Makine Öğrenmesi

Veri Ön İşleme

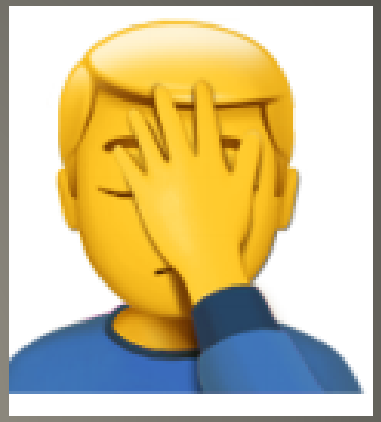
Doç. Dr. İlhan AYDIN

Veri ön işleme

- Alanı anla, ön bilgi, ve amaçlar
- Veri seçimi, temizleme, birleştirme, ön işleme
- Modeli eğitme
- Modeli değerlendirme ve analiz
- Model dağıtımı ve çıkarılan bilgi

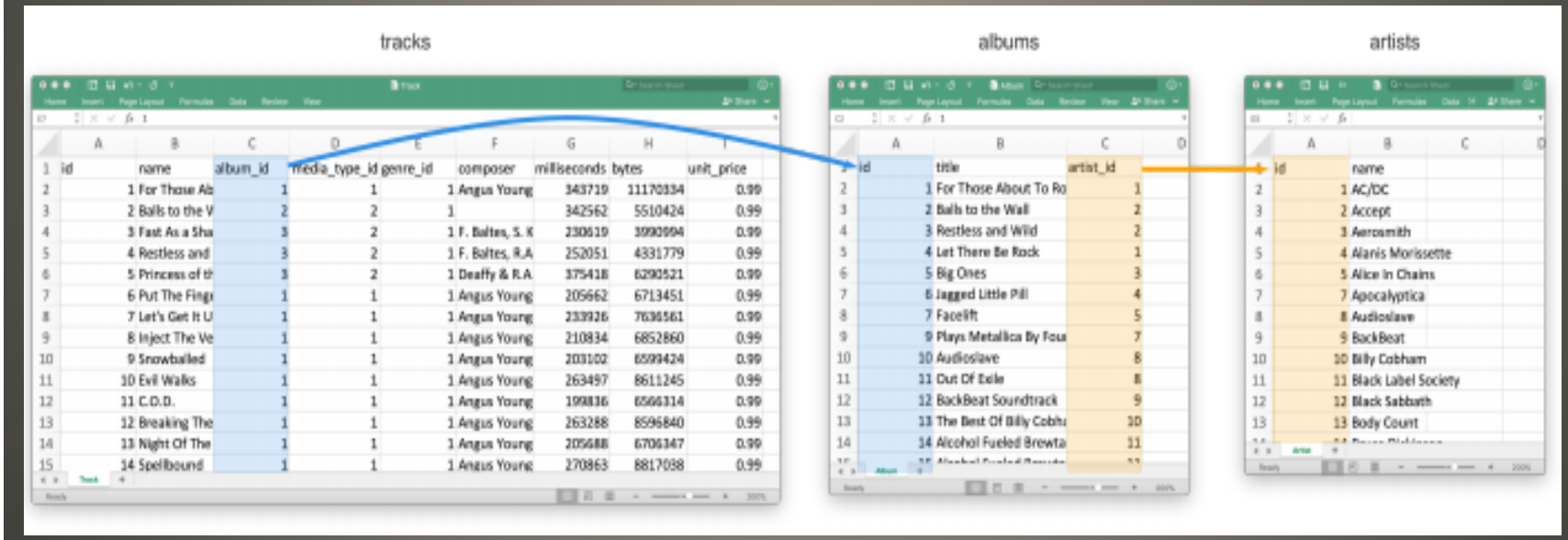
Veri Ön İşleme Pahallıdır

- Veriler nadiren "temiz" olur. Gerçek veriler dağınıktır
- Zamanın yaklaşık %50-80'i veri tartışmasına harcanıyor
  - Kime sorduğunuza, uygulamaya, veri kaynağına vb. bağlıdır...
  - Ve %80'i hafife alınabilir



Doğru özelliklere sahip iyi verilere sahip olmak başarı için kesinlikle çok önemlidir

- Veriler farklı dosyalar arasında bölünebilir
- Verileri tek bir tabloda birleştirmek için bir anahtara dayalı birleştirme yapmayı gerektirir

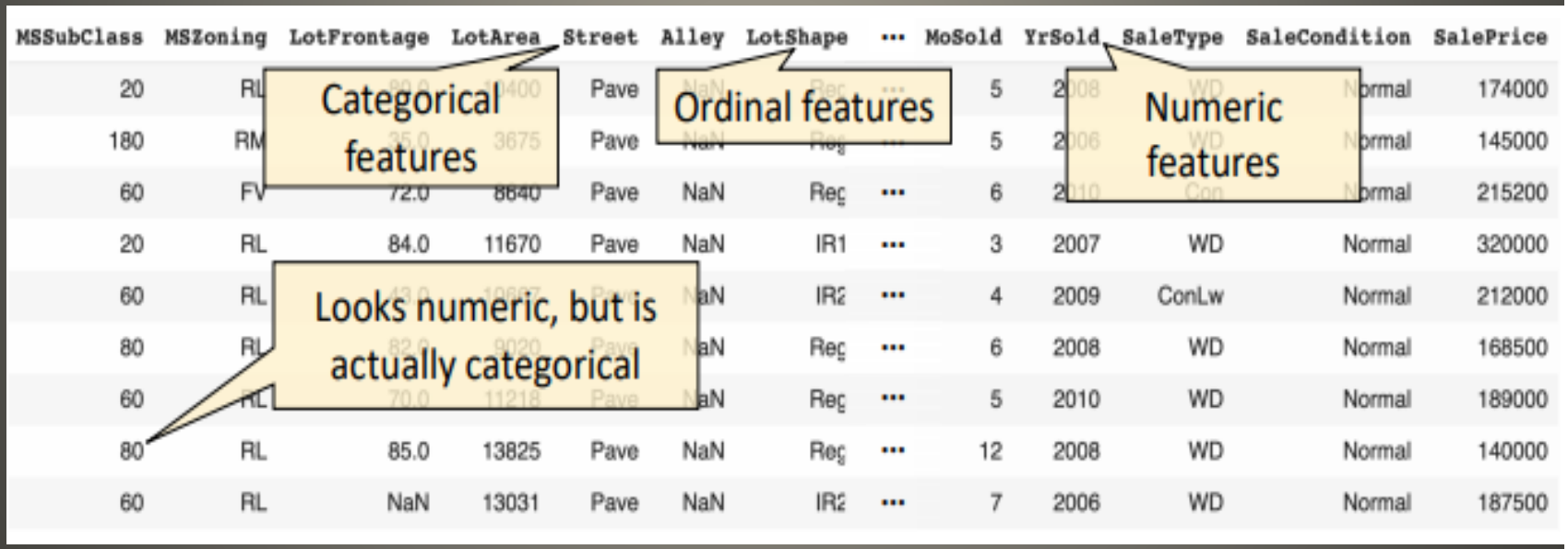


Veri Birleştirme

Birleştirmede Oluşan Problemler

- Tutarsız veri
  - Çakışan etiketlere sahip aynı örnek anahtarı
  - Kopya veriler
- Birleştirilmiş tablo bellek için çok büyük olabilir
  - SGD veya mini gruplar kullanarak verileri aşamalı olarak yükleyin ve birleştirin
  - Çevrimiçi öğrenme tekniklerini kullanın
- Kodlama konuları
  - Tutarsız veri biçimleri veya terminoloji
  - Hücre açıklamalarında veya yardımcı dosyalarda belirtilen temel hususlar

- Çoğu veri seti birden fazla özellik türü içerir
- Türler ve olası değerler açık olmayabilir
- Bir veri sözlüğüne danışmak çok önemlidir



MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	...	MoSold	YrSold	SaleType	SaleCondition	SalePrice
20	RL			Pave			...	5				174000
180	RM		3675	Pave			...	5				145000
60	FV	72.0	8640	Pave	NaN	Reg	...	6				215200
20	RL	84.0	11670	Pave	NaN	IR1	...	3	2007	WD	Normal	320000
60	RL				NaN	IR2	...	4	2009	ConLw	Normal	212000
80	RL				NaN	Reg	...	6	2008	WD	Normal	168500
60	RL				NaN	Reg	...	5	2010	WD	Normal	189000
80	RL	85.0	13825	Pave	NaN	Reg	...	12	2008	WD	Normal	140000
60	RL	NaN	13031	Pave	NaN	IR2	...	7	2006	WD	Normal	187500

Karışık özellik tipleri

Kodlanmış özellikler

- Kategorik özellikleri kodlamak gerekir
  - Tek etkin kodlamayı kullanın
- Sıralı özellikleri kodlama
  - Sırayı koruyarak bir sayıya dönüştürme (ör. [düşük, orta, yüksek] yerine [1, 2, 3])
  - Kodlama, görecelili farklılıkları yakalamayabilir

HouseStyle	FullBath	RoofMatl	BsmtCond	KitchenQual
1Story	2	CompShg	TA	TA
SLvl	1	CompShg	TA	TA
2Story	2	CompShg	TA	Gd
1Story	2	CompShg	Gd	Ex
2Story	2	CompShg	TA	Gd
SLvl	1	WdShngl	TA	TA
2Story	2	CompShg	TA	Gd
SLvl	1	CompShg	TA	TA
2Story	2	CompShg	TA	TA
2Story	2	CompShg	TA	Gd

HouseStyle	FullBath	RoofMatl	BsmtCond	KitchenQual
1Story	2	CompShg	3	3
SLvl	1	CompShg	3	3
2Story	2	CompShg	3	4
1Story	2	CompShg	4	5
2Story	2	CompShg	3	4
SLvl	1	WdShngl	3	3
2Story	2	CompShg	3	4
SLvl	1	CompShg	3	3
2Story	2	CompShg	3	3
2Story	2	CompShg	3	4

Veri Sorunları

- Eksik özellik değerleri
- Eksik öznitelik değerleri olan veri satırlarını (örnekleri) kullanma, onları sil (Oldukça kolay bir çözüm).
- Eksik öznitelik değerlerini elle doldur (Veri büyüdükçe ve eksik olan verinin önemine göre zaman alıcı ve etkin olmayan bir yönteme dönüşebilir).
- Eksik öznitelik değerleri için global bir değişken kullan (Null, bilinmiyor).
- Eksik öznitelik değerlerini ortalama değer ile doldur.
- Aynı sınıfa ait kayıtların öznitelik değerlerinin ortalaması ile doldur.
- Olasılığı en fazla olan öznitelik değeriyle doldur.
- Regresyon yöntemi ile sayısal eksik değerleri tahmin et ve doldur

```
import pandas as pd
import numpy as np

sozluk = {'İsim':pd.Series(['Ada','Cem','Sibel','Ahmet','Mehmet','Ali','Veli',
        'Ayşe','Hüseyin','Necmi','Nalan','Namık']),
        'Meslek':pd.Series(['işçi','işçi','memur','serbest','serbest',None,None,
        'sigortacı','işsiz',None,None,'memur']),

'Tarih':pd.Series(['11.11.2010','11.11.2010','11.11.2010','18.11.2011','18.11.201
1',None,None,
        None,'11.11.2010',None,'18.11.2011','18.11.2011']),
        'Yaş':pd.Series([21, 24, 25, 44, 31, 27, 35, 33, 42, 29, 41, 43]),
        'ÇocukSayısı':pd.Series([None, None, None, None, None, 1, 2, 0, None,
None, None, None]),
        'Puan':pd.Series([89, 87, 77, 55, 70, 79, 73, 79, 54, 92, 61, 69])}
df = pd.DataFrame(sozluk)
print(df)
```

Python pandas Kütüphanesi ile eksik veri Doldurma

df - DataFrame

Index	İsim	Meslek	Tarih	Yaş	ocukSayı	Puan
0	Ada	işçi	11.11.2010	21	nan	89
1	Cem	işçi	11.11.2010	24	nan	87
2	Sibel	memur	11.11.2010	25	nan	77
3	Ahmet	serbest	18.11.2011	44	nan	55
4	Mehmet	serbest	18.11.2011	31	nan	70
5	Ali	None	None	27	1	79
6	Veli	None	None	35	2	73
7	Ayşe	sigortacı	None	33	0	79
8	Hüseyin	işsiz	11.11.2010	42	nan	54
9	Necmi	None	None	29	nan	92
10	Nalan	None	18.11.2011	41	nan	61
11	Namık	memur	18.11.2011	43	nan	69

Python pandas Kütüphanesi ile eksik veri Doldurma

Toplam kaç hücrede eksik değer (NaN ya da None) var?
df.isnull().sum().sum()

17 tane ekisk değer var

Özniteliklerin değer almadığı kaç satır var?
Aşağıdaki kod satırını çalıştıralım ve görelim.
df.isnull().sum()

```
İsim           0
Meslek         4
Tarih          4
Yaş            0
ÇocukSayısı    9
Puan           0
dtype: int64
```

Python pandas Kütüphanesi ile eksik veri Doldurma

Eksik değerleri sayısal olarak görmek basit olsa da, eksik değerlerin satır bazında yüzdesini görmek, “bundan sonraki adıma karar vermek adına” daha sağlıklıdır.

```
def eksik_deger_tablosu(df):
    eksik_deger = df.isnull().sum()
    eksik_deger_yuzde = 100 * df.isnull().sum()/len(df)
    eksik_deger_tablo = pd.concat([eksik_deger, eksik_deger_yuzde], axis=1)
    eksik_deger_tablo_son = eksik_deger_tablo.rename(
    columns = {0 : 'Eksik Değerler', 1 : '% Değeri'})
    return eksik_deger_tablo_son
sonuc=eksik_deger_tablosu(df)
```

sonuc - DataFrame

Index	sik Değer	% Değeri
İsim	0	0
Meslek	4	33.3333
Tarih	4	33.3333
Yaş	0	0
ÇocukSayısı	9	75
Puan	0	0

Python pandas Kütüphanesi ile eksik veri Doldurma

Bu tabloda özellikle %75 eksik değer oranıyla (12 satırın 9'unda bu değer yok) “ÇocukSayısı” özniteliği göze çarpmaktadır.

*Bu kadar eksik değerin olduğu bir öznitelik büyük ihtimalle işe yaramayacaktır. Bu özniteliği veri kümesinden kaldırmak mantıklı olabilir. Aslında şöyle bir **strateji** de izleyebiliriz: Belirli bir eşik değerin üzerinde, örneğin %70, eksik değer olan öznitelikleri veri kümesinden sil…*

```
tr = len(df) * .3
df.dropna(thresh = tr, axis = 1, inplace
= True)
df
```

sonuc - DataFrame

Index	sik Değer	% Değeri
İsim	0	0
Meslek	4	33.3333
Tarih	4	33.3333
Yaş	0	0
ÇocukSayısı	9	75
Puan	0	0

	İsim	Meslek	Tarih	Yaş	Puan
0	Ada	işçi	11.11.2010	21	89
1	Cem	işçi	11.11.2010	24	87
2	Sibel	memur	11.11.2010	25	77
3	Ahmet	serbest	18.11.2011	44	55
4	Mehmet	serbest	18.11.2011	31	70
5	Ali	None	None	27	79
6	Veli	None	None	35	73
7	Ayşe	sigortacı	None	33	79
8	Hüseyin	işsiz	11.11.2010	42	54
9	Necmi	None	None	29	92
10	Nalan	None	18.11.2011	41	61
11	Namık	memur	18.11.2011	43	69

Python pandas Kütüphanesi ile eksik değeri yüksek sütunları silmek

Meslek ve *Tarih* öznitelikleri için farklı eksik değer doldurma stratejileri izlenebilir.
Örneğin, *Meslek* özniteliği olmayan kayıtlara “Diğer” değeri atanabilir. *Tarih* özniteliği ek

```
#Meslek özniteliğindeki Null değerleri 'Diğer
' değeri ile doldur
df['Meslek'] = df['Meslek'].fillna('Diğer')

#Tarih özniteliğindeki Null değerleri Tarih b
enzersiz değerlerden ilki ile doldur
print(df['Tarih'].unique()[0])
df['Tarih'] = df['Tarih'].fillna(df['Tarih'].
unique()[0])

df
```

	İsim	Meslek	Tarih	Yaş	Puan
0	Ada	işçi	11.11.2010	21	89
1	Cem	işçi	11.11.2010	24	87
2	Sibel	memur	11.11.2010	25	77
3	Ahmet	serbest	18.11.2011	44	55
4	Mehmet	serbest	18.11.2011	31	70
5	Ali	Diğer	11.11.2010	27	79
6	Veli	Diğer	11.11.2010	35	73
7	Ayşe	sigortacı	11.11.2010	33	79
8	Hüseyin	işsiz	11.11.2010	42	54
9	Necmi	Diğer	11.11.2010	29	92
10	Nalan	Diğer	18.11.2011	41	61
11	Namık	memur	18.11.2011	43	69

Python pandas Kütüphanesi ile eksik değerleri doldurma

Python pandas Kütüphanesi yeni öznitelik oluşturma

Bazı durumlarda mevcut öznitelikleri kullanarak, “daha fazla işe yarayacağı düşünülen” yeni öznitelikler oluşturulabilir. (1) Örneğin, aşağıdaki kod satırları çalıştırılarak, *Geçti* isimli yeni bir öznitelik oluşturulmuş ve sınavdan 70 (ve) üzerinde not alan kayıtların bu öznitelik değerleri “True” olarak ayarlanmıştır.

```
def basari_durumu(puan):
    return (puan >= 70)

df['Geçti'] = df['Puan'].apply(basari_durumu)
df
```

	İsim	Meslek	Tarih	Yaş	Puan	Geçti
0	Ada	işçi	11.11.2010	21	89	True
1	Cem	işçi	11.11.2010	24	87	True
2	Sibel	memur	11.11.2010	25	77	True
3	Ahmet	serbest	18.11.2011	44	55	False
4	Mehmet	serbest	18.11.2011	31	70	True
5	Ali	Diğer	11.11.2010	27	79	True
6	Veli	Diğer	11.11.2010	35	73	True
7	Ayşe	sigortacı	11.11.2010	33	79	True
8	Hüseyin	işsiz	11.11.2010	42	54	False
9	Necmi	Diğer	11.11.2010	29	92	True
10	Nalan	Diğer	18.11.2011	41	61	False
11	Namık	memur	18.11.2011	43	69	False

Veri kümemizdeki *Tarih* özniteliğini kullanarak yıl bilgisini almak ve yeni bir öznitelik olarak eklemek istediğimiz durumda aşağıdaki kod satırlarını çalıştırabiliriz.

```
tarih = pd.to_datetime(df['Tarih'])
df['Yıl'] = tarih.dt.year
df
```

	İsim	Meslek	Tarih	Yaş	Puan	Geçti	Yıl
0	Ada	işçi	11.11.2010	21	89	True	2010
1	Cem	işçi	11.11.2010	24	87	True	2010
2	Sibel	memur	11.11.2010	25	77	True	2010
3	Ahmet	serbest	18.11.2011	44	55	False	2011
4	Mehmet	serbest	18.11.2011	31	70	True	2011
5	Ali	Diğer	11.11.2010	27	79	True	2010
6	Veli	Diğer	11.11.2010	35	73	True	2010
7	Ayşe	sigortacı	11.11.2010	33	79	True	2010
8	Hüseyin	işsiz	11.11.2010	42	54	False	2010
9	Necmi	Diğer	11.11.2010	29	92	True	2010
10	Nalan	Diğer	18.11.2011	41	61	False	2011
11	Namık	memur	18.11.2011	43	69	False	2011

Bilgisayar bilimlerinde kategorik verilerle çalışmak, hesaplama ve bilgisayarın bu değerleri anlaması açısından zorluklar içerir. Özellikle makine öğrenmesi modellerinin doğru çalışabilmesi için kategorik verileri, sayısal karşılıklarına (temsillerine) dönüştürmemiz gerekmektedir. Bunu yapmanın en yaygın iki yolu Sklearn kütüphanesi altında yer alan LabelEncoder yaklaşım olarak görmek mümkündür.

Label Encoder

Elimizdeki verileri direk sayısal temsillerine dönüştürmeye yarar ve kategorik her veriye sayısal bir değer atar. Genelde sadece iki değere sahip özniteliklerde kullanılır. Örneğin, yeni oluşturduğumuz *Geçti* özniteliğinin sahip olduğu değerlerde (True/False) aşağıdaki dönüşümüzü yapmak istediğimizi varsayalım.

```
from sklearn import preprocessing
label_encoder = preprocessing.Labe
lEncoder()
df['Geçti']= label_encoder.fit_tra
nsform(df['Geçti'])
df['Meslek']=label_encoder.fit_tra
nsform(df['Meslek'])
df
```

	İsim	Meslek	Tarih	Yaş	Puan	Geçti	Yıl
0	Ada	2	11.11.2010	21	89	1	2010
1	Cem	2	11.11.2010	24	87	1	2010
2	Sibel	3	11.11.2010	25	77	1	2010
3	Ahmet	4	18.11.2011	44	55	0	2011
4	Mehmet	4	18.11.2011	31	70	1	2011
5	Ali	0	11.11.2010	27	79	1	2010
6	Veli	0	11.11.2010	35	73	1	2010
7	Ayşe	5	11.11.2010	33	79	1	2010
8	Hüseyin	1	11.11.2010	42	54	0	2010
9	Necmi	0	11.11.2010	29	92	1	2010
10	Nalan	0	18.11.2011	41	61	0	2011
11	Namık	3	18.11.2011	43	69	0	2011

Python pandas Kütüphanesi ile sayısallaştırma

Veri Sorunları

- Yanlış özellik değerleri
  - Yazım hataları: ör. renk = {“blue”, “green”, “gren”, “red”}
  - Çöp: ör. renk = “w r□śïį”
  - Tutarsız yazım (ör. "color", «colour") veya büyük harf kullanımı
  - Tutarsız kısaltmalar (ör. “Oak St.”, “Oak Street”)
- Eksik etiketler
  - Yalnızca birkaçında eksik etiket varsa örnekleri silin
  - Yarı denetimli öğrenme tekniklerini kullanın
  - Kendi kendini denetleme yoluyla eksik etiketleri tahmin edin

Veri Ön İşleme için Script Kullanımı

- Bir elektronik tablo programı aracılığıyla manuel olarak düzenleme yapmayı
  - Değişiklik geçmişi yok
  - Hataları tanıtmak çok kolay
  - Daha önceki kararları düzeltmek zor
- Bunun yerine, ham verileri yükleyen ve tüm ön işlemleri yapan bir komut dosyası yazın.
  - Tüm adımları belgeler
  - Artımlı hata ayıklama
  - Daha önceki adımlarda değişiklik yapmak kolay
  - Tekrarlanabilir

- Bütün veri setlerinde ilk incelemek gereken şey

Eksik değer

Eksik hedef değer yok

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	...	SalePrice
count	1022.000000	1022.000000	832.000000	1022.000000	1022.000000	1022.000000	1022.000000	1022.000000	1019.000000		1022.000000
mean	732.338552	57.059687	70.375000	10745.437378	6.128180	5.564579	1970.995108	1984.757339	105.261040		181312.692759
std	425.860402	42.669715	25.533607	11329.753423	1.371391	1.110557	30.748816	20.747109	172.707705		77617.461005
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.000000	1950.000000	0.000000		34900.000000
25%	367.500000	20.000000	59.000000	7564.250000	5.000000	5.000000	1953.000000	1966.000000	0.000000		130000.000000
50%	735.500000	50.000000	70.000000	9600.000000	6.000000	5.000000	1972.000000	1994.000000	0.000000		165000.000000
75%	1100.500000	70.000000	80.000000	11692.500000	7.000000	6.000000	2001.000000	2004.000000	170.000000		215000.000000
max	1460.000000	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.000000	2010.000000	1378.000000		745000.000000

Aykırı değer

Veriyi Anlama

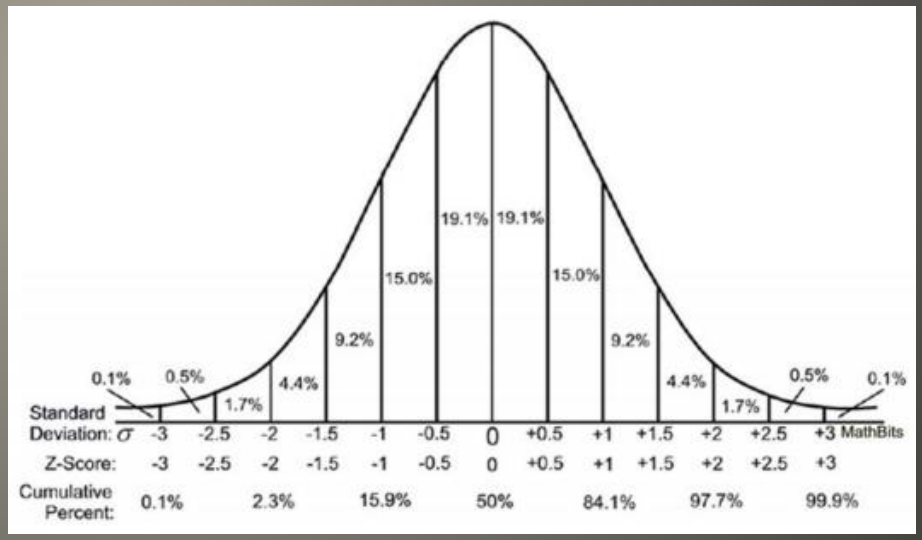
Aykırı Değerlerin Sebebi

- Hatalar
  - Veri toplama veya veri girişinde insan hatası
  - Ölçüm/enstrümantasyon hataları
  - Deneysel hatalar
  - Veri birleştirme hataları
  - Örneğin, farklı ölçeklerle veri kümelerini birleştirme
- Veri ön işleme hataları
  - Doğal
- Verilerdeki yenilikler – hatalar değil!

Aykırı Değerlerin Tespiti: Z-skor

- Özellik değerlerinin Gauss dağılımlı olduğunu varsayın
  - Ortalamadan k standart sapmadan daha uzak noktaları atın
  - k için iyi değerler: 2.5, 3, 3.5+

Uyarılar:
• Çoğunlukla makul ölçüde küçük-orta veri kümelerinde düşük d özellik alanları için
• Parametrik varsayım tutmuyorsa yanlış



Diğer Ön İşlemeler

- Veri Standardizasyonu
  - Sıfır ortalama ve birim varyansa sahip olmak için özellikleri yeniden ölçeklendirir

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$$

  - Her değeri aşağıdaki gibi değiştir.

$$x_{ij} \leftarrow \frac{x_{ij} - \mu_j}{s_j}$$

Hem eğitim hem de tahmin için aynı dönüşümü kullanmalıdır (μj ve sj eğitim verilerinde hesaplanır ve ayrıca test verilerinde kullanılır)

- Veri ölçeklendirme ve normalize etme adımları birbirlerine benzer işler gibi görünseler de (hatta birbirleri yerine kullanılsalar da) uygulanma şekilleri farklıdır. Ölçeklendirme işleminde elimizdeki verinin sadece aralığını (range) değiştirirken (örneğin 0–1 arası ya da 1–100 arası gibi), veriyi normalize etme sürecinde verinin dağılımını normal bir dağılım olarak değiştiriyoruz.

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

```
x = df[['Puan']].values.astype(float)
#Ölçeklendirme için MinMaxScaler fonksiyonunu kullanıyoruz.
min_max_scaler = preprocessing.MinMaxScaler()
x_scaled = min_max_scaler.fit_transform(x)
df['Puan2'] = pd.DataFrame(x_scaled)
df
```

	İsim	Meslek	Tarih	Yaş	Puan	Geçti	Yıl	Puan2
0	Ada	2	11.11.2010	21	89	1	2010	0.921053
1	Cem	2	11.11.2010	24	87	1	2010	0.868421
2	Sibel	3	11.11.2010	25	77	1	2010	0.605263
3	Ahmet	4	18.11.2011	44	55	0	2011	0.026316
4	Mehmet	4	18.11.2011	31	70	1	2011	0.421053
5	Ali	0	11.11.2010	27	79	1	2010	0.657895
6	Veli	0	11.11.2010	35	73	1	2010	0.500000
7	Ayşe	5	11.11.2010	33	79	1	2010	0.657895
8	Hüseyin	1	11.11.2010	42	54	0	2010	0.000000
9	Necmi	0	11.11.2010	29	92	1	2010	1.000000
10	Nalan	0	18.11.2011	41	61	0	2011	0.184211
11	Namık	3	18.11.2011	43	69	0	2011	0.394737

Veriyi Ölçeklendirme (Scaling) ve Normalize Etme

	LotFrontage	LotArea	Street	LotShape	Utilities	LandSlope	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	ExterQual	ExterCond	BsmtQual	BsmtExposure	BsmtFinType1	BsmtFinSF1	BsmtFinType2	...	SaleCondition_Abnorml
0	65.0	8450	2	4	4	3	7	5	2003	2003	196.0	4	3	4	0	6	706	1	...	0
1	80.0	9600	2	4	4	3	6	8	1976	1976	0.0	3	3	4	3	5	978	1	...	0
2	68.0	11250	2	3	4	3	7	5	2001	2002	162.0	4	3	4	1	6	486	1	...	0
3	60.0	9550	2	3	4	3	7	5	1915	1970	0.0	3	3	3	0	5	216	1	...	1
4	84.0	14260	2	3	4	3	8	5	2000	2000	350.0	4	3	4	2	6	655	1	...	0
5	85.0	14115	2	3	4	3	5	5	1993	1995	0.0	3	3	4	0	6	732	1	...	0
6	75.0	10084	2	4	4	3	8	5	2004	2005	186.0	4	3	5	2	6	1369	1	...	0
7	0.0	10382	2	3	4	3	7	6	1973	1973	240.0	3	3	4	1	5	859	4	...	0
8	51.0	6120	2	4	4	3	7	5	1931	1950	0.0	3	3	3	0	1	0	1	...	1
9	50.0	7420	2	4	4	3	5	6	1939	1950	0.0	3	3	3	0	6	851	1	...	0
10	70.0	11200	2	4	4	3	5	5	1965	1965	0.0	3	3	3	0	3	906	1	...	0
11	85.0	11924	2	3	4	3	9	5	2005	2006	286.0	5	3	5	0	6	998	1	...	0
12	0.0	12968	2	2	4	3	5	6	1962	1962	0.0	3	3	3	0	5	737	1	...	0
13	91.0	10652	2	3	4	3	7	5	2006	2007	306.0	4	3	4	2	1	0	1	...	0
14	0.0	10920	2	3	4	3	6	5	1960	1960	212.0	3	3	3	0	4	733	1	...	0
15	51.0	6120	2	4	4	3	7	8	1929	2001	0.0	3	3	3	0	1	0	1	...	0
16	0.0	11241	2	3	4	3	6	7	1970	1970	180.0	3	3	3	0	5	578	1	...	0
17	72.0	10791	2	4	4	3	4	5	1967	1967	0.0	3	3	0	0	0	0	0	...	0
18	66.0	13695	2	4	4	3	5	5	2004	2004	0.0	3	3	3	0	6	646	1	...	0
19	70.0	7560	2	4	4	3	5	6	1958	1965	0.0	3	3	3	0	2	504	1	...	1
20	101.0	14215	2	3	4	3	8	5	2005	2006	380.0	4	3	5	2	1	0	1	...	0
21	57.0	7449	2	4	4	3	7	7	1930	1950	0.0	3	3	3	0	1	0	1	...	0
22	75.0	9742	2	4	4	3	8	5	2002	2002	281.0	4	3	4	0	1	0	1	...	0
23	44.0	4224	2	4	4	3	5	7	1976	1976	0.0	3	3	4	0	6	840	1	...	0

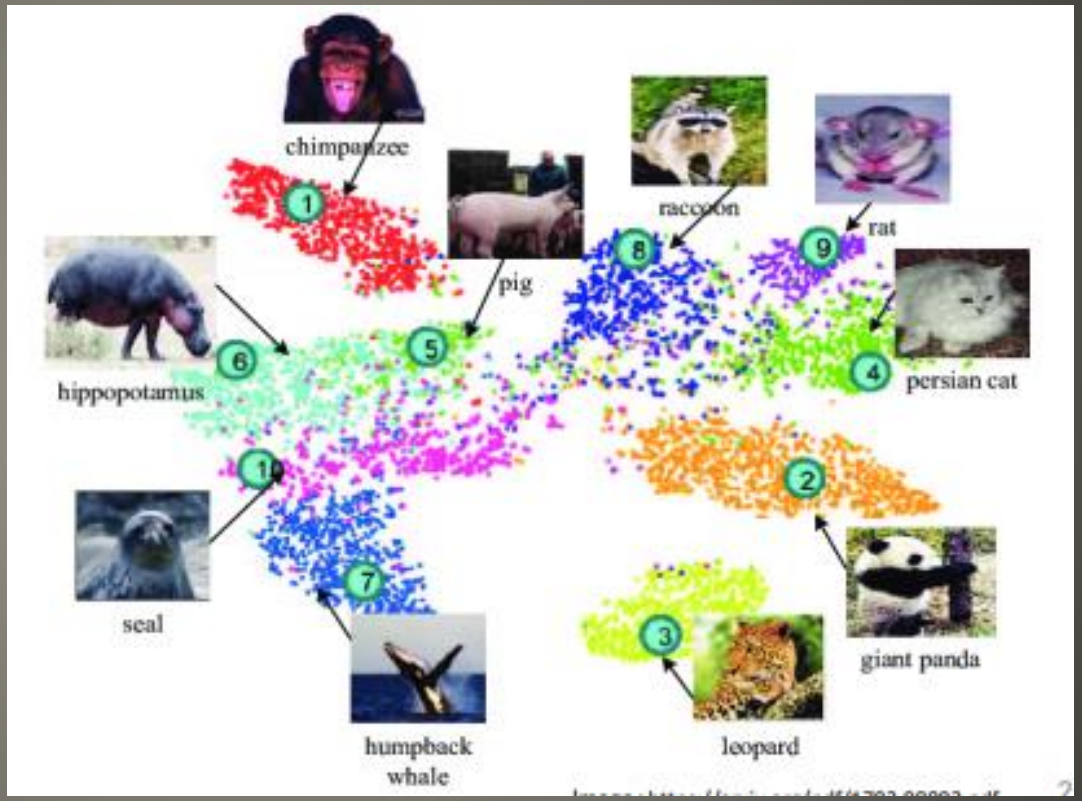
227 features

Temel Bileşen Analizi ile Veri Gösterimi

- Ham özelliklerden daha iyi bir temsil var mı?

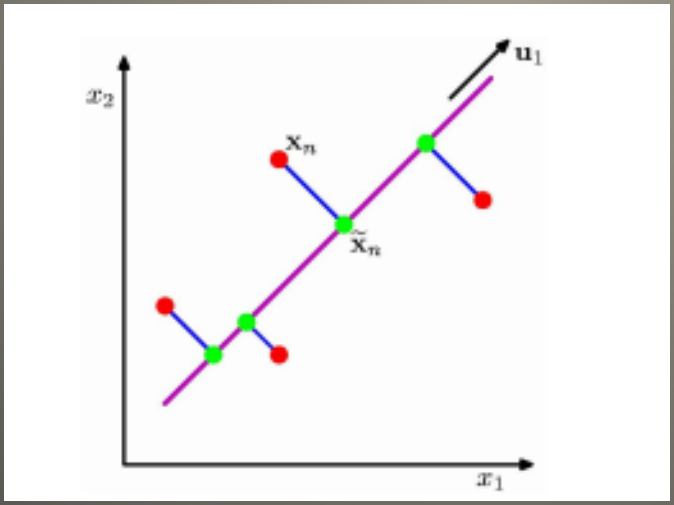
Fikir: Orijinal verilerle ilgili bilgilerin çoğunu tutan bir alt uzay bulun

Birçok farklı yöntem olmasına rağmen, odak noktamız şu olacak:
Temel bileşenler Analizi



Temel Bileşen Analizi ile Veri Gösterimi

- Aşağıdakileri sağlayan alt boyutlu doğrusal uzaya verilerin dikey izdüşümü:
  - öngörülen verilerin varyansını maksimize eder (mor çizgi)
  - veri noktası ve projeksiyonlar arasındaki ortalama kare mesafeyi en aza indirir (mavi çizgilerin toplamı)

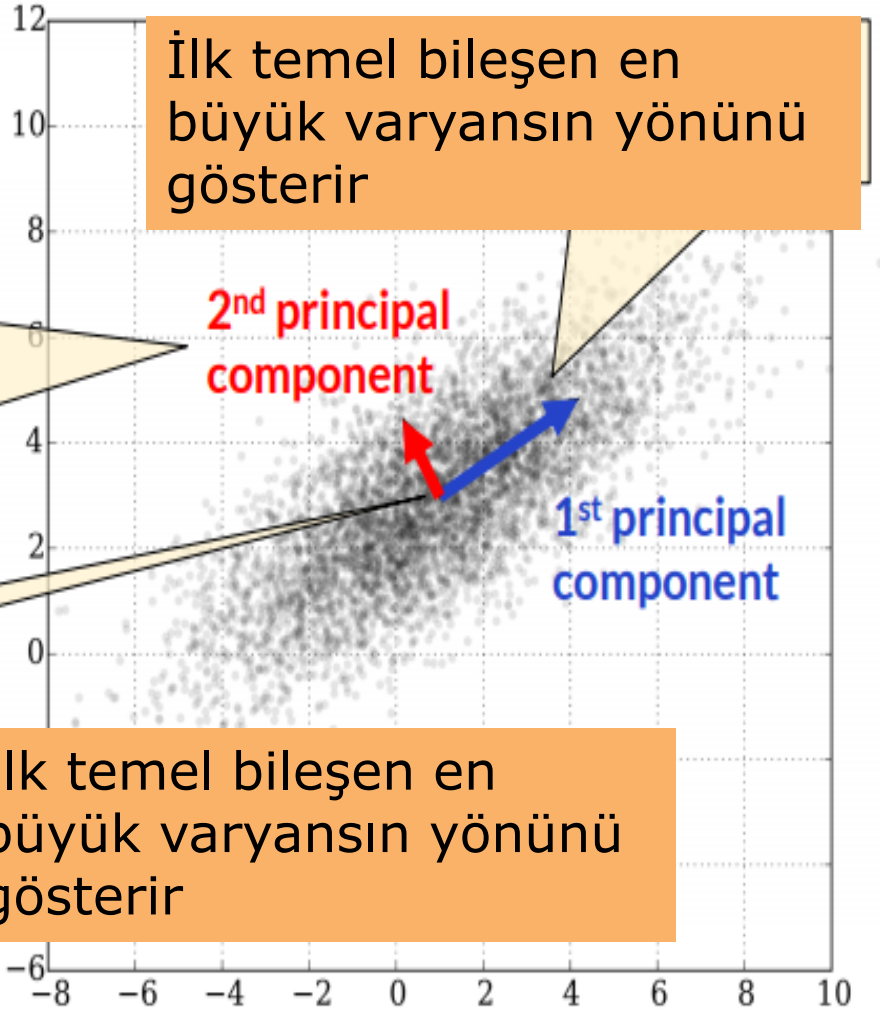


Temel Bileşen Analizi

Temel Bileşen Analizi

Sonraki her bir temel bileşen:
- önceki tüm bileşenlere ortogonaldir
- alt uzayın en büyük varyansının yönünü gösterir

İlk temel bileşen en büyük varyansın yönünü gösterir

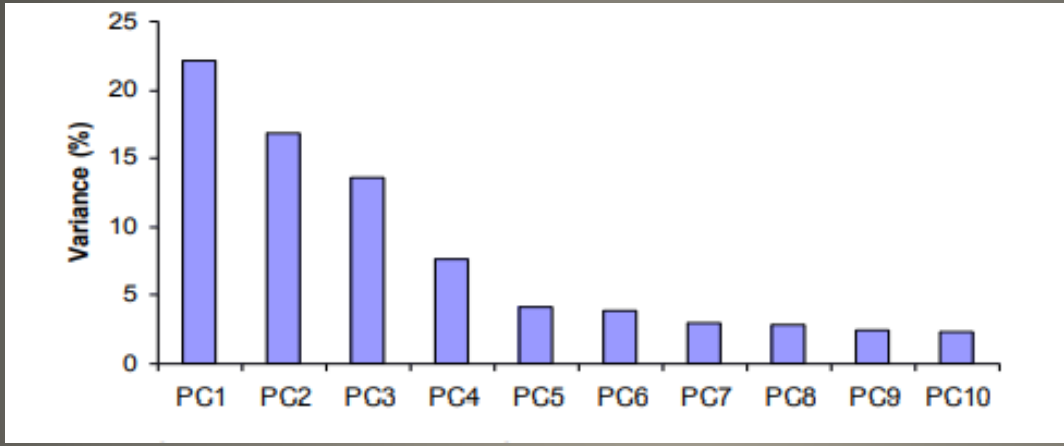


Temel vektörler ortalamadan kaynaklanır

İlk temel bileşen en büyük varyansın yönünü gösterir

Temel Bileşen Analizi

- Daha az öneme sahip bileşenleri görmezden gelebilir.

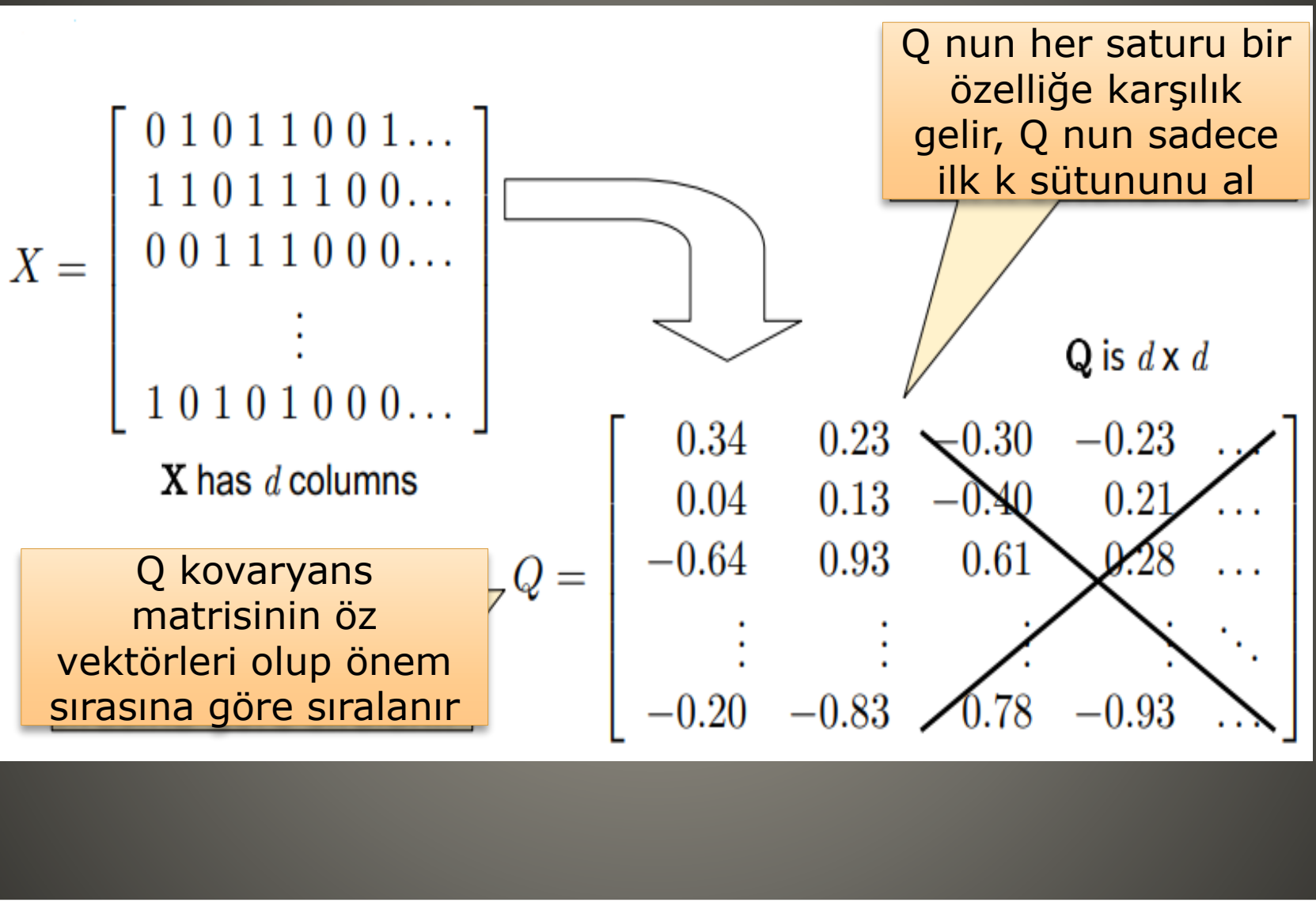


- • Bazı bilgileri kaybedersiniz, ancak özdeğerler küçükse, fazla bir şey kaybetmezsiniz.
  - özdeğerlerine göre yalnızca ilk k özvektörleri seçin
  - nihai veri seti sadece k boyuta sahiptir

Temel Bileşen Algoritması

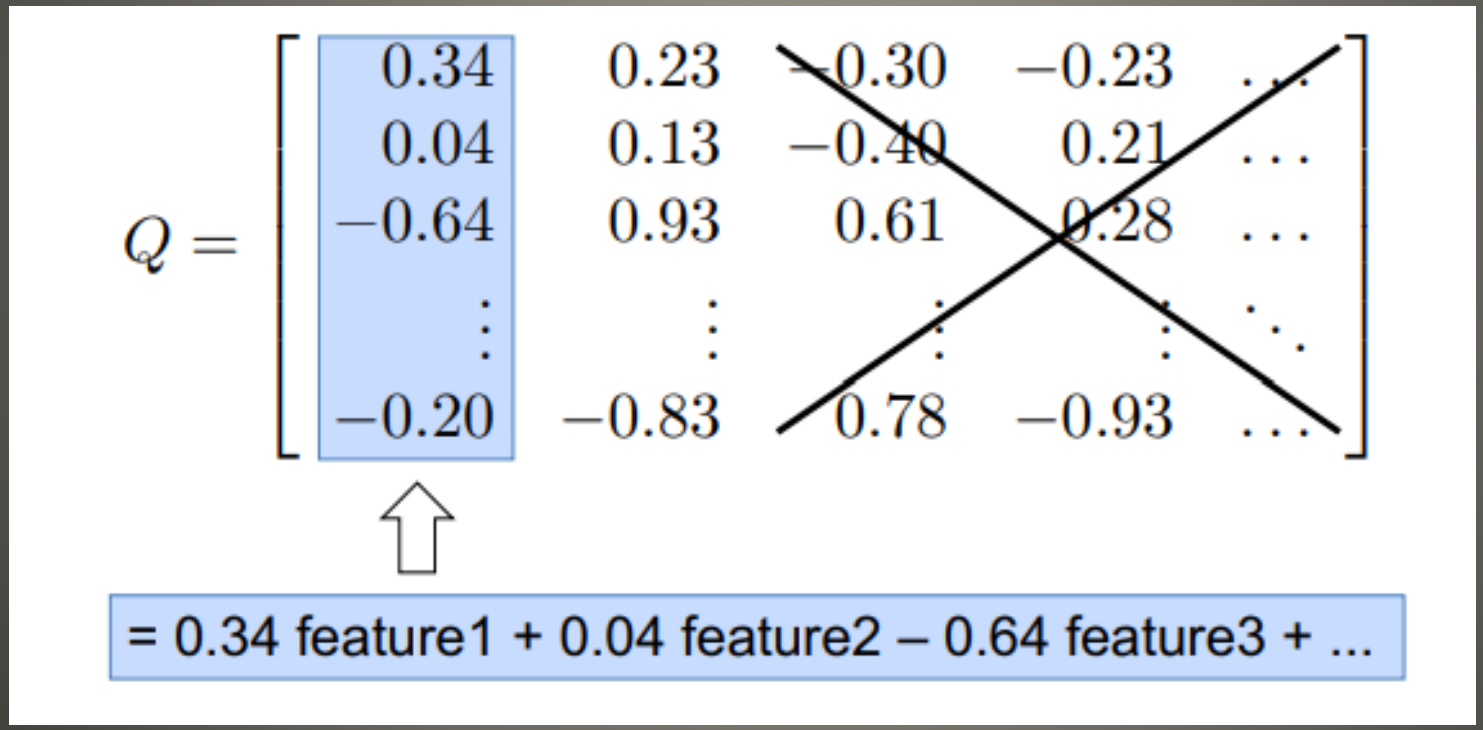
- Verilen veri {x1, …, xn}, kovaryans Σ matrisi 'yi hesaplayın
- o X, n x d boyutlu veri matrisidir
- o Verilerin ortalamasını hesapla (tüm X satırlarının ortalaması)
- o Her X satırından ortalamayı çıkar (verileri ortalayarak)
- o kovaryans matrisini hesapla
- • PCA tabanlı vektörler, S'nin özvektörleri tarafından verilir.

$$\Sigma = X^T X$$



Temel Bileşen Analizi

- Q'nun her sütunu, orijinal özelliklerin doğrusal bir kombinasyonu için ağırlıklar verir.



Temel Bileşen Analizi

- Her x örneği için yeni temsili elde etmek için bu formülleri uygulayın

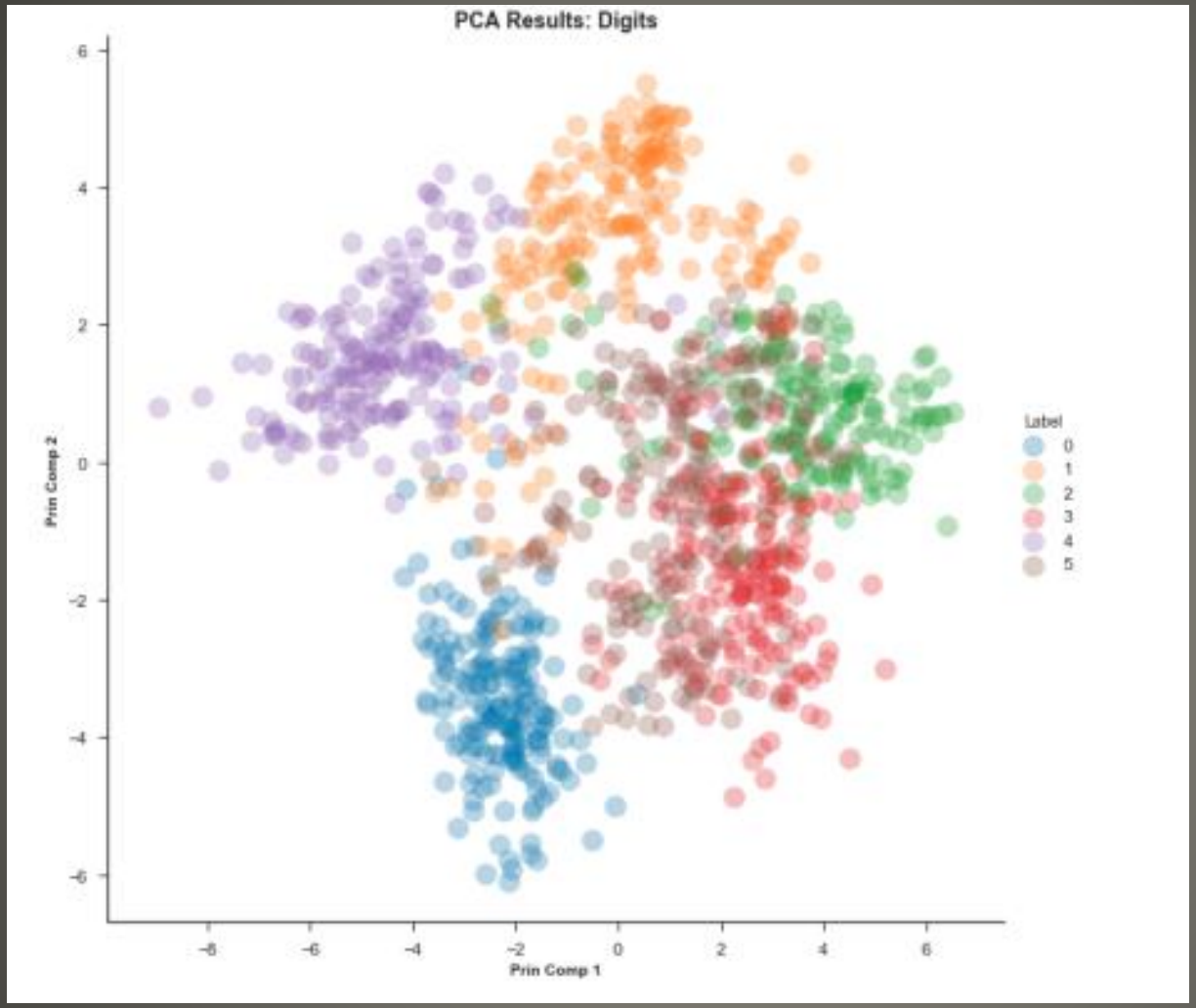
$$X = \begin{bmatrix} 0\,1\,0\,1\,1\,0\,0\,1\ldots \\ 1\,1\,0\,1\,1\,1\,0\,0\ldots \\ 0\,0\,1\,1\,1\,0\,0\,0\ldots \\ \vdots \\ 1\,0\,1\,0\,1\,0\,0\,0\ldots \end{bmatrix} \quad x_3 \qquad \hat{Q} = \begin{bmatrix} 0.34 & 0.23 \\ 0.04 & 0.13 \\ -0.64 & 0.93 \\ \vdots & \vdots \\ -0.20 & -0.83 \end{bmatrix}$$

- x3 için yeni 2B gösterimi şu şekilde verilir:

$\hat{x}_{31}$ = 0.34(0) + 0.04(0) - 0.64(1) + ...
$\hat{x}_{32}$ = 0.23(0) + 0.13(0) + 0.93(1) + ...

- Yeniden yansıtılan veri matrisi X = XQ ile verilir

Temel Bileşen Analizi



Rakamların Temel Bileşen Analizi ile Görselleştirilmesi

Özellik Azaltımı için Temel Bileşen Analizi Kullanımı

- Orijinal özellikler yerine verilerin temel bileşen dönüşümünü kullanabiliriz
  - Yalnızca k < d PCA özelliklerini koru
- Temel bileşen analizi, verilerin çoğu varyansını tutar
- • Bu nedenle, veri kümesini, veri kümesinin anlamlı varyasyonlarını koruyan özelliklere indirgiyoruz.