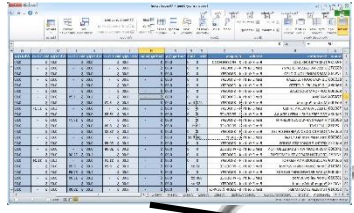


VERİ MADENCİLİĞİ

Fırat İsmailoğlu, PhD

Veri Ön İşleme



veri toplama

**veri
ön işleme**

**veri madenciliği
algoritma uygulaması**

bilgi

Bugün veri ön işleme çalışacağız.

Veri Matrisi (Veri Seti)

Veri hangi formatta verilirse verilsin (ses, video, resim yada excel dosyası) veriyi ‘veri matrisine’ çeviririz. Örneğin bir kanser verisininde veri matrisi şu formda olabilir:

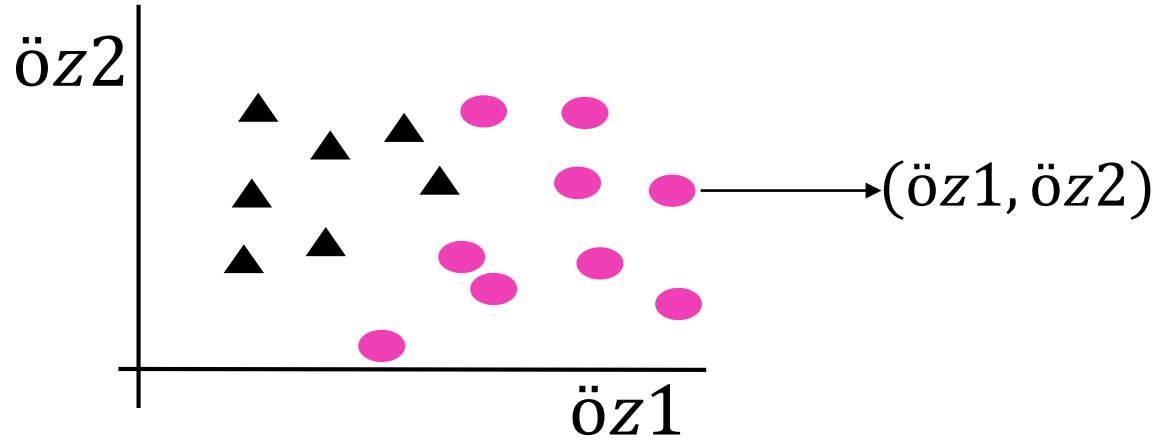
	Hasta Adı Soyadı	Yaş	Kilo	Boy	Sigara Alışkanlığı	Ailede Kanserli Kişi Varlığı	Kanser
objeler (kayıtlar) (gözlemler) (örnekler)	Hasta 1	45	90	178	1	1		1
	Hasta 2	26	56	165	1	0		0
	...							
	Hasta n	78	68	163	1	0		1
	özellikler (features)							sınıf (class)

- Her bir satır bir özellikler serisi ile tanımlanır/nitelenir.
- Özelliklerin ayırt edici olması gerekir. Bütün satırlar (objeler) için aynı olan özellik, özellik değildir.

Özellikler (Features / Attributes)

Özellikler objeleri karakterize ederler, diğer objelerden ayırmaya yardımcı olurlar.

Özellikler sayesinde bir objeyi bir vektör olarak gösterebiliriz, böylece öklid uzayında (kartezyen uzayda) nesneleri gösterebiliriz.



Özellik bir çok farklı şey olabilir: renk, ölçüm (sıcaklık gibi), boolyan değer (var-yok), sokak numarası...

Bu özellik tiplerini ayrı ayrı incelemek gerekir. Çünkü örneğin yaş ve kimlik numarası bir tam sayıdır; fakat yaşları küçükten büyüğe (yada büyüktен küçüğe) anlamlı iken, kimlik numaralarını sıralamak anlamlı değildir.

Yada yaşların ortalamasını alabiliriz, fakat kimlik numaralarının ortalamasını alamayız. Kimlik numaraları üzerinde yapabileceğimiz tek işlem numaralar aynı mı değil mi diye test etmek olabilir. Fakat yaşlarla çok daha fazla işlem yapabiliriz: maksimum, minimum, ortalama..

Bu yüzden veri setimizdeki özelliklerin (kolonların) hangi tipte olduğunu bilmek önemlidir. Böylece bu özelliklerle neler yapabileceğimizi bilebiliriz.

Özellik Tipleri

Özellikleri birbirinden izin verdiği işlem türüne göre ayırırız. Bu işlem türleri 4 tanedir:

1. Farklılık ($=$, \neq)
2. Sıralama ($<$, $>$, \leq , \geq)
3. Toplama – Çıkarma ($+$, $-$)
4. Çarpma – Bölme (\times , \div)



Özellik Tipleri

Kategorik

Sembolik (nominal)

$=, \neq$

Sıralı (ordinal)

$=, \neq$

$<, >, \leq, \geq$

Sayısal

Aralık (interval)

$=, \neq$

$<, >, \leq, \geq$

$+, -$

Bölüm (ratio)

$=, \neq$

$<, >, \leq, \geq$

$+, -$

\times, \div

Burada soldan sağa doğru gittikçe özellik tiplerinde yapabileceğimiz işlemler artar. Buna göre *bölüm* en değerli özellik tipidir, daha sonra *aralık* gelir, daha sonra *sıralı* ve en son *sembolik* gelir.



Sembolik Özellik: Sembolik özellikte değerler yalnızca isimlerdir (sembollerdir) (char tipi düşünebiliriz). Ve bu isimler yalnızca değerler aynı mı değil mi diye anlamamızı sağlarlar. Burada değerleri sıralayamayız, bölemeyiz, çarpamayız...

ör. Kimlik numarası, posta kodu, göz rengi, cinsiyet, hasta şikayeti..

Sıralı Özellik: Sıralı özellik de sembolik özellik gibi isimlerden oluşur; fakat sembolik özellikten farklı olarak burada değerleri sıralayabiliriz.

ör.notlar AA, BA, BB, CB,; fakir, orta sınıf, zengin; en iyi; çok mutsuz, mutsuz, yani, mutlu, çok mutlu.

Aralık Özellik: Sayısal bir değerdir. Eşitlik, sıralama işlemlerine ilaveten burada toplama ve çıkarma işlemleri de yapabiliriz. Fakat aralık özellik tipinde olan değerler için bölme , çarpma yapamayız anlamsız olur. Çünkü değerlerin birbirine oranı anlamsızdır.

ör. Takvim yılı: 900, 1800.. 1800 yılı 900 yılının iki katıdır gibi bir ifade anlamsızdır. Fakat $1800-900=900$, bu iki yılın arasında 900 yıl vardır anlamlı bir ifadedir.

ör. Celcius olarak ölçülmüş hava sıcaklığı. Berlin 20°C, Adana 40°C ölçülmüş olsun. Burada Adana, Berlin'den iki kat daha sıcaktır diyemeyiz; Adana Berlin'den 20°C daha sıcaktır diyebiliriz.



Bölüm Özellik: Sayisal bir degerdir. Eşitlik, sıralama, toplama ve çıkarma işlemlerine ilaveten bu tipteki degerlerde bölme ve çarpma da yapabiliriz. Degerlerin birbirine oranı anlamlidir.

ör. Ağırlık ölçümü: 2 kilo elma bir kilo elmadan 2 kat ağırdır. Uzunluk, miktarlar...

Not: Aralık özellik ile bölüm özelliğın temel farkı, bölüm özelliğinde 0 değeri hiçlığı yoklugu ifade eder. 0 kg elma hiç elmadır, yoktur; fakat 0 °C sıcaklık sıcaklığın olmadigi anlamına gelmez, sıcaklık vardır ve 0 °C ölçülmüştür.

Veri Kalitesi

Veri madenciligi uygulamalarında kullanılan veri genellikle veri madenciligi için toplanmamıştır, bu yüzden sistematik değildir, bir çok hata tutarsizlik barındırır.

Veri madenciliğinde çoğunlukla veri toplama aşamasına geri dönemeyiz. O yüzden hali hazırda toplanan verinin hatalarını önce tespit edip, daha sonra bunları düzeltmeye çalışırız.





Veri Temizleme

Bir veri madenciliği görevine (sınıflandırma yada kümeleme örneğin) başlamadan önce çoğu kez veriyi temizlemek gerekir. Böylece daha sonra uygulanacak veri madenciliği algoritması temizlenen veri üzerinde daha kesin, daha doğru sonuçlar verir.

Veri temizleyerek verinin içinde olması muhtemel gürültü ve tutarsızlıklar ile anomalilerden (outlier) kurtulmak hedeflenir.

Gürültü (Noise) ve Tutarsızlıklar (Inconsistency)

Gürültü bir ölçüm yapılırken yapılan hatalardır. Gürültü şu şekilde formüle edilebilir:

$$\text{Gürültü} = |\text{ölçüm} - \text{gerçek değer}|$$

örneğin gerçekte boyu 192 cm olan birinin 189 cm olarak ölçülmesi; yada gerçekte 0.005 milimetre olan bir tahtanın 0.05 olarak ölçülmesi ve bu değerlerin veri olarak kaydedilmesi.

! Gürültüyü tespit etmek güçtür; çünkü çoğu kez ölçtüğümüz şeyin gerçek değerini bilmeyiz (zaten gerçek değerini bilsek ölçmezdik). Su halde gürültüyü veriden uzaklaştırmak zordur.

Tutarsızlık ise yanlış girilen değerdir. Tutarsızlık örnekleri:

isim kolonuna 123 girilmesi.

Uzunluk kolonuna -903 girilmesi.

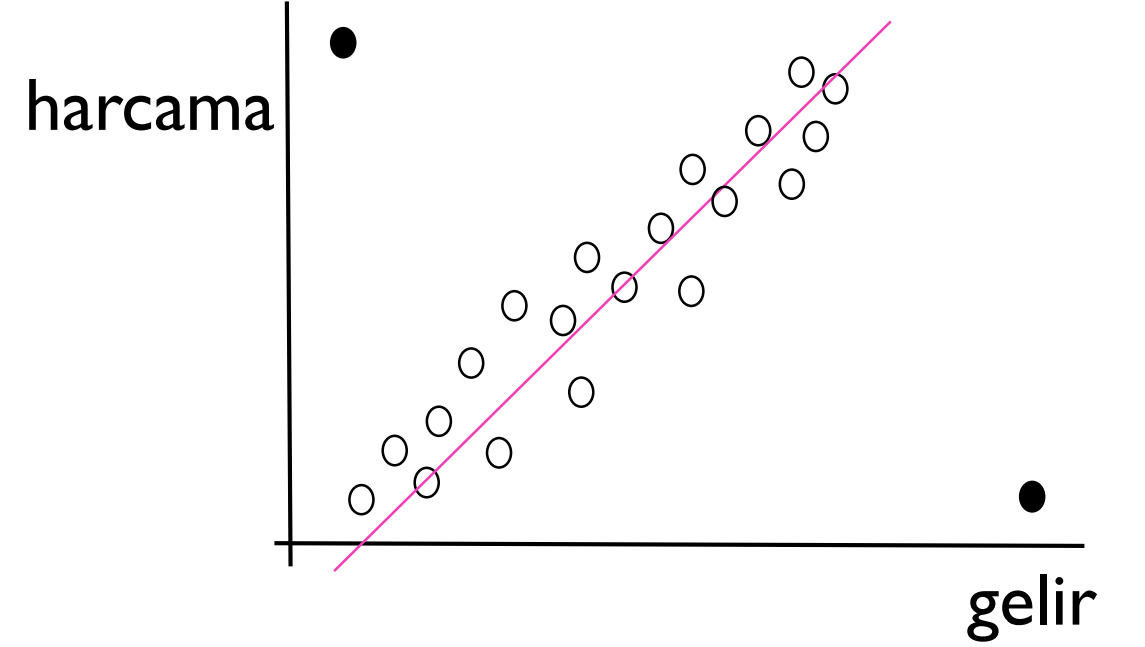
1977 doğumlu birinin yaşı 23 olarak girilmesi.

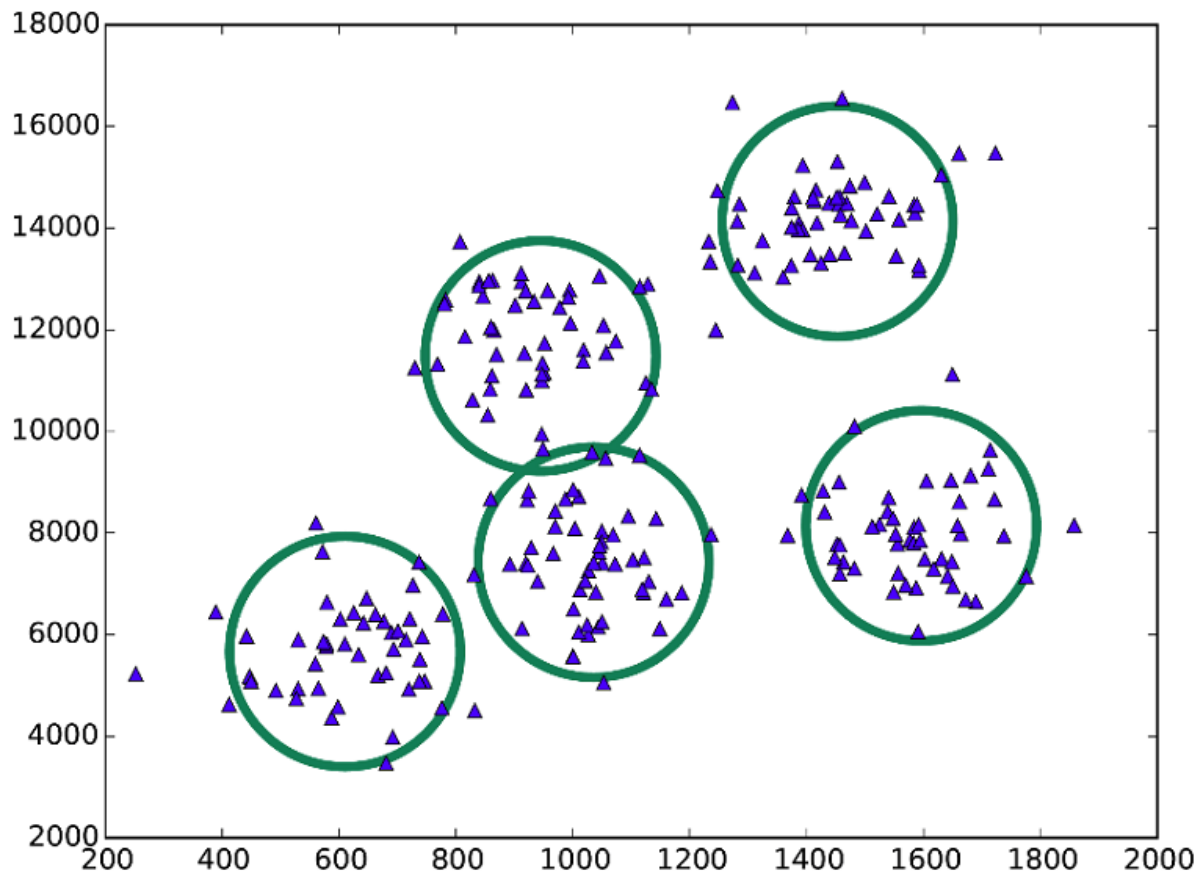
Amerika'daki ev fiyatları veri setinde adres olarak Shanghai girilmesi...

Not: Tutarsızlığın tespit edebilmek için verinin alındığı alanı iyi bilmek gerekir. Örneğin bir finans verisi ise finans bilmek, yada tıbbi bir veri için tıp bilmek..

Anomali (Outlier)

Anomali verinin genel davranışına uymayan objedir.





Yanda halkalarla sinirlari belirtilen kumelerin icine duscmeyen nesneleri (noktalari) anomali olarak duscunebiliriz.

ör. Diyelimki bir siniftaki kisilerin agirliklarini 65,71,63,66 ve 75 kg olarak olctuk. Bu beş kisinin ortalama agirligi 68 dir.

Eger sinifa 92 kg olan bir kisi gelirse siniftaki alti kisinin ortalama agirligi 86,4 olur.

Ortaya cikan yeni ortalama veriyi yanliş yorumlamamiza neden olur. Çünkü agirliklar genelde 65-75 arasindadir. Bu anlamda 92 kg olan kisiyi (anomaliyi) hesaplamamiza dahil etmezsek daha dogru sonuc elde ederiz.

Öte yandan anomaliler verideki tuhaflıkları tespit etmemizde önemlidir. Böyle durumlarda veri setinden çıkarılmaz.

ör.

Tarih	Harcama Türü	Şirket	Miktar
22 May 11.42	Yemek	Nişantaşı Cafe	210 TL
23 May 19.32	Yemek	Albatros	18 TL
...
1 Haz 09.11	İnşaat	Beton A.Ş.	<u>850 TL</u>
1 Haz 09.12	İnşaat	Beton A.Ş.	<u>850 TL</u>
3 Haz 18.19	Kozmetik	Gratis	21 TL
7 Haz 20.21	Yemek	Albatros	26 TL
13 Haz 17.45	Yakıt	Opet	130 TL
14 Haz 18.55	Yemek	Albatros	22 TL
17 Haz 12.24	İnşaat	Beton A.Ş.	<u>850 TL</u>
17 Haz 12.25	İnşaat	Beton A.Ş.	<u>850 TL</u>

Nişantaşı?

Hep aynı miktar?



Sonuç olarak gürültü yada tutarsız degerler kabul edilebilir (yasal) degildir; tespit edildiğinde derhal veri setinden çıkarmak gerekir. Anomali ise aykırı objelerdir, yasaldir; veri madenciligi amacina göre veri setinden çıkarilir yada çıkarılmaz.

Kayıp Değerler (Missing Values)

Veri toplanırken bazı nedenlerden dolayı bazı değerler girilmemiş olabilir. Örneğin bir anket yaparken bazı katılımcılar kilosunu yada yaşını vermek istememiş olabilir. Yada veriyi toplayan sensor kısa süreliğine bozulmuş, bazı verileri toplayamamış olabilir.

Ad- Soyad	Yaş	Kilo	Medeni Hal	Aylık Gelir	Kozmetik Harcaması	Market Harcaması
Pinar Aylin		61	Evli	28300	3200	6000
Harika Avcı	48	59	Bekar	34000	11000	5300
Merve İldeniz			Bekar	32000		
Aysun Kayacı		49		88700	4400	9900



Kayıp değerler sorunu çözmek için iki yöntem vardır. 1) Objeleri (satırları) yada özellikleri (kolonları) silme 2) Kayıp değerleri tahmin etme.

1. Satırları yada Kolonları Silme

Kaybın çok olduğu satır (obje kayıt) silinebilir ve/veya kaybın çok olduğu kolon (özellik) silenebilir. Bir önceki örnekte Merve İldeniz'e denk gelen satır silinebilir; yada yaş kolonu silinebilir.

Not: İçinde kayıp değerler olan satır yada kolonları silmek bilgi kaybına yol acar.

2. Kayıp Değerleri Tahmin Etme

Veri matrisindeki diğer değerler kullanılarak matristeki kayıp değerler tahmin edilebilir. Eğer özellik (kolon) sayısal tipte ise, o özelliğin veri matrisindeki ortalaması kayıp değerlere yazılabilir. Eğer özellik kategorik tipte ise o özellikteki en yaygın kategori kayıp değerlere yazılabilir.

Örneğin bir önceki örnekte Merve İldeniz'in kilosu: $(61+59+49)/3 \approx 56$ olarak tahmin edilebilir.

Aysun Kayacı'nın medeni hali ise bekar olarak tahmin edilebilir; çünkü medeni hal kolonunda bekar sayısı evli sayısından daha fazladır.



Boyut Azaltma (Dimensionality Reduction)

Veri setlerinde çok fazla sayıda özellik (kolon) olabilir. Örneğin her bir satır dokumana her bir kolon bir kelimeye denk gelen aşağıdaki veri matrisini düşünelim.

	ben	sen	biz	yarın	bugün	yemek	gitmek	...	hiç
Dök. 1	23	2	1	9	2	4	2	...	1
Dök. 2	12	5	1	0	0	11	12	...	6
Dök. 3	8	12	0	1	2	3	5	...	2
Dök. 4	21	16	2	3	7	8	0	...	4

Böyle bir veri matrisinde binlerce onbinlerce kolon olabilir (binlerce onbinlerce farklı kelime olabilir).

Yada satırlar resimlere, kolonlar piksellere denk gelen bir veri matrisinde milyonlarca kolon (piksel) olabilir.

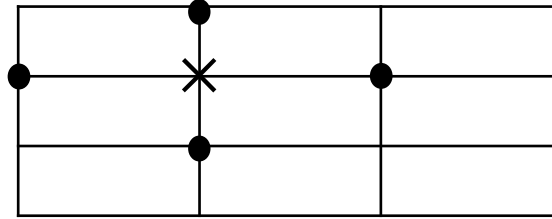


Çok sayıda özellik varsa çok fazla kolon çok fazla boyut vardır, böylece ‘boyutun laneti’ (the curse of dimensionality) ortaya çıkar:

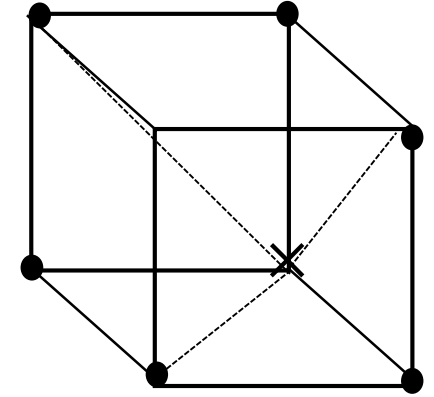
- Objeler birbirinden uzaklaşır; benzerlik, yakınlık kavramları yok olmaya başlar.
- Objeleri görüntülemek zorlaşır; gerçek dünya 3 boyutlu olduğundan çok boyutlu veriyi anlayamayız; gözümüzde canlandıramayız.
- Veri madenciliği algoritması daha düşük performans gösterir.



2 tane komşusu var
bir boyutlu



4 tane
komşusu var
iki boyutlu



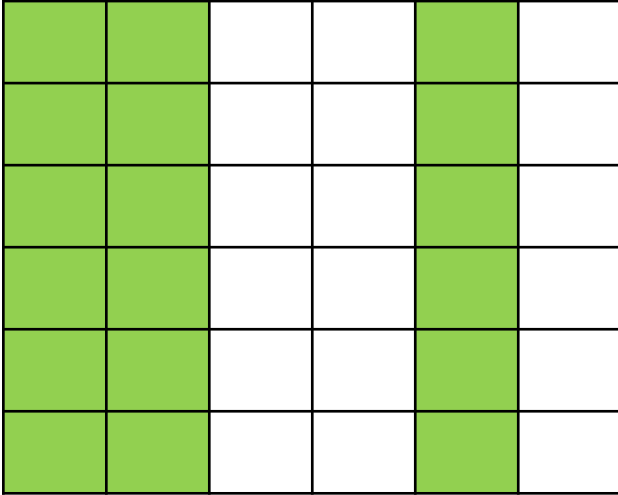
6 tane
komşusu var
üç boyutlu

Boyut Azaltma

Seleksiyon

(feature subset selection)

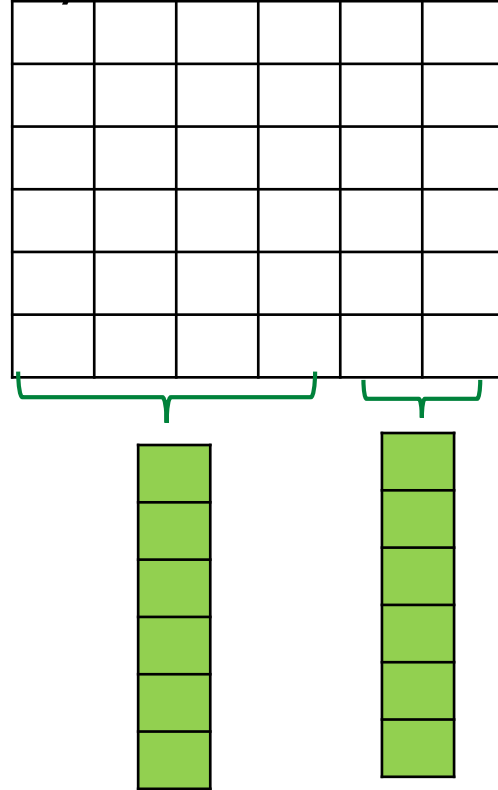
Var olan özelliklerin içinden secim yaparız.



Kombinasyon

(feature extraction)

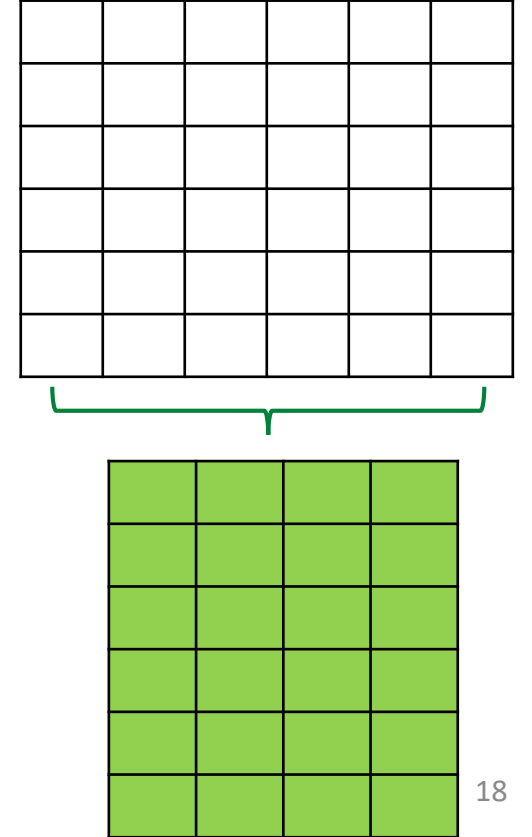
Var olan özellikleri kombine ederek yeni özellikler elde ederiz.



Transformasyon

(feature transformation)

Özellikleri dönüştürerek daha düşük boyutlu yeni bir uzaya gideriz.



Seleksiyon (Feature Subset Selection)

Seleksiyon ikiye ayrilir: filtreleme ve sarmalama.

I. Filtreleme (Filtering)

Filtreleme, yapacagimiz veri madenciligi görevinden (siniflandirma, kümeleme) tamamen bağımsiz olarak her bir ozelligin ayri ayri kalip kalmayacagina karar vermektir.

Filtrelemede, belirleyeceğimiz bir kritere gore her bir ozellik bir skor alır. Daha sonra özellikler bu skorlara gore siralanir. En yüksek skor alan özellikler kalir, diğerleri gider.

Özelliklere neye göre skor (puan) veririz?

- Varyasyon
- Information Gain
- Korelasyon
-



Bir kolonda eger tüm deęerler birbirinin ayni ise varyasyon 0 ıkar. Eger varyasyon filtreleme kullanirken kriterimiz ise bu kolonu semeyiz. Zaten eger tum deęerler ayni ise bu ozellik objeler iin ayirt edici bir ozellik deęildir.



Filtreleme'nin en buyuk dezavantaji zellikleri teker teker ele aldigindan zellikler korelasyonu/iliişkiyi dikkate almaz. Yani belki bir ozellik tek basina nemli deęildir ama onu tamamlayan bir kac ozellik ile beraber anlamlı hale gelir.

2. Sarmalama (Wrapping)

Sarmalamada zellikleri tek tek deęil alt kmeler halinde inceleriz (yani birbirine sararız). Oluşan her bir alt kme (grup) iin bir skor (puan) veririz.

Bir zellik grubunu puanlarken bu zelliklerden oluşan veri zerinde siniflandirma yapariz. Bu siniflandirmanin verimlilięine gore bu gruba bir puan veririz.

Sonu olarak en yksek puani alan gruptaki zellikler seilir.

Diyelimki A, B, C diye üç tane özelliğimiz olsun. Bunlardan bir yada bir kacini sarlamama yöntemi ile elemek istiyoruz. Bu üç özellik ile olusturabileceğimiz toplam alt küme (grup) sayısı $2^3 = 8$ dir. Bu gruplari kullanarak elde ettiğimiz başarı oranlari aşağıdaki gibi olsun:

Özellik Alt Kumesi	Başarı Oranı
A, B, C	%98
A, B	%98
A, C	%87
B, C	%89
A	%77
B	%68
C	%70
$\{ \}$	%60

En yüksek başarı oranı A ve B özellikleri varken elde edildiğinden A, B 'yi seçeriz.



Sarmalama yöntemindeki temel problem test edilmesi gereken alt küme sayisinin çok fazla olmasidir. n tane özellik için 2^n tane muhtemel özellik alt kümesi vardır. Makul sayıda alt kümeyi test etmek için bir arama algoritmasi (best first, deep first...) kullanılır.

Kombinasyon (Feature Extraction)

Veri matrisindeki bazı özellikleri kombine ederek yeni bir özellik inşa edebiliriz.

ör. Diyelimki elimizde fotoğraflardan oluşan bir veri matrisi var; oyleki satirlar fotoğraflara kolonlar ise piksellere karsilik geliyor. Ve diyelimki amacimiz fotoğraflarda insan yüzü olup olmadigini tespit etmek. Burada ornegin göze denk gelen pikselleri kombine ederek yeni bir kolon oluşturabiliriz. Bu yeni kolonu eğer fotoğrafta göz varsa 1 yoksa 0 olarak kodlayabiliriz.

ör.

Fizik	Kimya	Biyoloji	Cebir	Geometri
78	70	90	80	91
17	45	63	47	50
21	28	30	40	28
83	77	82	79	86

Bu üç kolon kombine edilerek
tek bir Fen Bilimleri
oluşturulabilir

Bu iki kolondan Matematik
kolonu oluşturulabilir.



Transformasyon

Bir diğer boyut azaltma yöntemi transformasyondur. Transformasyon ile çok boyutlu öklid uzayında yaşayan objeler daha düşük boyutlu öklid uzayına gönderirler.

Veri madenciliğinde kullanılan önemli transformasyon yöntemleri şunlardır:

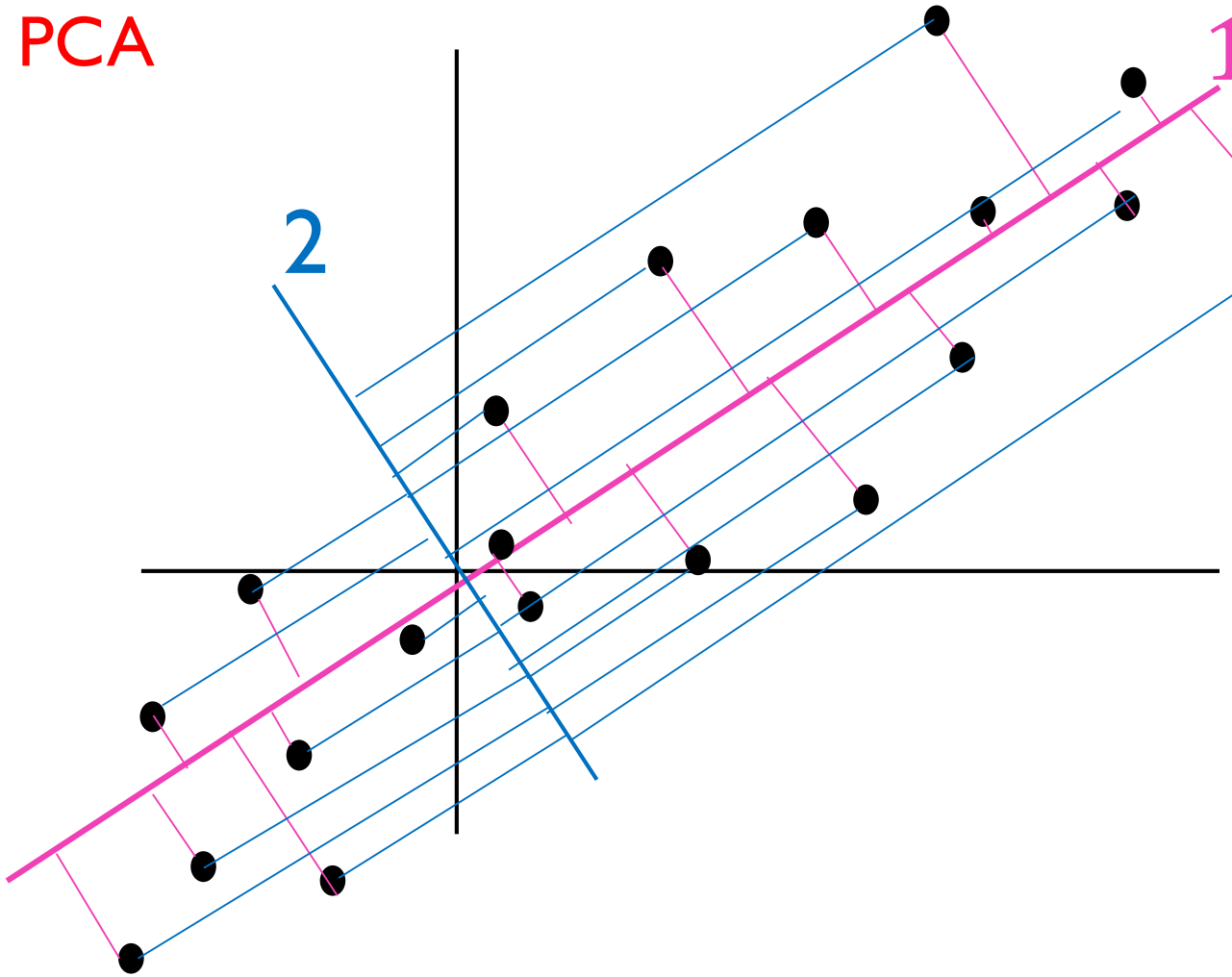
- Principal Component Analysis (PCA),
- Linear Discriminant Analysis (LDA)
- Multidimensional Scaling

Biz bunların içindeki en meşhur olan PCA'yi inceleyeceğiz.

Principal Component Analysis (PCA) (Temel Bileşenler Analizi)

Diyelimki objelerimiz iki özellik tarafından ifade edilsin, yani iki boyutlu öklid uzayında yaşasın. Amacımız bu objeleri tek boyutlu hale getirmek olsun, yani bir doğru üzerinde ifade etmek.





Bu noktaları (objeleri) bir doğru üzerinde göstermek istiyoruz. Böylece iki boyutta gösterilmiş olan objeler tek boyutta gösterilmiş olacak.

Transformasyon yaparken kayıpları minimize etmek istiyoruz. Yani noktalar olabildiğince en yakın noktaya transform olsun istiyoruz.

Eğer 1. doğru üzerine transformasyon yaparsak kayıplar (yani noktaların doğru üzerine olan uzaklıkları toplamı) minimum olur.

PCA

Ayrıca 1. doğru üzerine transformasyon yaptığımızda verinin içindeki varyasyon (çeşitlilik) maksimize edilir. Böylece noktaların birbirlerine uzaklıkları korunmuş olur.



Noktalar eğer 1.doğru üzerine transform edilirse



Noktalar 2.doğru üzerine transform edilirse

Soru: Peki bu arzu ettiğimiz doğruları (yönleri) nasıl bulacağız?

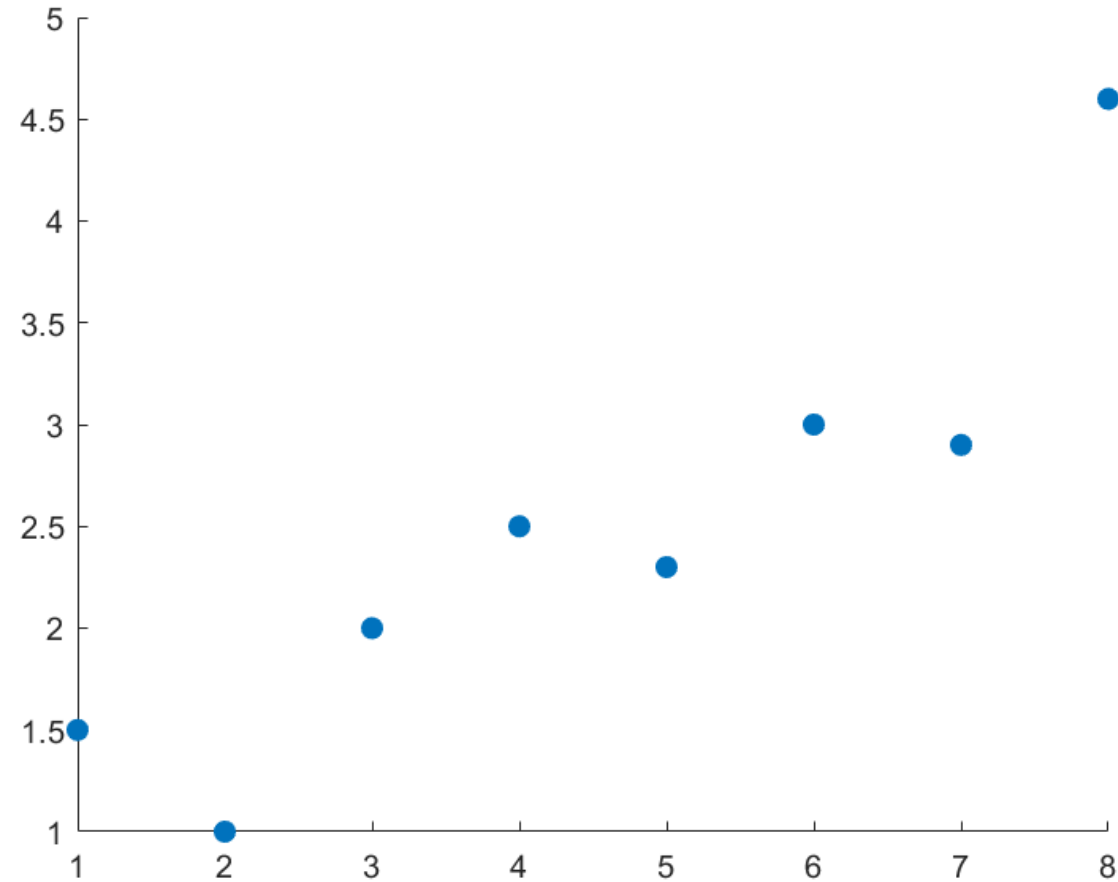
Cevap: Bu doğrular veri matrisinin kovaryans matrisinin en yüksek öz değere (eigen value) sahip öz vektörleridir (eigen vectors).

Demekki önce kovaryans matrisi hesaplayacağız, daha sonra bu matrisin öz değer ve öz vektörlerini bulacağız.

ör. Veri matrisi A=

1	1.5
2	1
3	2
4	2.5
5	2.3
6	3
7	2.9
8	4.6

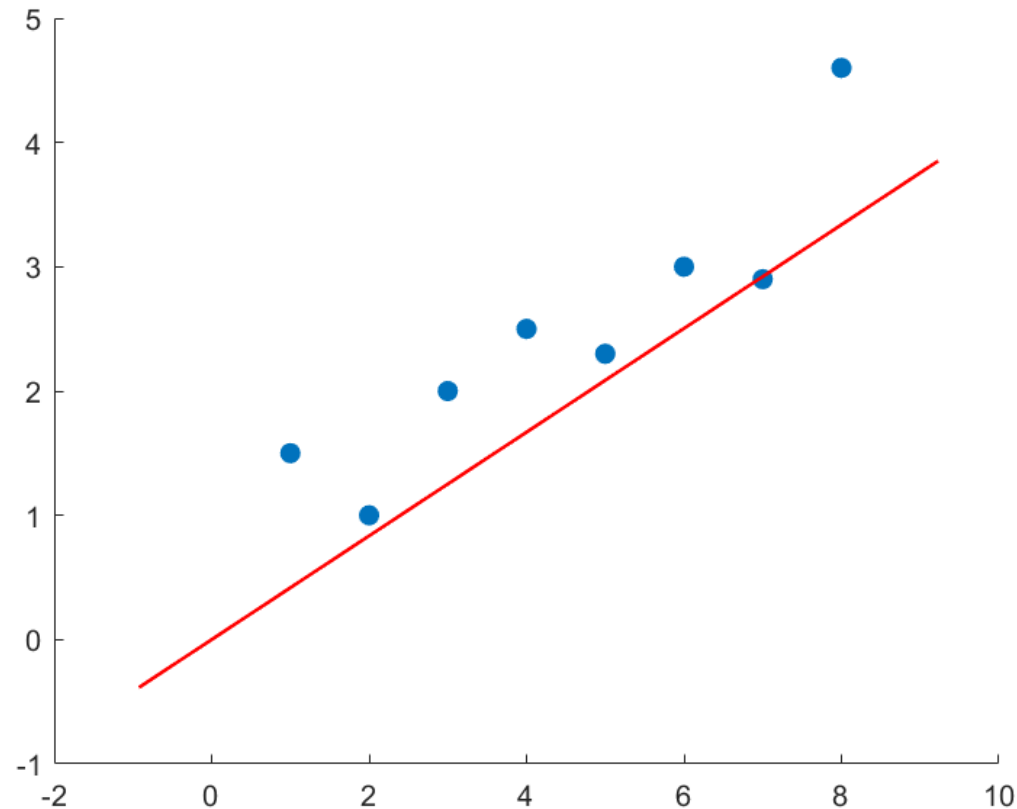
olsun. Bu veriyi görüntüleyelim:



ör. A matrisinin kovaryans matrisi: $\text{cov}(A) = \begin{bmatrix} 6 & 2.42 \\ 2.42 & 1.19 \end{bmatrix}$

Kovaryans matrisin birinci öz vektörü $\begin{bmatrix} 0.3 \\ -0.9 \end{bmatrix}$ ikinci öz vektörü: $\begin{bmatrix} -0.9 \\ -0.3 \end{bmatrix}$

Birinci öz vektörü çizelim:



PCA'ile Transformasyon

$$\begin{bmatrix} 1 & 1.5 \\ 2 & 1 \\ 3 & 2 \\ 4 & 2.5 \\ 5 & 2.3 \\ 6 & 3 \\ 7 & 2.9 \\ 8 & 4.6 \end{bmatrix}_{8 \times 2} \times \begin{bmatrix} 0.3 \\ -0.9 \end{bmatrix}_{2 \times 1} = \begin{bmatrix} -1.05 \\ -0.3 \\ -0.9 \\ -1.05 \\ -0.57 \\ -0.9 \\ -0.51 \\ -1.74 \end{bmatrix}$$