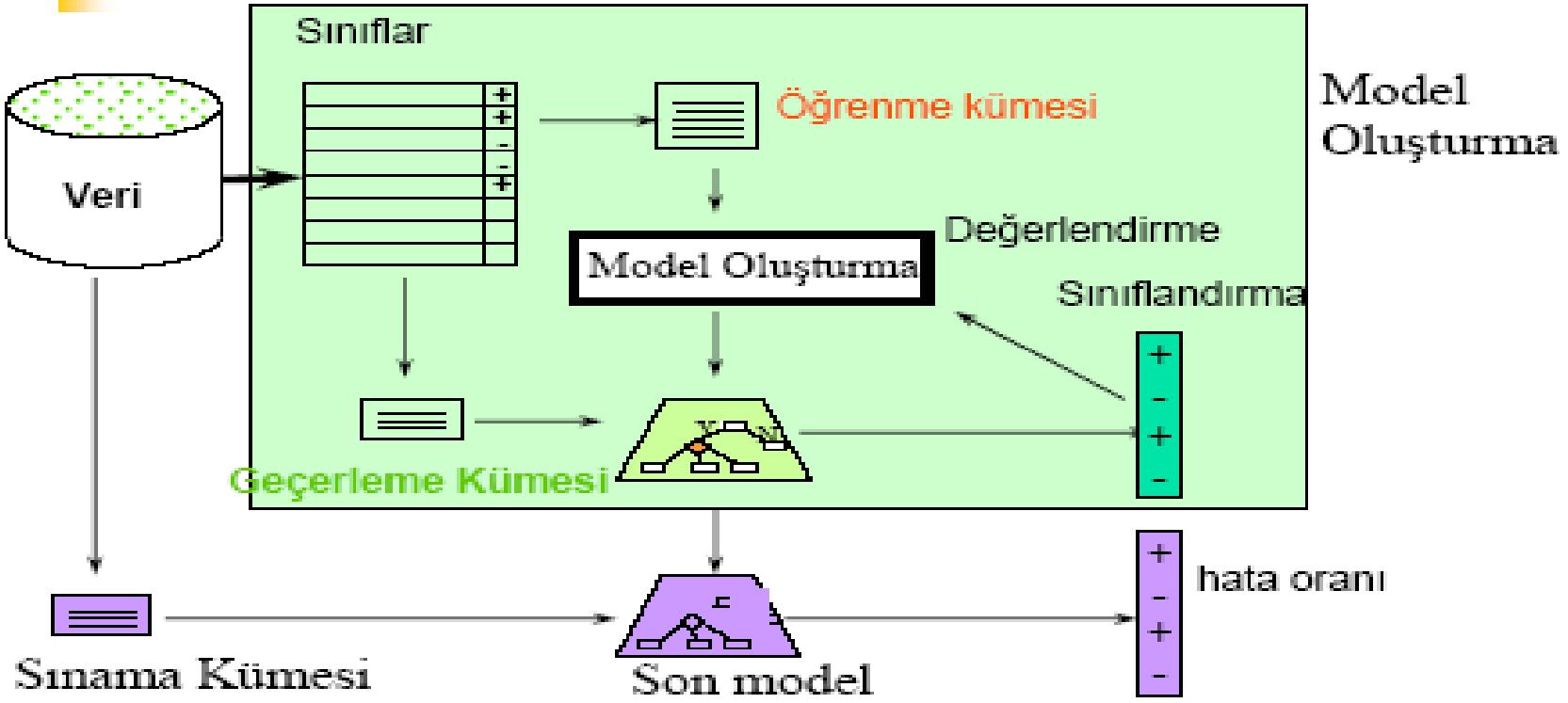
$$\text{F-ölçütü} = \frac{2 * \text{duyarlılık} * \text{kesinlik}}{\text{duyarlılık} + \text{kesinlik}}$$

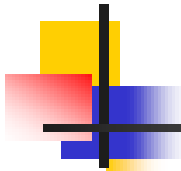


Model Parametrelerini Belirleme

- Sınama kümesi sınıflandırıcı oluşturmak için kullanılmaz
- Bazı sınıflandırıcılar modeli iki aşamada oluşturur
  - modeli oluştur
  - parametreleri ayarla
- Sınama kümesi parametreleri ayarlamak için kullanılmaz
- Uygun yöntem üç veri kümesi kullanma: öğrenme, geçerleme, sınama
  - geçerleme kümesi parametre ayarlamaları için kullanılır
  - model oluşturulduktan sonra öğrenme ve geçerleme kümesi son modeli oluşturmak için kullanılabilir

Sınıflandırma: Öğrenme, Geçerleme, Sınama





Konular

- Sınıflandırma yöntemleri
- Model Değerlendirme
- Öğrenme, sınama, geçerleme kümelerini oluşturma
  - holdout
  - k-kat çapraz geçerleme
  - Bootstrap
- Sınıflandırıcıları birleştirme

Verinin Dengesiz Dağılımı

- Küçük veya dengesiz veri kümeleri için örnekler tanımlayıcı olmayabilir
- Veri içinde bazı sınıflardan çok az örnek olabilir
  - tıbbi veriler: %90 sağlıklı, %10 hastalık
  - elektronik ticaret: %99 alışveriş yapmamış, %1 alışveriş yapmış
  - güvenlik: %99 sahtekarlık yapmamış, %1 sahtekarlık yapmış
- Örnek: Sınıf1: 9990 örnek, Sınıf2: 10 örnek
  - bütün örnekleri sınıf1'e atayan bir sınıflandırıcının hata oranı: 9990 / 10000= %99,9
    - hata oranı yanıltıcı bir ölçüt olabilir

Dengeli Dağılım Nasıl Sağlanır?

- Veri kümesinde iki sınıf varsa
  - iki sınıfın eşit dağıldığı bir veri kümesi oluştur
  - Az örneği olan sınıftan istenen sayıda rastgele örnekler seç
  - Çok örneği olan sınıftan aynı sayıda örnekleri ekle
- Veri kümesinde iki sınıftan fazla sınıf varsa
  - Öğrenme ve sınama kümesini farklı sınıflardan aynı sayıda örnek olacak şekilde oluştur

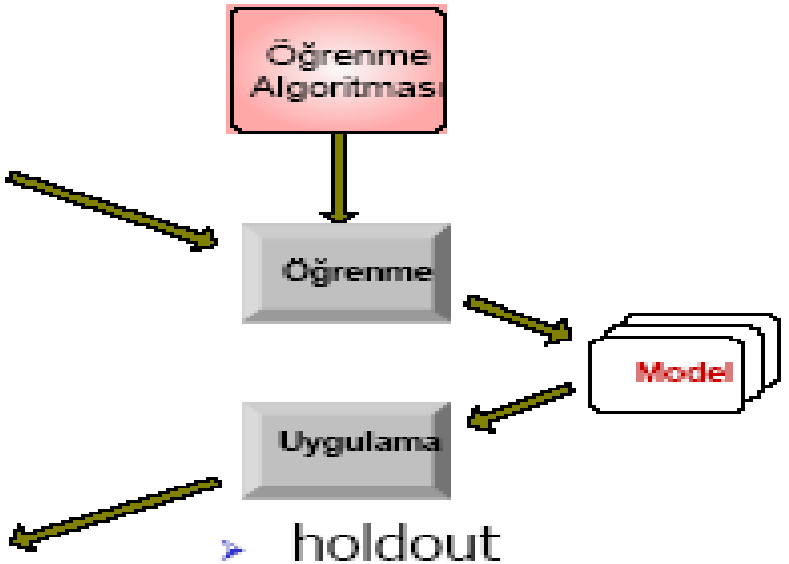
Örnek

Tid	Hİ1	Hİ2	Hİ3	Sınıf
1	1	Büyük	125K	0
2	0	Orta	100K	0
3	0	Küçük	70K	0
4	1	Orta	120K	0
5	0	Büyük	95K	1
6	0	Orta	60K	0
7	1	Büyük	220K	0
8	0	Küçük	85K	1
9	0	Orta	75K	0
10	0	Küçük	90K	1

Öğrenme Kümesi

Tid	Hİ1	Hİ2	Hİ3	Sınıf
11	0	Küçük	55K	?
12	1	Orta	80K	?
13	1	Büyük	110K	?
14	0	Küçük	95K	?
15	0	Büyük	67K	?

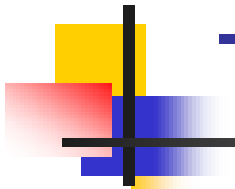
Sınama Kümesi



- holdout
- repeated holdout
- k-fold cross validation
- bootstrapping

Büyük Veri Kümelerinde Değerlendirme

- Veri dağılımı dengeli ise: Veri kümesindeki örnek sayısı ve her sınıfa ait örnek sayısı fazla ise basit bir değerlendirme yeterli
  - *holdout* yöntemi: Belli sayıda örnek sınama için ayrılır, geriye kalan örnekler öğrenme için kullanılır
    - genelde veri kümesinin 2/3'ü öğrenme, 1/3'i sınama kümesi olarak ayrılır
  - öğrenme kümesi kullanılarak model oluşturulur ve sınama kümesi kullanılarak model değerlendirilir

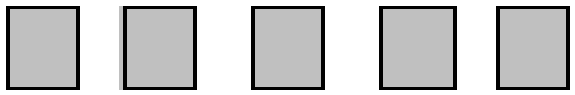


Tekrarlı Holdout Yöntemini

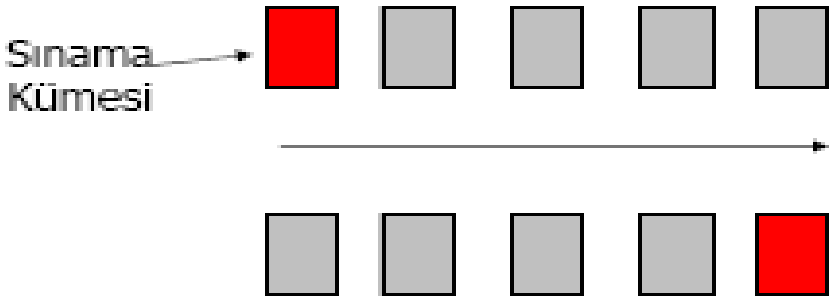
- Veri kümesini farklı altkümelere bölerek holdout yöntemini tekrarlama
  - Her eğitme işleminde veri kümesinin belli bir bölümü öğrenme kümesi olarak ayrılır
  - Modelin hata oranı, işlemler sonunda elde edilen modellerin hata oranlarının ortalaması
- Problem: Farklı eğitme işlemlerindeki sınama kümeleri örtüşebilir

k-Kat Çapraz Geçerleme

- Veri kümesi eşit boyutta *k* adet farklı gruba ayrılır.



- Bir grup sınama, diğerleri öğrenme için ayrılır.



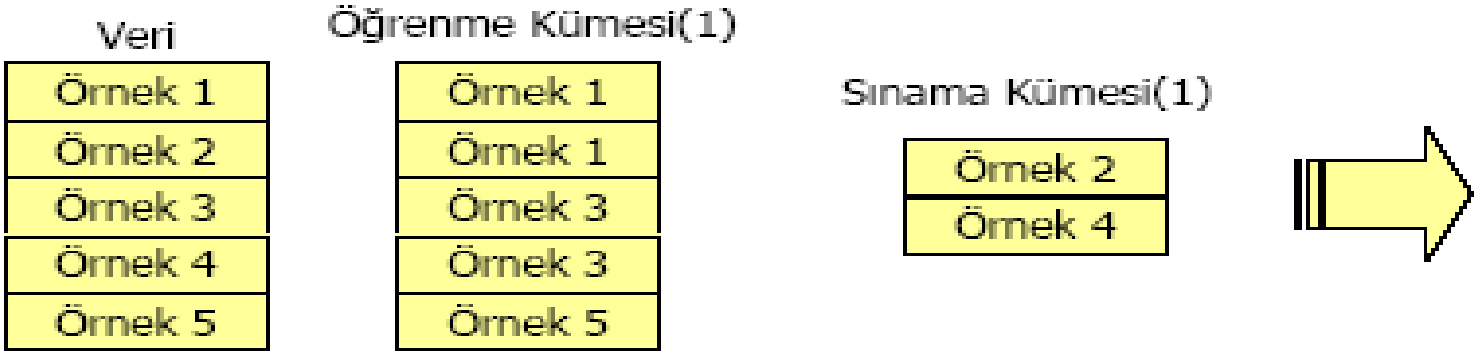
- Her grup bir kere sınama kümesi olacak şekilde deneyler *k* kere tekrarlanır.

Biri Hariç Çapraz Geçerleme

- k-kat çapraz geçerlemenin özel hali
  - k sayısı veri kümesindeki örnek sayısına (N) eşit
- Model *N-1* örnek üzerinde eğitilir, dışarıda bırakılan *1* örnek üzerinde sınanır
- Bu işlem her örnek *1* kez sınama için kullanılacak şekilde tekrarlanır
  - model *N* kez eğitilir
- Model başarımı denemelerin başarımının ortalaması
- Verinin en etkin şekilde kullanımı

Bootstrap Yöntemi

- Veri kümesinden yerine koyma yöntemi ile örnekleri seçerek öğrenme kümesi oluşturur
  - N örnekten oluşan veri kümesinden yerine koyarak N örnek seç
  - Bu kümeyi öğrenme kümesi olarak kullan
  - Öğrenme kümesinde yer almayan örnekleri sınama kümesi olarak kullan



0.632 bootstrap

- N örnekten oluşan bir veri kümesinde bir örneğin seçilmeme olasılığı: $1-\frac{1}{N}$
- Sınama kümesinde yer alma olasılığı:

$$\left(1-\frac{1}{n}\right)^n \approx e^{-1} = 0.368$$

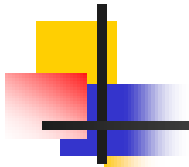
- Sınama kümesi veri kümesindeki örneklerin %63,2'sinden oluşuyor

Bootstrap Yönteminde Model Hatasını Belirleme

- Model başarımını sadece sınama kümesi kullanarak belirleme kötümser bir yaklaşım
  - model örneklerin sadece ~%63'lük bölümüyle eğitiliyor
- Model başarımı hem öğrenme kümesindeki hem de sınama kümesindeki başarım ile değerlendirilir

$$\text{hata} = 0{,}632\ \text{hata}_{(\text{sınama})} + 0{,}368\ \text{hata}_{(\text{öğrenme})}$$

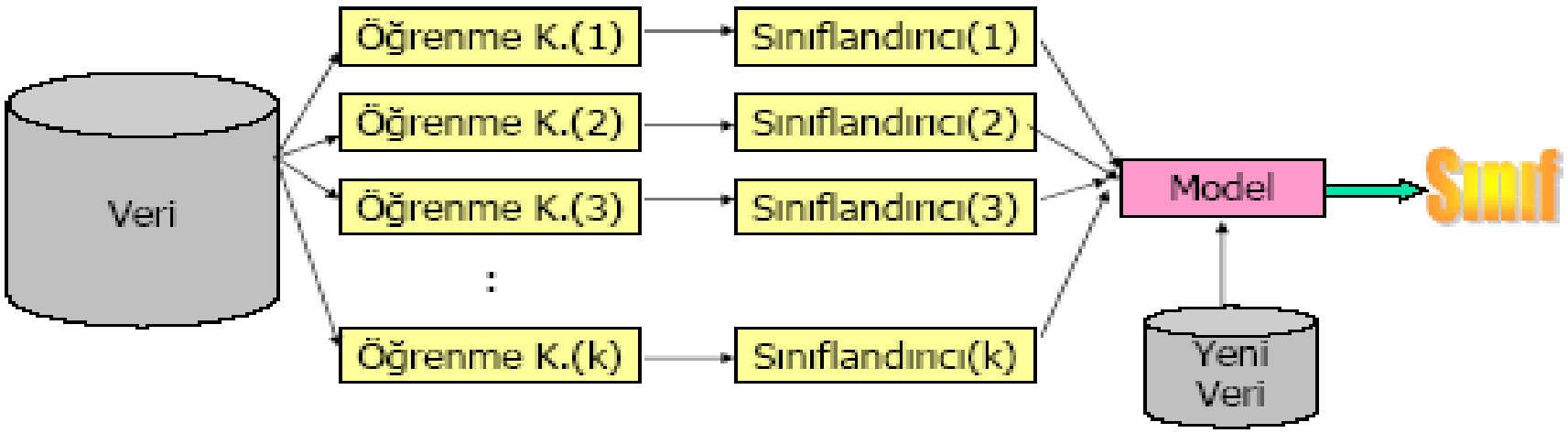
- İşlem birkaç kez tekrarlanarak hatanın ortalaması alınır.



Konular

- Sınıflandırma yöntemleri
- Model Değerlendirme
- Öğrenme, sınama, geçerleme kümelerini oluşturma
- Sınıflandırıcıları birleştirme
  - Bagging
  - Boosting

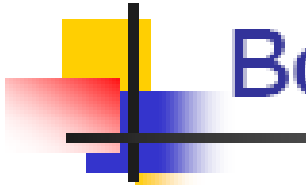
Model Başarımını Artırma



- Bir grup sınıflandırıcı kullanma
  - Bagging
  - Boosting

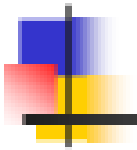
Bagging

- *N* örnekten oluşan bir veri kümesinde bootstrap yöntemi ile *T* örnek seç
- Bu işlemi *k* öğrenme kümesi oluşturmak üzere tekrarla
- Aynı sınıflandırma algoritmasını *k* öğrenme kümesi üzerinde kullanarak *k* adet sınıflandırıcı oluştur
- Yeni bir örneği sınıflandırmak için her sınıflandırıcının sonucunu öğren
- Yeni örnek en çok hangi sınıfa atanmışsa o sınıfın etiketiyle etiketlendir.

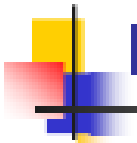


Boosting

- Öğrenme kümesindeki her örneğin bir ağırlığı var
- Her öğrenme işleminden sonra, her sınıflandırıcı için yapılan sınıflandırma hatasına bağlı olarak örneklerin ağırlığı güncelleniyor
- Yeni bir örneği sınıflandırmak için her sınıflandırıcının doğruluğuna bağlı olarak ağırlıklı ortalaması alınıyor.



VERİ MADENCİLİĞİ
Demetleme Yöntemleri

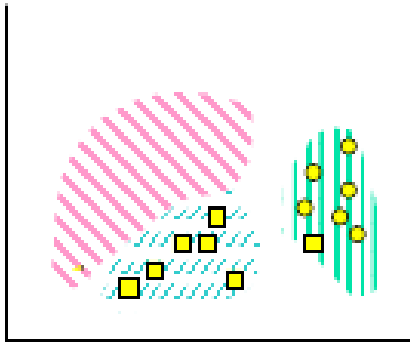


Demetleme

- Veri kümesini uygun gruplara ya da demetlere ayırmak
- Kullanıcının veri dağılımını anlamasını sağlar
  - kullanıcıya veri hakkında genel bilgi sunar
- Demet: birbirine benzeyen nesnelerden oluşan grup
  - Aynı demetteki nesneler birbirine daha çok benzer
  - Farklı demetlerdeki nesneler birbirine daha az benzer

Demetleme

- Nesneleri demetlere (gruplara) ayırma
- Gözetimsiz öğrenme: Hangi nesnenin hangi sınıfa ait olduğu ve sınıf sayısı belli değil
- Uygulamaları:
  - verinin dağılımını anlama
  - başka veri madenciliği uygulamarı için ön hazırlık



Demetleme Uygulamaları

- Örüntü tanıma
- Görüntü işleme
- Ekonomi
- Aykırılıkları belirleme
- WWW
  - Doküman demetleme
  - Kullanıcı davranışlarını demetleme
  - Kullanıcıları demetleme

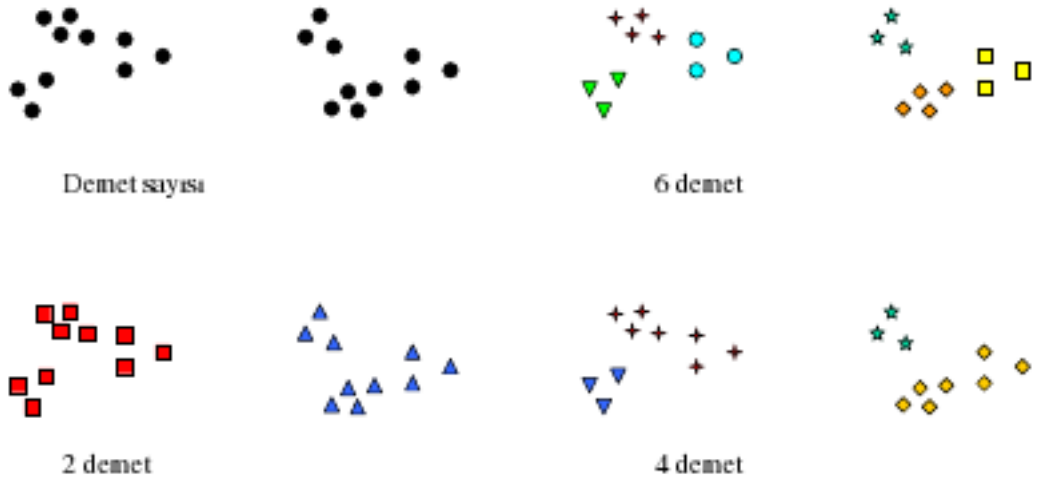
Veri Madenciliğinde Demetleme

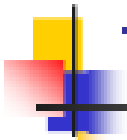
- Ölçeklenebilirlik
- Farklı tipteki niteliklerden oluşan nesneleri demetleme
- Farklı şekillerdeki demetleri oluşturabilme
- En az sayıda giriş parametresi gereksinimi
- Hatalı veriler ve aykırılıklardan en az etkilenme
- Model oluşturma sırasında örneklerin sırasından etkilenmeme
- Çok boyutlu veriler üzerinde çalışma
- Kullanıcıların kısıtlarını göz önünde bulundurma
- Sonucun yorumlanabilir ve anlaşılabilir olması

İyi Demetleme

- İyi demetleme yöntemiyle elde edilen demetlerin özellikleri
  - aynı demet içindeki neszneler arası benzerlik fazla
  - farklı demetlerde bulunan nesneler arası benzerlik az
- Oluşan demetlerin kalitesi seçilen benzerlik ölçütüne bağlı
  - Uzaklık / Benzerlik nesnelerin nitelik tipine göre değişir
    - Nesneler arası benzerlik: s(i,j)
    - Nesneler arası uzaklık: d(i,j) = 1 – s(i,j)
- İyi bir demetleme yöntemi veri içinde gizlenmiş örüntüleri bulabilmeli
- Veriyi gruplama için uygun demetleme kriteri bulunmalı
- Demetleme sonucunun kalitesi seçilen demetlerin şekline ve temsil edilme yöntemine bağlı

Farklı Demetler





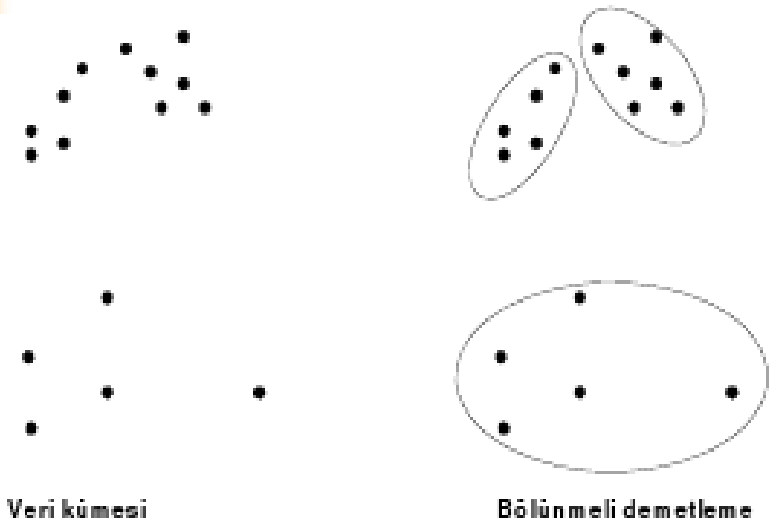
Temel Demetleme Yaklaşımları

- Bölünmeli yöntemler: Veriyi bölerek, her grubu belirlenmiş bir kritere göre değerlendirir
- Hiyerarşik yöntemler: Veri kümelerini (ya da nesneleri) önceden belirlenmiş bir kritere göre hiyerarşik olarak ayırır
- Yoğunluk tabanlı yöntemler: Nesnelerin yoğunluğuna göre demetleri oluşturur
- Model tabanlı yöntemler: Her demedin bir modele uyduğu varsayılır. Amaç bu modellere uyan verileri gruplamak

Bölünmeli Yöntemler

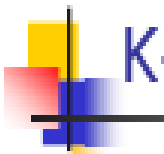
- Amaç: *n* nesneden oluşan bir veri kümesini (*D*) *k* (*k*≤*n*) demede ayırmak
  - her demette en az bir nesne bulunmalı
  - her nesne sadece bir demette bulunmalı
- Yöntem: Demetleme kriterini enbüyütücek şekilde *D* veri kümesi için istenen sayıda *k* grubu belirleme
  - Global çözüm: Mümkün olan tüm bölünmeleri yaparak en iyisini seçme (NP karmaşık)
  - Sezgisel çözüm: k-means ve k-medoids
    - k-means (MacQueen'67): Her demet kendi merkezi ile temsil edilir
    - k-medoids veya PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Her demet, demette bulunan bir nesne ile temsil edilir

Bölünmeli Demetleme



Veri kümesi

Bölünmeli demetleme

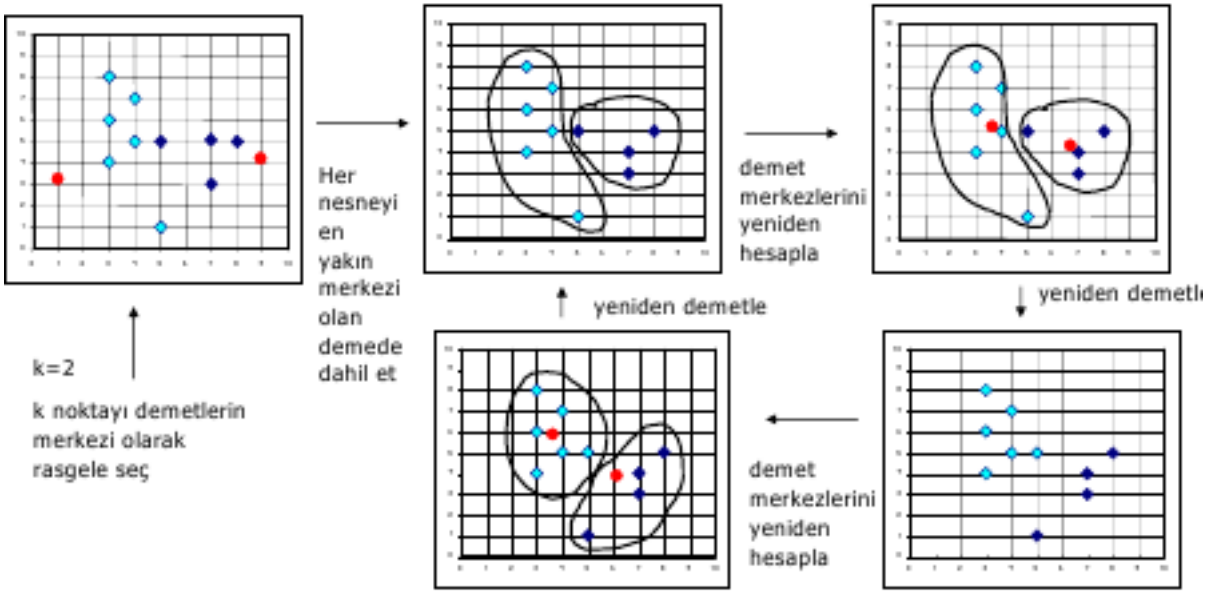


K-means Demetleme

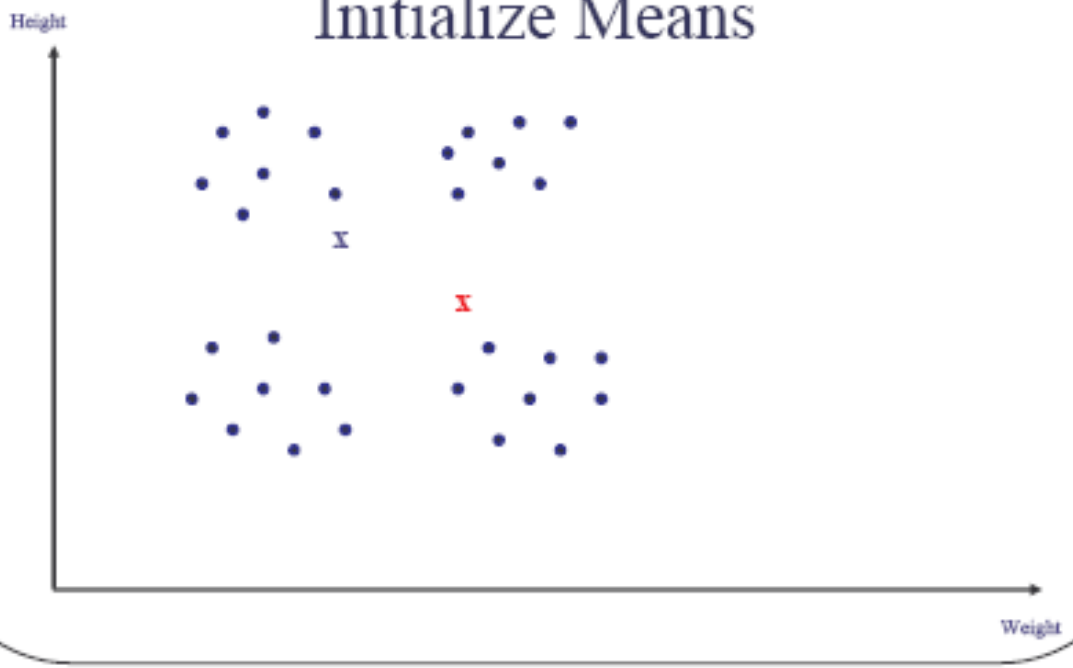
- Bilinen bir k değeri için k-means demetleme algoritmasının 4 aşaması vardır:
  - Veri kümesi k altkümeye ayrılır (her demet bir altküme)
  - Her demedin ortalaması hesaplanır: merkez nokta (demetteki nesnelerin niteliklerinin ortalaması)
  - Her nesne en yakın merkez noktanın olduğu demede dahil edilir
  - Nesnelerin demetlenmesinde değişiklik olmayana kadar adım 2'ye geri dönülür.

K-means Demetleme Yöntemi

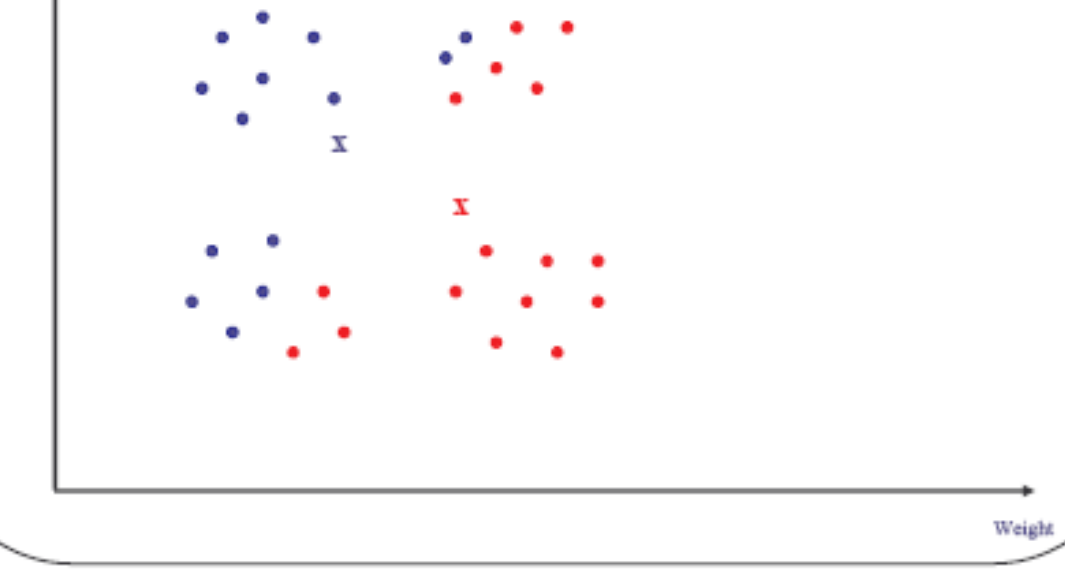
- Örnek



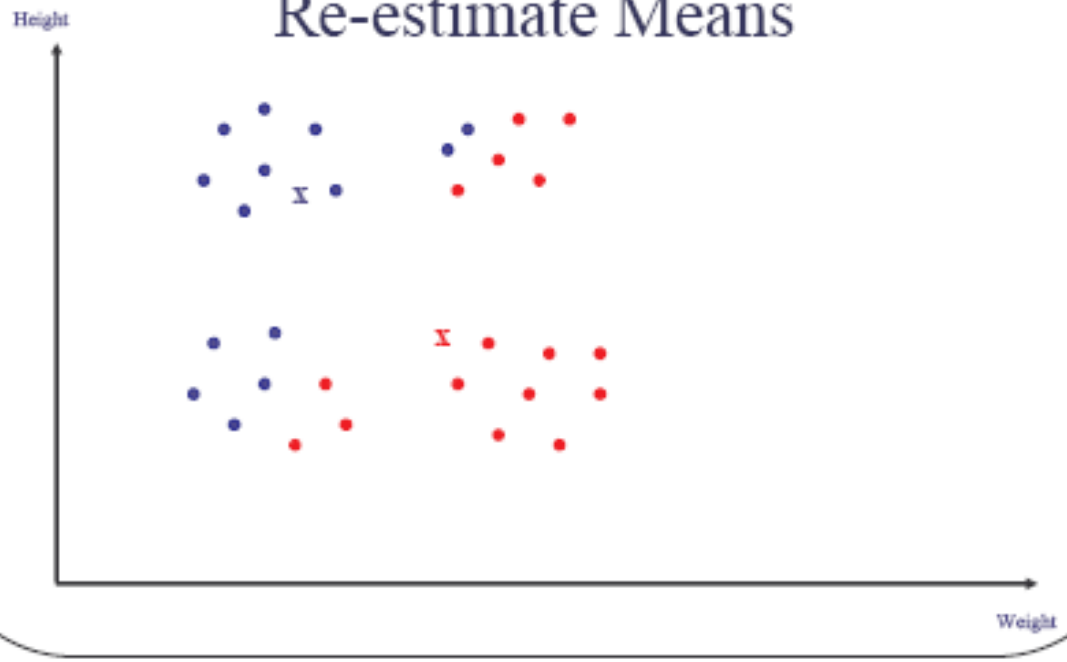
K Means Example (k=2)
Initialize Means



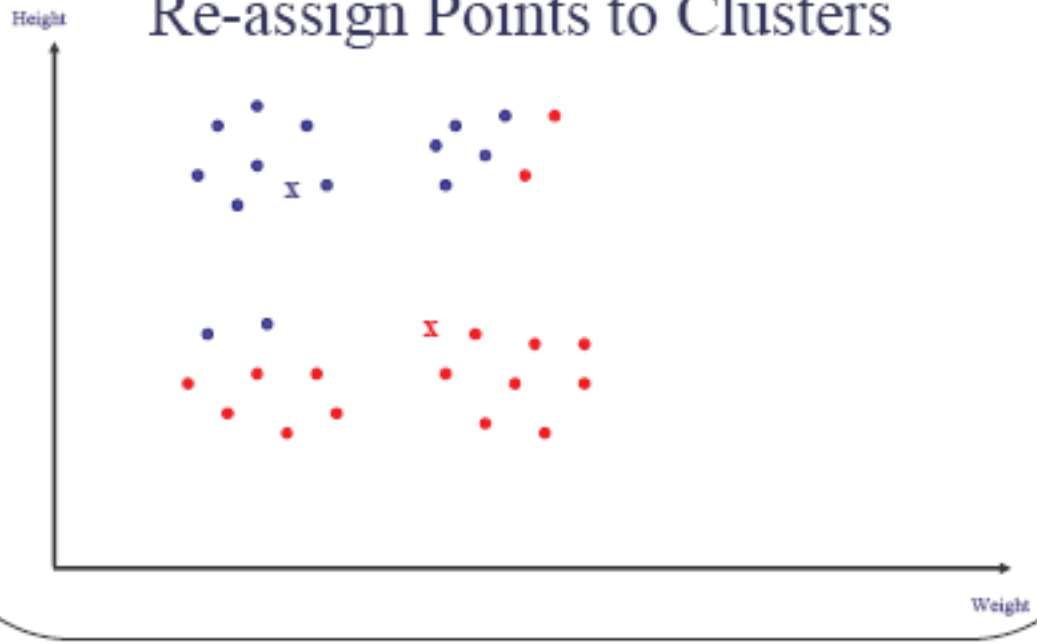
K Means Example
Assign Points to Clusters



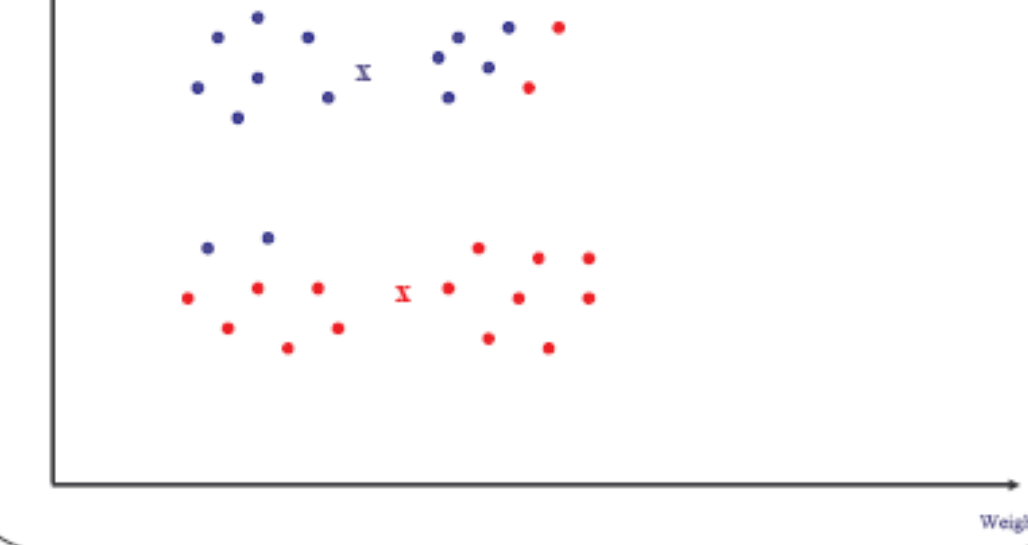
K Means Example
Re-estimate Means



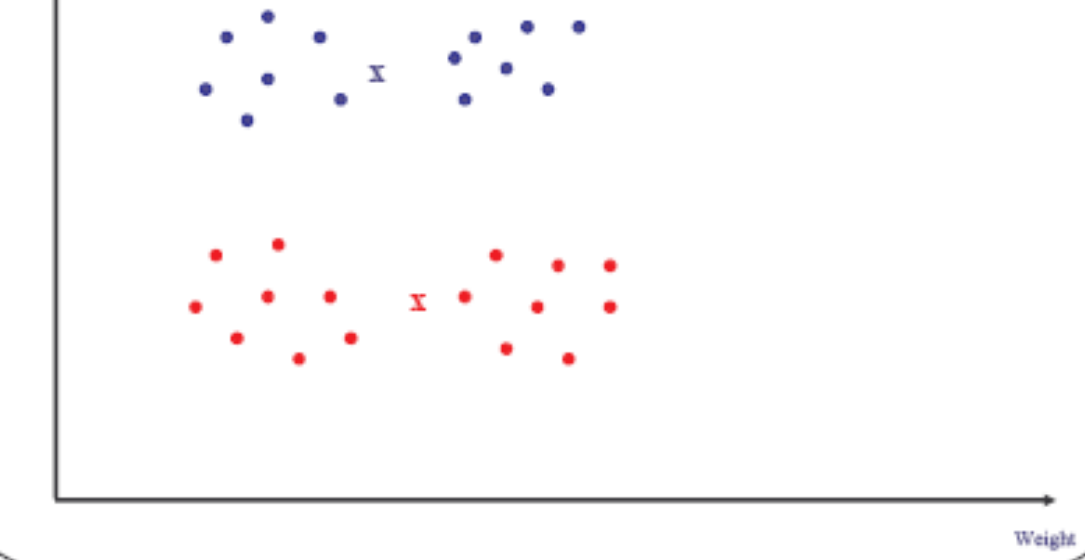
K Means Example
Re-assign Points to Clusters



K Means Example
Re-estimate Means

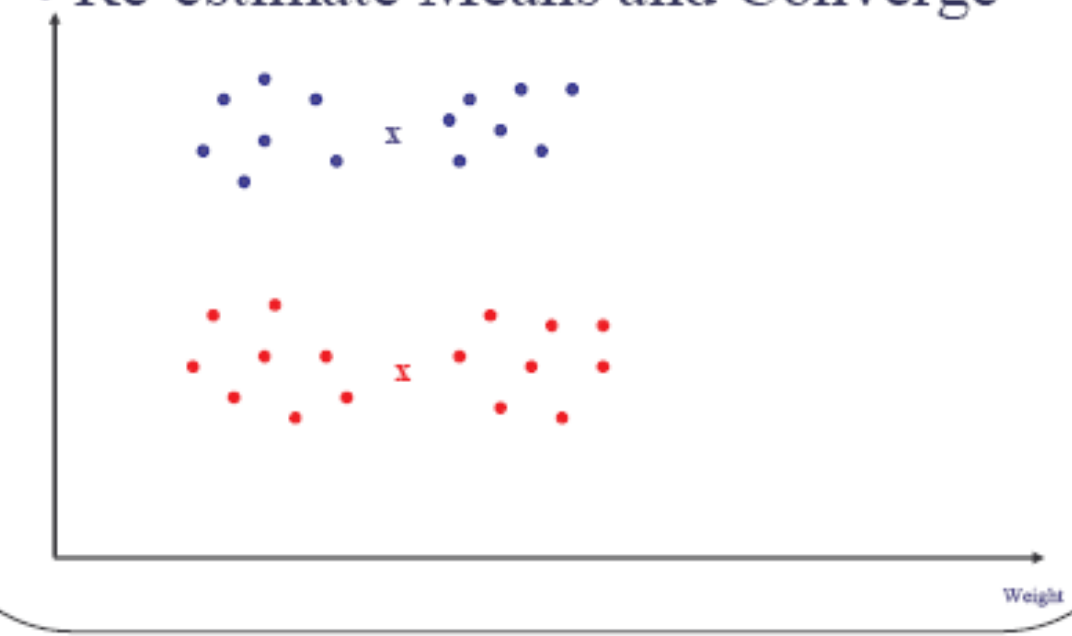


K Means Example
Re-assign Points to Clusters

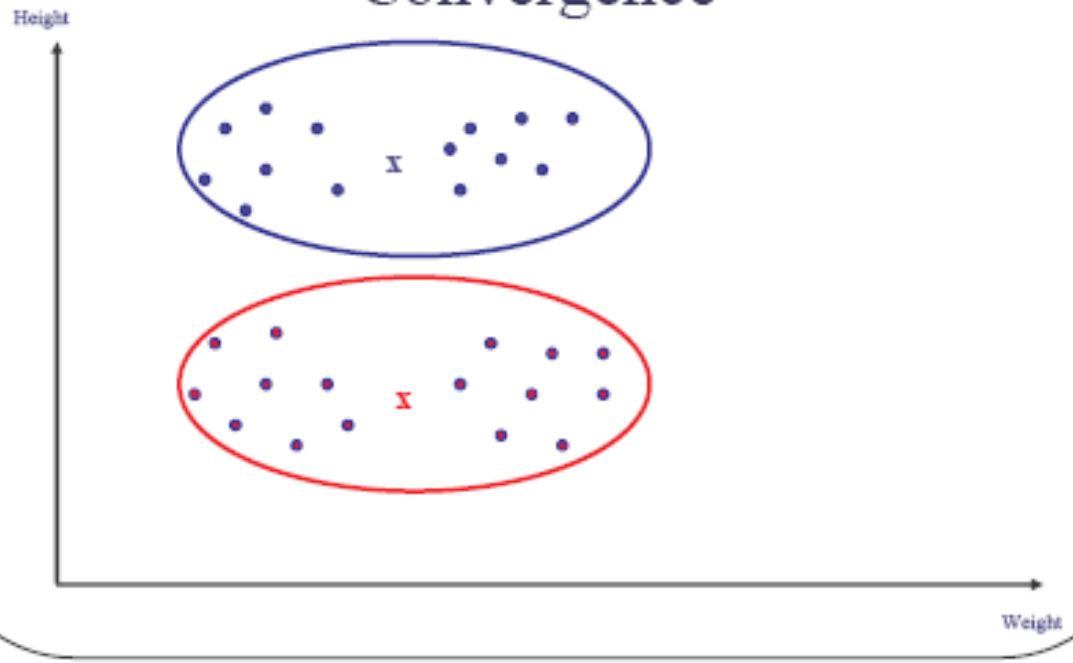


K Means Example

Re-estimate Means and Converge



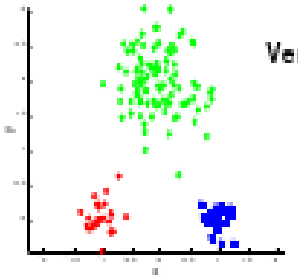
K Means Example
Convergence



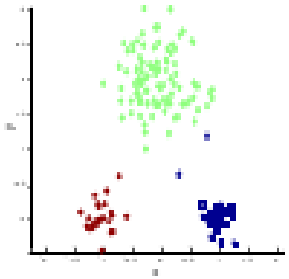
k-means Demetleme Yöntemi

- Demet sayısının belirlenmesi gerekir
- Başlangıçta demet merkezleri rasgele belirlenir
  - Her uygulamada farklı demetler oluşabilir
- Benzerlik Öklid uzaklığı, cosine benzerliği gibi yöntemlerle ölçülebilir
- Az sayıda tekrarda demetler oluşur
  - Yakınsama koşulu çoğunlukla az sayıda nesnenin demet değiştirmesi şekline dönüştürülür
- Karmaşıklığı: $O(ktn)$
  - k: demet sayısı, t: tekrar sayısı, n: nesne sayısı

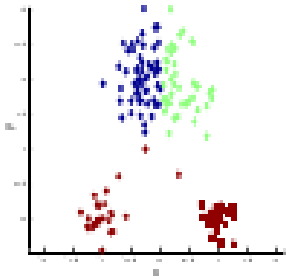
K-Means: İki Farklı Demetleme



Veri Kümesi



Optimal Demetleme



Lokal optimum

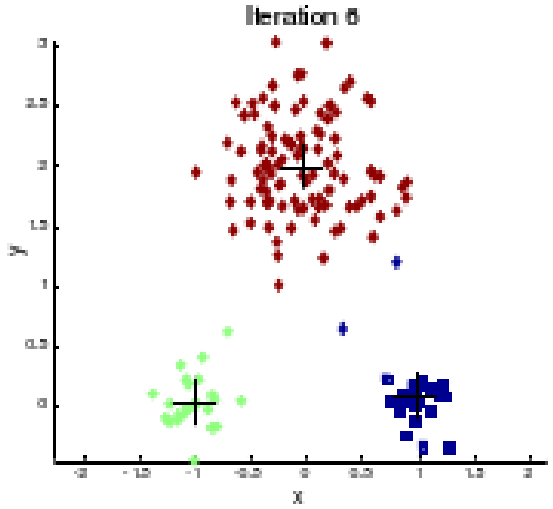
K-Means Demetleme Yöntemini Değerlendirme

- Yaygın olarak kullanılan yöntem hataların karelerinin toplamı (Sum of Squared Error SSE)
  - Nesnelerin bulundukları demedin merkez noktalarına olan uzaklıklarının karelerinin toplamı

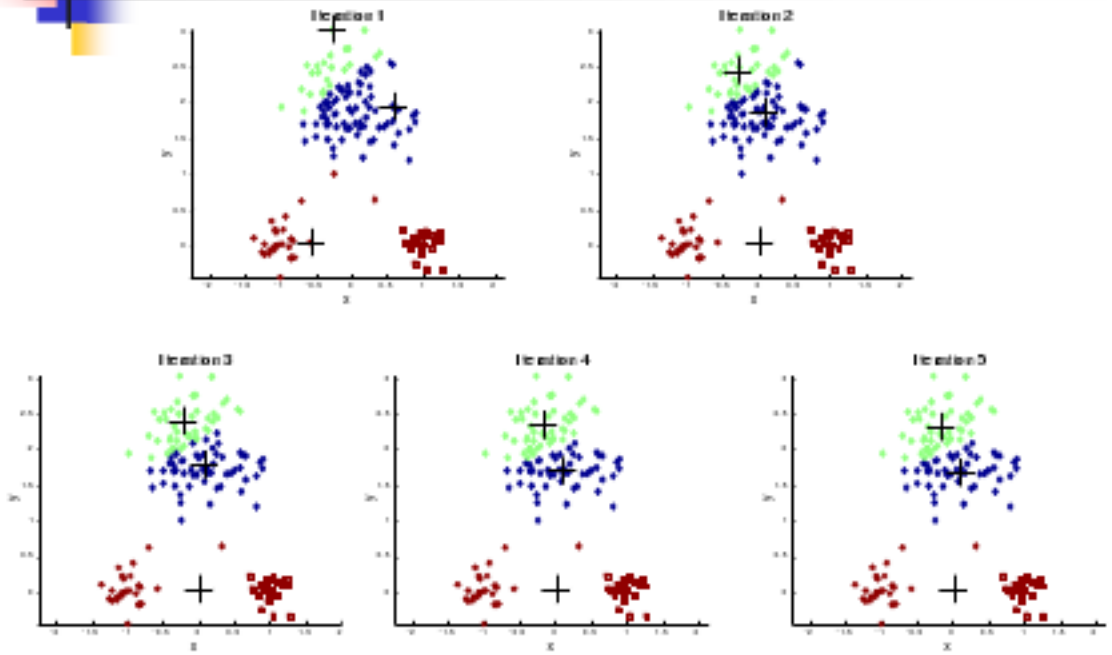
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

  - x, C_i demedinde bulunan bir nesne, m_i C_i demedinin merkez noktası
- Hataların karelerinin toplamını azaltmak için k demet sayısı artırılabilir
  - Küçük k ile iyi bir demetleme, büyük k ile kötü bir demetlemeden daha az SSE değerine sahip olabilir.
- Başlangıç için farklı merkez noktaları seçerek farklı demetlemeler oluşturulur
- En az SSE değerini sahip olan demetleme seçilir

Merkez Noktaların Seçimi



Merkez Noktaların Seçimi



K-Means Demetleme Çeşitleri

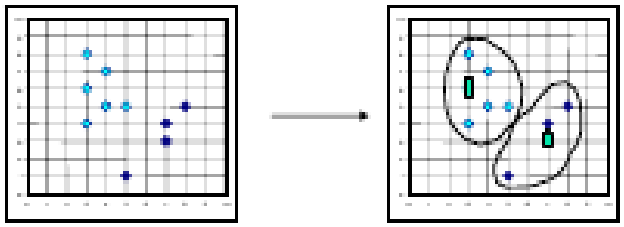
- K-Means demetlemeye başlamadan önce yapılanlar
  - Veri kümesini örnekleyerek hiyerarşik demetleme yap. Oluşan k demedin ortalamasını başlangıç için merkez nokta seç
  - Başlangıçta k'dan fazla merkez nokta seç. Daha sonra bunlar arasından k tane seç.
- K-Means demetleme işlemi sonrasında yapılanlar
  - Küçük demetleri en yakın başka demetlere dahil et
  - En büyük toplam karesel hataya sahip olan demedi böl
  - Merkez noktaları birbirine en yakın demetleri birleştir
  - Toplam karesel hatada en az artışa neden olacak iki demedi birleştir

K-Means Demetleme Algoritmasının Özellikleri

- Gerçeklemesi kolay
- Karmaşıklığı diğer demetleme yöntemlerine göre az
- K-Means algoritması bazı durumlarda iyi sonuç vermeyebilir
  - Veri grupları farklı boyutlarda ise
  - Veri gruplarının yoğunlukları farklı ise
  - Veri gruplarının şekli küresel değilse
  - Veri içinde aykırılıklar varsa

K-Medoids Demetleme Yöntemi

- Her demedi temsil etmek için demet içinde orta nokta olan nesne seçilir.
  - 1, 3, 5, 7, 9 ortalama: 5
  - 1, 3, 5, 7, 1009 ortalama 205
  - 1, 3, 5, 7, 1009 orta nokta 5



K-Medoids Demetleme Yöntemi

- PAM (Partitioning Araound Medoids 1987)
  - Başlangıçta *k* adet nesne demetleri temsil etmek üzere rasgele seçilir x_{ik}
  - Kalan nesneler en yakın merkez nesnenin bulunduğu demede dahil edilir
  - Merkez nesne olmayan rasgele bir nesne seçilir x_{rk}
  - x_{rk} merkez nesne olursa toplam karesel hatanın ne kadar değiştiğini bulunur

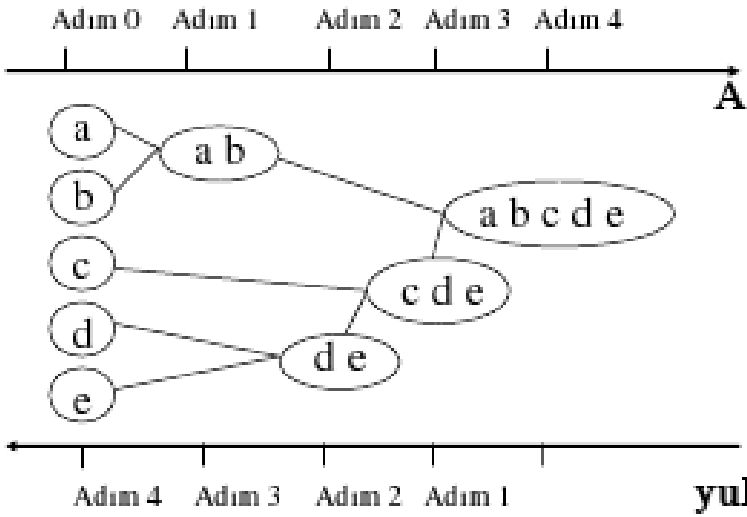
$$TC_{ik} = \sum_{j=1}^{n_k} (x_{ik} - x_{jk})^2 - \sum_{j=1}^{n_k} (x_{rk} - x_{jk})^2$$

n_k: *k* demedi içindeki nesne sayısı
x_{jk}: *k* demedi içindeki *j.* nesne

  - $TC_{ik} < 0$ ise O_{rk} merkez nesne olarak atanır.
  - Demetlerde değişiklik oluşmayana kadar 3. adıma geri gidilir.
- Küçük veri kümeleri için iyi sonuç verebilir, ancak büyük veri kümeleri için uygun değil
- *CLARA* (Kaufmann & Rousseeuw, 1990)
- *CLARANS* (Ng & Han, 1994)

Hiyerarşik Demetleme

- Demet sayısının belirlenmesine gerek yok
  - Sonlanma kriteri belirlenmesi gerekiyor



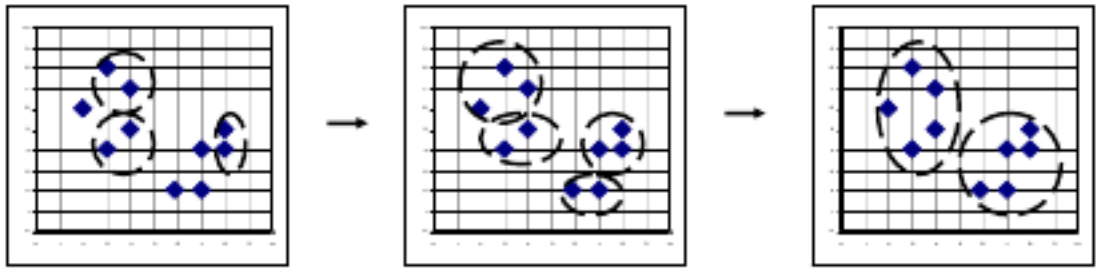
agglomerative:
Aşağıdan yukarıya
(AGNES)

divisive:
yukarıdan aşağıya

(DIANA)

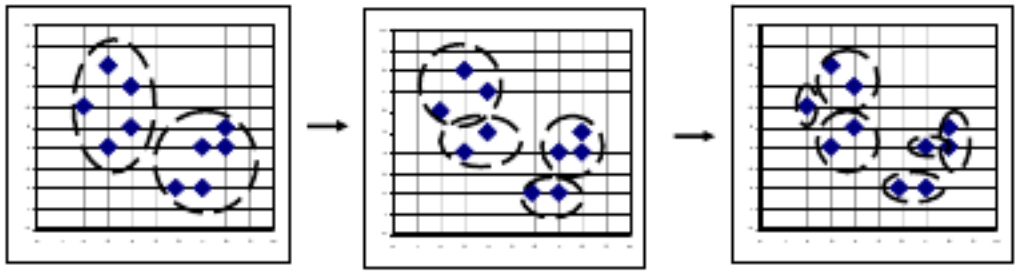
Hiyerarşik Yöntemler

- *AGNES* (*AG*glomerative *NES*ting):
  - Kaufmann ve Rousseeuw tarafından 1990 yılında önerilmiştir.
  - Birinci adımda her nesne bir demet oluşturur.
  - Aralarında en az uzaklık bulunan demetler her adımda birleştirilir.
  - Bütün nesneler tek bir demet içinde kalana kadar ya da istenen sayıda demet elde edene kadar birleştirme işlemi devam eder.



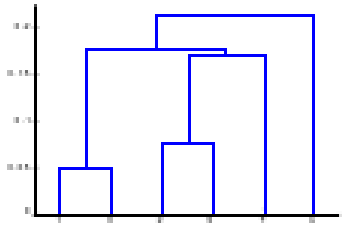
Hiyerarşik Yöntemler

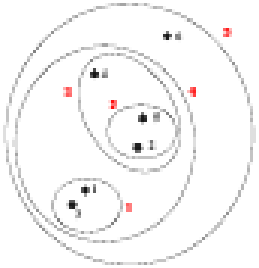
- *DIANA* (*DI*visive *ANA*lysis):
  - Kaufmann ve Rousseeuw tarafından 1990 yılında önerilmiştir.
  - AGNES'in yaptığı işlemlerin tersini yapar.
  - En sonunda her nesne bir demet oluşturur.
  - Her nesne ayrı bir demet oluşturana ya da istenilen demet sayısı elde edene kadar ayrılma işlemi devam eder.



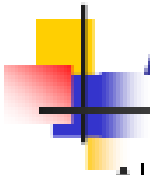
Hiyerarşik Demetleme

- Dendogram şeklinde görüntülenebilir
  - Her seviyede demetlerin birleşmesini vaya ayrılmasını belirten ağaç yapısı
- Dendogram istenen seviyede kesilerek demetler elde edilir





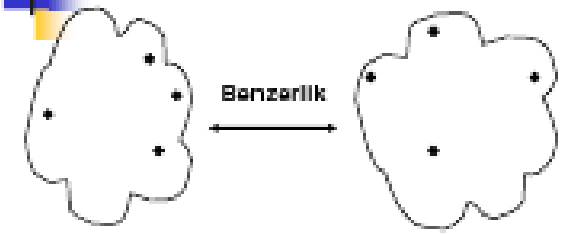
- **Dendogram: Demetler hiyerarşik olarak ağaç yapısı şeklinde görüntülenebilir**
- **Ara düğümler çocuk düğümlerdeki demetlerin birleşmesiyle elde edilir**
  - **Kök: bütün nesnelerden oluşan tek demet**
  - **Yapraklar: bir nesneden oluşan demetler**
- **Dendogram istenen seviyede kesilerek demetler elde edilir**



Aşağıdan Yukarıya Demetleme

- Algoritma
  1. Uzaklık matrisini hesapla
  2. Her nesne bir demet
  3. Tekrarla
  4. En yakın iki demedi birleştir
  5. Uzaklık matrisini yeniden hesapla
  6. Sonlanma: Tek bir demet kalana kadar
- Uzaklık matrisini hesaplarken farklı yöntemler farklı demetleme sonuçlarına neden olurlar

Demetler Arası Uzaklık

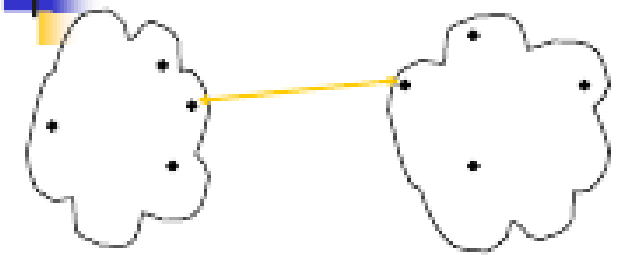


- MIN (Tek Bağ)
- MAX (Tam Bağ)
- Ortalama
- Merkezler arası uzaklık

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

Uzaklık Matrisi

Demetler Arası Uzaklık

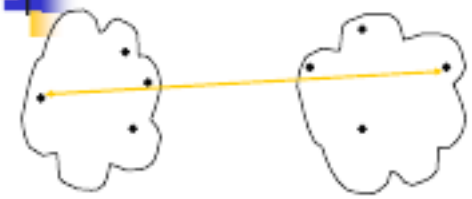


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

Uzaklık Matrisi

- MIN (Tek Bağ)
- MAX (Tam Bağ)
- Ortalama
- Merkezler arası uzaklık

Demetler Arası Uzaklık

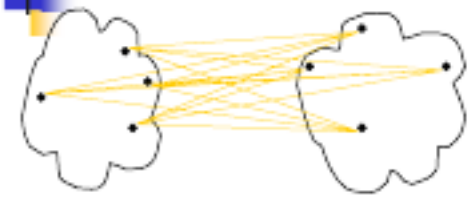


- MIN (Tek Bağ)
- MAX (Tam Bağ)
- Ortalama
- Merkezler arası uzaklık

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Uzaklık Matrisi

Demetler Arası Uzaklık

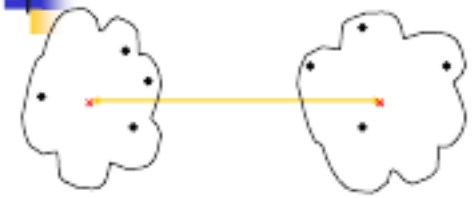


- MIN (Tek Bağ)
- MAX (Tam Bağ)
- Ortalama
- Merkezler arası uzaklık

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

Uzaklık Matrisi

Demetler Arası Uzaklık

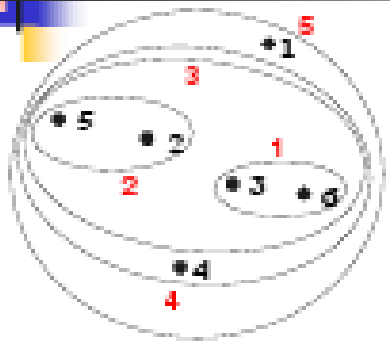


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

Uzaklık Matrisi

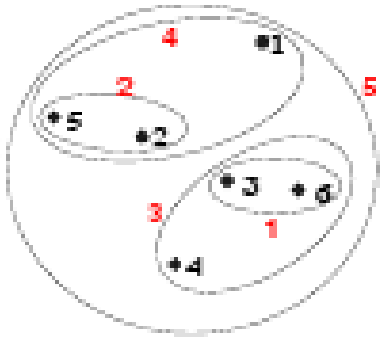
- MIN (Tek Bağ)
- MAX (Tam Bağ)
- Ortalama
- Merkezler arası uzaklık

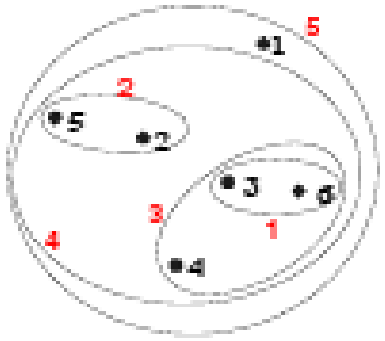
Farklı Uzaklık Yöntemlerinin Etkisi



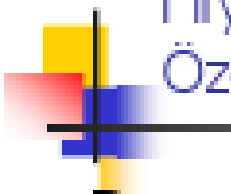
MIN

MAX





Ortalama

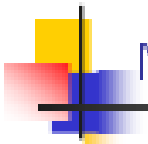


Hiyerarşik Demetleme Yöntemlerinin Özellikleri

- Demetleme kriteri yok
- Demet sayılarının belirlenmesine gerek yok
- Aykırılıklardan ve hatalı verilerden etkilenir
- Farklı boyuttaki demetleri oluşturmak problemli olabilir

Yoğunluk Tabanlı Yöntemler

- Demetleme nesnelerin yoğunluğuna göre yapılır.
- Başlıca özellikleri:
  - Rasgele şekillerde demetler üretilebilir.
  - Aykırı nesnelerden etkilenmez.
  - Algoritmanın son bulması için yoğunluk parametresinin verilmesi gerekir.
- Başlıca yoğunluk tabanlı yöntemler:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)



Model Tabanlı Demetleme Yöntemleri

- Veri kümesi için öngörülen matematiksel model en uygun hale getiriliyor.
- Verinin genel olarak belli olasılık dağılımlarının karışımından geldiği kabul edilir.
- Model tabanlı demetleme yöntemi
  - Modelin yapısının belirlenmesi
  - Modelin parametrelerinin belirlenmesi
- Örnek *EM (Expectation Maximization)* Algoritması