

# Data Mining: Concepts and Techniques

## Chapter 1: Introduction

### What Is & Why Data Mining?

### What Kind of Data Can Be Mined?

- Database Data
- Data Warehouses
- Transactional Data
- Other Kinds of Data

### What Kinds of Patterns Can Be Mined?

- Class/Concept Description: Characterization and Discrimination
- Mining Frequent Patterns, Associations, and Correlations
- Classification and Regression for Predictive Analysis
- Cluster Analysis
- Outlier Analysis
- Are All Patterns Interesting?

### What Technology Are Used?

- Statistics
- Machine Learning
- Database Systems and Data Warehouses
- Information Retrieval

## Why & What Is Data Mining?

### Veri Madenciliğinin Doğusu

Her gün dünya üzerinde bulunan datanın gittikçe büyümesi ile birlikte büyüyen bu datanın analiz ihtiyacı ortaya çıkmıştır. Büyüyen bu mevcut verilerin güçlü kaynaklar kullanılarak bilgi formuna dönüşmesi ihtiyacı sonucunda bu işi yapan Veri Madenciliği doğmuştur.

Veri madenciliği bu analizi çeşitli tool/araçlar kullanarak yapar. Buradaki tool kavramı aslında methodlarımızdır.

Data Mining kavramını şu isimlendirmeler ile görmek de mümkündür:

- Knowledge Discovery From Data (KDD)
- Knowledge Mining From Data
- Knowledge Extraction Data/Pattern Analysis
- Data Archaeology
- Data Dredging

### Veri ve Bilgi Karşılaştırması

Veri ve bilgi kavramları farklı şeylerdir. Her veri bir bilgi, her bilgi de bir veri olmayabilir. Verinin içerisinde faydalı veya faydasız her şey olabilir. Verinin içinden anamlı sonuçlar elde ettiğimizde bu sonuçlar bilgi olarak değerlendirilir. Bilginin veri olabilmesi için de somut olması gerekmektedir.

Veri madenciliği en temelde veriden bilgi çıkarmamızı sağlamaktadır. Örneğin doktora giden bir hasta doktorun teşhis koyabilmesi adına rahatsızlıklarını gerekli veya gereksiz bir takım ifadelerle dile getirir. Bunlar doktora ulaşan verilerdir. Doktor hastadan röntgen veya kan tahlili gibi bir takım isteklerde bulunur ve bu araçları kullanarak, elindeki verilerden bilgiye ulaşmaya çalışır. En neticede doktorun bir takım araçları kullanarak bilgiye ulaşma çabası veri madenciliği olarak düşünülebilir.

### Veritabanı Teknolojilerinin Evrimi

#### 1960s:

- Data Collection
- Database Creation
- IMS and network DBMS

#### 1970s:

- Relational Data Model
- Relational DBMS Implementation

#### 1980s:

- RDBMS
- Advanced Data Models (extended-relational, OO, deductive, etc.)
- Application-oriented DBMS (spatial, scientific, engineering, etc.)

## 1990s:

- Data mining
- Data warehousing
- Multimedia Databases and Web Databases

## 2000s

- Stream data management and mining
- Data mining and its applications
- Web technology (XML, data integration) and global information systems

### Veri Ambarı Nedir? (*Data Warehousing*)

Derin bir analizin gerekliliği neticesinde ortaya çıkmıştır. Farklı veri kaynaklarının aynı çatı altında birleştirilmesi olarak ifade edilebilir.

### Veri Kaynakları Nelerdir?

- Veritabanları
- Veri Ambarları
- Web
- ....

Data Mining bu verileri işleyerek bir sonuç/bilgi elde eder.

### Veriden Bilgi Keşfetmenin (KDD) / Veri Madenciliğinin Adımları

#### 1. *Data cleaning / Veri Temizlemesi* (to remove noise and inconsistent data)

Gelen veriye direkt olarak herhangi bir yöntem uygulanamaz. Verinin önce gürültülerinden ve tutarsızlıklarından arındırılması/temizlenmesi gereklidir.

#### 2. *Data integration / Veri Birleştirilmesi* (where multiple data sources may be combined)

Farklı veri kaynaklarını birleştirme işlemidir. Verdiğimiz doktor-hasta örneğinde doktorun analiz yapabilmek için hastadan istediği röntgen ve kan tahlili iki farklı veri kaynağı olarak düşünülebilir. Bilgiyi elde edebilmek için bu iki kaynağın birlikte değerlendirilmesi gereklidir. Bunun için bu iki kaynağı birleştiririz.

#### 3. *Data selection / Veri Seçimi* (where data relevant to the analysis task are retrieved from the database)

Yalnızca analiz yapacağımız ilgili dataya ulaşmamız durumudur. Örneğin bir okulun veritabanında öğrenciler hakkında bir veriye ulaşmak için sınıflar ve öğretmenlerin yer aldığı verilere ulaşmam gerekmek.

#### 4. *Data transformation / Veri Transferi* (where data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations)

Verinin analiz edilmeden önce belirli bir formda olması gereklidir. Verinin uygun biçimde dönüştürülmesi edilmesi kısımdır.

**5. Data mining** (*an essential process where intelligent methods are applied to extract data patterns*)

Onceki dört aşamayı uyguladıktan sonra data mining işlemi uygulanabilir. Bu kısım- da zeki methodlar uygulanarak veri kalıpları çıkarılır.

**6. Pattern evaluation / Model Değerlendirmesi** (*to identify the truly interesting patterns representing knowledge based on interestingness measures*)

Bu aşamaya kadar bir model elde ederiz. Bu kısımda modelimizin performansını performans metrikleri/ölçütleri ile ölçeriz ve değerlendiririz. Bu ölçütler aşağıdaki gibidir.

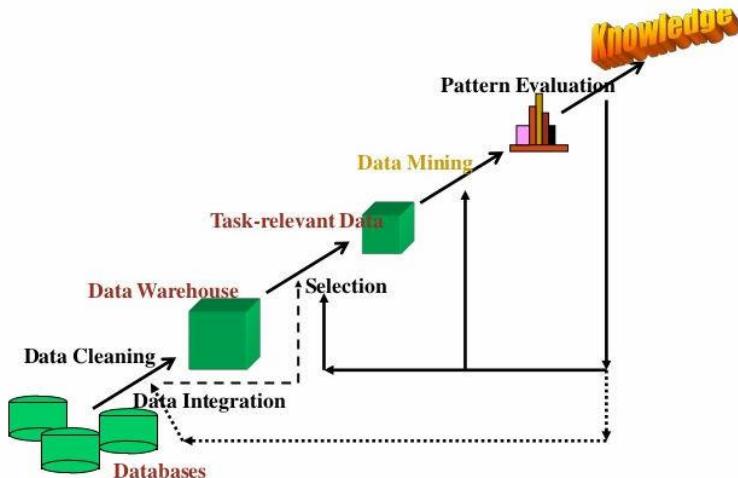
- Gerçekten bu modelden bir bilgi çıktı mı? (Her model bilgi içermez.)
- Model güvenilir mi?
- Model sağlam mı?

**7. Knowledge Presentation / Bilgi Sunumu** (*where visualization and knowledge representation techniques are used to present mined knowledge to users*)

Modelimizi iyi bir şekilde sunmamız gereklidir. Modelimizde orijinal olan şeyleri güzel pazarlayabilme ve sunma işlemidir.

1-4 arası adımlar: **Data Preprocessing**

Verinin işlenmek için hazırlandığı süreç “Data Preprocessing” olarak adlandırılır ve bu başlık altında yukarıda belirttiğimiz ilk dört süreç bulunur.



## Hangi Tür Datalar İşlenebilir?

Veri tipi ne olursa olsun o veriye data mining uygulanabilir. Fakat işlemeye yönelik en basit form ***Database Data***, ***Datawarehouse Data*** ve ***Transactional (İşlemsel) Data***'dır. Bunun yanında aşağıdaki gibi başka türde verilere de data mining uygulanabilir:

- Graph or networked data
- Data streams and sensor data
- Time series data, temporal data, sequence data (incl. bio sequences)
- Structure data, graphs, social networks and multi linked data
- Object relational databases
- Heterogeneous databases and legacy databases
- Spatial data and spatiotemporal data
- Multimedia database
- Text databases
- The World Wide Web

## Hangi Tür Modeller İşlenebilir?

Cok sayıda veri madenciliği aşaması/fonksiyonu vardır. Bunlardan bazı ana başlıklar:

- *Characterization* (Karakterize Etme) & *Discrimination* (Ayırt Etme)
- *The Mining of Frequent Patterns* (Sık Kalıpların Madenciliği), *Associations* (Birlikteşlik) & *Correlations* (Korelasyon)
- *Classification and Regression* (Sınıflandırma ve Regresyon)
- *Clustering & Outlier Analysis* (Kümeleme ve Aykırı Değerler Analizi)

Veriyi tanımlarken daha iyi bir model oluşturabilmek için bu özellikleri kullanmamız gereklidir.

Veri madenciliği fonksiyonlarını iki kategoriye ayıralım:

- 1) **Descriptive:** Tanımlama yapılır. Verinin özelliklerini karakterize ederler.
- 2) **Predictive:** Veriyi kullanarak tahmin etme işlemi yapılır.

### a) Data Characterization

Verinin sınıfını, veriyi özetleyecek kelimeleri kullanmak.

### b) Data Discrimination

Hedef classın karşılaştırılması yapılarak (bir veya daha fazla diğer classlarla) veriyi diğer sınıflarla kıyaslayıp ayırt etmemizi sağlamak.

### c) Mining Frequent Patterns

Veride sıklıkla oluşan bir kalıptır. Üç adet sıklık kalıbü vardır:

1. *Frequent Itemsets*: Aynı işlem sırasında birlikte gözüken ürünlerin kümesi.
2. *Frequent Subsequences (Sequential Patterns)*: Sıralı işlemleri ifade eder.
3. *Frequent Substructures*: Bu iki kavramın birlikte olma durumudur.

*Örnek*: Markete gittin. Bir ekmek ve bir yumurta aldı. Senden sonra gelen müşteriler de aynı alışveriş yaptı. Kasiyerin gözünde aynı işlemin birlikte gözüktüğü bir durum söz konusu olur. Bu Frequent Itemsets olarak ifade edilir. Eve gittiğinde yumurtayı haşlamak için bir cezveye ihtiyacın olduğunu öğrenebilirsin. Tekrar markete gider ve cezve alırsın. Bu durum da sıralı işlem olarak Frequent Subsequences olarak ifade edilir.

### d) Association Analysis

Örnekler üzerinden açıklayalım:

$\text{buys}(X, \text{"computer"}) \Rightarrow \text{buys}(X, \text{"software"})$  [support = 1%, confidence = 50%]

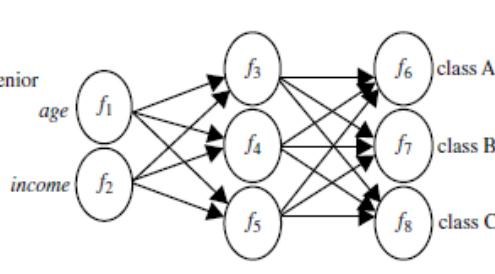
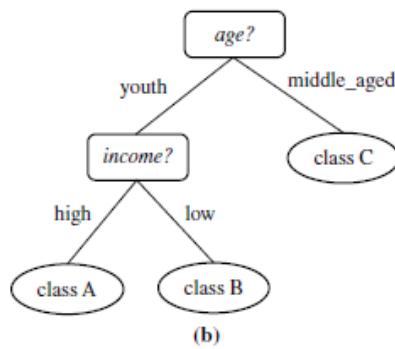
Eğer X müsterisi bilgisayar satın alıyor ise yazılım ürünlerinden birini de satın alıyor. Bilgisayarın yanında yazılımı alması için 50% ihtimal var. B ilgisayar alıyor ise yazılım ürünü alacağı kesindir ifadesi %1.

$\text{age}(X, \text{"20..29"}) \wedge \text{income}(X, \text{"40K..49K"}) \Rightarrow \text{buys}(X, \text{"laptop"})$  [support = 2%, confidence = 60%]

X müsterisi 20 ile 29 yaş arasında ise ve geliri 40 ile 49 bin arasında ise laptop alır. 60% oranında alırlar ve 2% oranında kesindir.

$age(X, "youth") \text{ AND } income(X, "high")$	$\longrightarrow class(X, "A")$
$age(X, "youth") \text{ AND } income(X, "low")$	$\longrightarrow class(X, "B")$
$age(X, "middle\_aged")$	$\longrightarrow class(X, "C")$
$age(X, "senior")$	$\longrightarrow class(X, "C")$

(a)



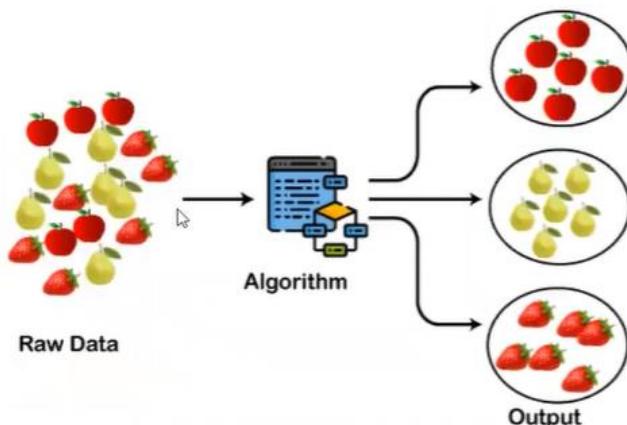
A classification model can be represented in various forms: (a) IF-THEN rules, (b) a decision tree, or (c) a neural network.

- a) Yaşı genç ve geliri yüksekse sınıfı A. Yaşı genç ve geliri az ise sınıfı B...
- b) Sınıflandırma Algoritmalarından bir tanesi olan **Decision Tree. (If-Then rules)**
- c) Neural Network yapısı. Nasıl üretildiği hakkında bir bilgimiz yok.

#### f) Cluster Analysis

Classification ve regression analizlerinde training setinde sınıflandırılmış etiketlerle(labels) çalışıyoruz. Kümeleme algoritmasında ise veri sınıf etiketine bakılmaksızın/başvur- maksızın analiz edilir.

Çoğu durumda sınıf etiketi başlangıçta yoktur. Sınıf etiketlerini oluşturmak için kümeleme algoritması kullanılabilir. Bu algoritma verileri kümeleyerek farklı veri kümeleri oluşturur. Etiketlenmemiş veri setlerini gruplar.



### g) Outlier Analysis

Bir veri seti, verinin genel davranışına veya modeline uymayan nesneler içerebilir. Bu veri nesneleri aykırı değerlerdir. Birçok veri madenciliği yöntemi, aykırı değerleri gürültü veya istisna olarak atar.

Örneğin insan yaşamının en fazla 150 olduğu bir veri setinde bir insanın 600 yıl yaşadığı verisi bulunabilir. Bu tür aykırı veriler Aykırı Algoritması ile ayıklanır.

İki veri arasındaki benzerliği bulmak için bir çok yöntem vardır. Bunlardan birisi, uzaklık belirlemektir. İki veri arasındaki uzaklık belirlenen bir sınırın altında ise bu iki veri Cluster Algoritması ile grüplendirilebilir. Eğer uzaklık sınırın üstünde ise bu veriler aykırı olarak değerlendirilir ve Aykırı Algoritması ile ayıklanır.

Future Selection: Oluşturacağımız sınıflandırma modelinde önemli özelliklerini belirlemede ve gereksiz olan özelliklerini atar.

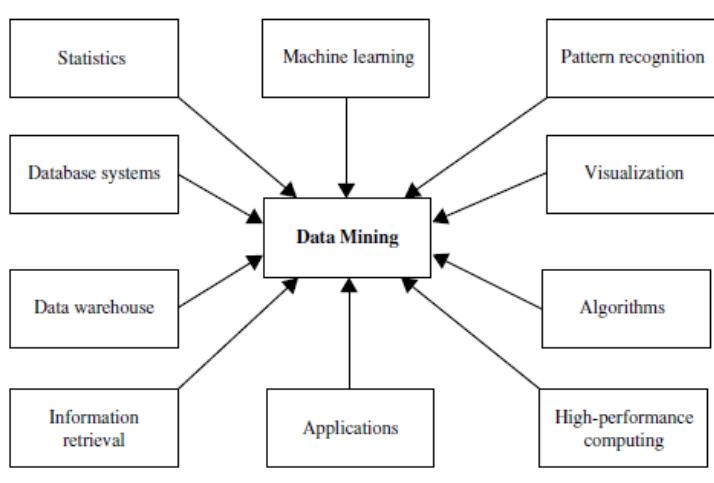
### Bütün Kalıplar İlgi Çekici Midir? Bir Kalıbı İlgi Çekici Yapan Özellikler Nelerdir?

Data Mining milyonlarca pattern üretebilir ve bunlardan yalnızca bazıları ilgi çekici olabilir. Bir modelin ilginç olması için modelimizin;

- *Herkes tarafından anlaşılabilir*
- *Geçerli (doğu)*
- *Faydalı*
- *Yeni (daha önceden oluşmamış)*

olması gereklidir.

### Hangi Teknolojiler Kullanıldı?



#### Statics

Veri setini özetleycek kelimeler ve cümleler kurabilmemiz için istatistikten yararlanır. Verinin olduğu yerde istatistik de olmalı.

## Machine Learning

Veriye dayanarak bilgisayarın nasıl öğreneceğine dayanır. Kendi içinde şu başlıklara ayrıılır:

- **Supervised Learning:** Gözetimli öğrenme. Eğitim kümesindeki etiketlenmiş örnekler.
- **Unsupervised Learning:** Gözetimsiz öğrenme. Veriler sınıfısız ise gruplandırma yapılır.
- **Semi-supervised Learning:** Supervised ve Unsupervised Learning içerir.
- **Active Learning:** Kullanıcıya aktif rol veren bir yöntemdir.

Gözetim verinin etiketinden gelir.

### Örnekler:

Classification ve Regression algoritmaları *Supervised Learning* algoritmalarından biridir.

Clustering algoritması *Unsupervised Learning* algoritmalarından biridir.

## **Chapter 2: Getting to Know Your Data**

### **❑ Data Objects and Attribute Types**

- What Is an Attribute?
- Nominal Attributes
- Binary Attributes
- Ordinal Attributes
- Numeric Attributes
- Discrete Versus Continuous Attributes

### **❑ Basic Statistical Descriptions of Data**

- Measuring the Central Tendency: Mean, Median, and Mode
- Measuring the Dispersion of Data: Range, Quartiles, Variance, Standard Deviation, and Interquartile Range
- Graphic Displays of Basic Statistical Descriptions of Data

### **❑ Measuring Data Similarity and Dissimilarity**

- Data Matrix versus Dissimilarity Matrix
- Proximity Measures for Nominal Attributes
- Proximity Measures for Binary Attributes
- Dissimilarity of Numeric Data: Minkowski Distance
- Proximity Measures for Ordinal Attributes
- Dissimilarity for Attributes of Mixed Types
- Cosine Similarity

## Data Objects and Attribute Types

Preprocessing yapabilmek için öncelikle elimizdeki datayı tanıtmamız gereklidir. Bunun için aşağıdaki soruların cevaplarını bilmemiz gereklidir:

- What are the types of attributes or fields that make up your data?
- What kind of values does each attribute have?
- Which attributes are discrete, and which are continuous-valued?
- What do the data look like?
- How are the values distributed?
- Are there ways we can visualize the data to get a better sense of it all?
- Can we spot/test any outliers?
- Can we measure the similarity of some data objects with respect to others?

### Veri Nesnesi Nedir? (Data Objects)

Veri setleri veri nesnelerinden oluşur. Veri nesnesi bir varlık temsil eder. Veri nesneleri attribute ler ile tanımlanır. Object yerine şu terimlerin de kullanımı yaygındır:

- Samples
- Examples
- Instances
- Data points

### Attribute Nedir?

Veri nesnesinin özelliklerini ve karakterini temsil eder. Veriyi tanıtmamıza yardımcı olur. Attribute yerine şu terimlerin de kullanımı yaygındır:

- Dimension (Genellikle veri ambarında kullanılır.)
- Feature (Genellikle makine öğreniminde kullanılır.)
- Variable

*Bivariate Attributes:* İki veya daha fazla attribute'a sahip olduğumuzda verdığımız isim.

*Univariate Attributes:* Tek attribute'a sahip olduğumuzda verdığımız isim. Tek değişkenli.

Attribute'lar tip olarak dörde ayrılır:

- Nominal attributes
- Binary attributes
- Ordinal attributes
- Numeric attributes

**a) Nominal Attributes:** Bir şeylerin isimleri veya sembolü değerine sahip attributelardır. "Categorical" olarak da adlandırılır.

*Örnek: Elimizde "person" veri kümemiz olsun. Bu person verisini tanımlamak için hair\_color ve marital\_status olarak iki tane attribute'a sahibiz (Bivariate attribute). Bu attribute'lar string değerler alabilir. Ayrıca örneğin saç rengi için siyah 0, gri 1 gibi sembolize de edilebilir. Bu değerler saç renklerinin ortalması gibi herhangi bir nicelikli hesaplamaya dahil edilemez.*

**b) Binary Attributes:** Yalnızca iki değer alabilen bir nominal attribute'dur. Her binary attribute nominal attribute'dur ama her nominal bir binary attribute değildir. Çünkü nominal attributes birden çok değer alabilen özelliklere sahip olabilir ama binary attributes yalnızca iki değer alabilir.

Örnek: Ya 0 ya da 1, ya kız ya da erkek.

**c) Ordinal Attributes:** Sıralama veya derecelendirme var ise ordinal attribute olur. Ordinal attribute'lar da nominal'dır.

*Örnek: Notlandırmada AA, BB, CC gibi değerler alabilen sınav sonuçları vardır. Veya lisans, master, doktora gibi sırayla ilerleme katedilen bir akademik kariyer.*

**d) Numeric Attributes:** Niceliksel ve ölçülebilir ise numeric olur. Kendi içerisinde ayrırlar:

- *Interval-Scaled Attributes:* Bağıl sıfır (0 koyduğunda o şey yok olmuyor ise)
- *Ratio-Scaled Attributes:* Mutlak sıfır (0 koyduğunda o şey yok oluyor ise)

Örnek:

- "Bugün hava ısısı 5°C derece."

Burada 5 yerine 0 koyduğumuzda havanın ısısı olmadığı anlamına, ısının yok olduğu anla- mına gelmez. Bu sebeple Interval attribute.

- "Arabanın saatteki hızı 50Km'dir."

Burada 50 yerine 0 koyduğumuzda arabanın hızı yok olur. Bu sebeple Ratio-Scaled attribute-dur.

### Discrete & Continuous Attributes

Sınıflandırma algoritmalarında bir attribute discrete veya continuous olarak değerlendirilir.

*Discrete Attributes* sayılabilir özellikli attribute'lar için geçerlidir. (Countably özelliği)

Sayılabılır yerine ölçülebilir özellikli attribute ise *Continuous Attribute*. (Measurable özelliği)

Örnek:

Arabanın hızı => Ölçülür => Continuous Attribute

Sokaktaki kedi sayısı => Sayılır => Discrete Attribute

Kilo bilgisi => Ölçülür => Continuous Attribue

## Basic Statistical Descriptions of Data

İstatistiksel tanımlamalar verilerimizin özelliklerini/attributelerini belirlemede kullanılır. Verideki gürültüleri ve aykırı değerleri istatistiksel tanımlamalar ile bulabiliriz.

Merkezi eğilim ölçülerini hesaplamak için 4 adet istatistiksel ifade vardır:

- Mean
- Median
- Mode
- Midrange

Bir verinin bu bilgilerini bilirse o verinin dağılımı hakkında fikir sahibi olup yorum yapabiliriz.

### Measuring the Central Tendency: Mean, Median, and Mode

#### a) Mean

On tane işçinin ortalama maaşını bulmak isteyelim:

$$\bar{x} = \frac{\sum_{i=1}^N x_i}{N} = \frac{x_1 + x_2 + \dots + x_N}{N}. \quad (2.1)$$

This corresponds to the built-in aggregate function, *average* (*avg()*) in SQL, provided in relational database systems.

**Example 2.6 Mean.** Suppose we have the following values for *salary* (in thousands of dollars), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110. Using Eq. (2.1), we have

$$\begin{aligned}\bar{x} &= \frac{30 + 36 + 47 + 50 + 52 + 52 + 56 + 60 + 63 + 70 + 70 + 110}{12} \\ &= \frac{696}{12} = 58.\end{aligned}$$

Thus, the mean salary is \$58,000. ■

Sometimes, each value  $x_i$  in a set may be associated with a weight  $w_i$  for  $i = 1, \dots, N$ . The weights reflect the significance, importance, or occurrence frequency attached to their respective values. In this case, we can compute

$$\bar{x} = \frac{\sum_{i=1}^N w_i x_i}{\sum_{i=1}^N w_i} = \frac{w_1 x_1 + w_2 x_2 + \dots + w_N x_N}{w_1 + w_2 + \dots + w_N}. \quad (2.2)$$

This is called the **weighted arithmetic mean** or the **weighted average**.

Mean aykırı değerlere karşı hassastır. Örneğin veri setimizindeki işçilerden birinin salary bilgisi 1mn dolar olabilir.

#### b) Median

Verinin merkezini en iyi şekilde ölçen yöntemdir. Sıralanmış bir veri kümесinin orta/merkez değeridir. Çift sayı ise  $N+1 / 2$

Örnek: Yukarıdak pay kısmında sıralanmış işçi maaşlarının ortadaki iki değeri 52 ve 56 dir (toplamlık değer sayı çift olduğu için 2 sayı ortada olur). Bunları toplayıp 2 ye bölerek aritmetik ortalamasını alırız ve bu değer bizim medyanımız olur.

### c) Mode

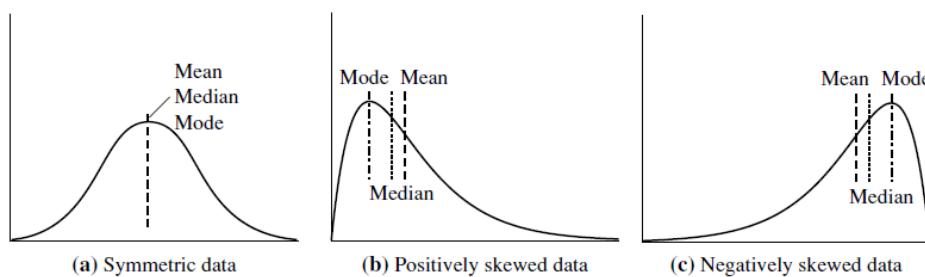
Bir seride en sık geçen değerdir.

Örnek:

- $1-2-2-3-3-3 \Rightarrow$  Modumuz 3'dür. (*Unimodal*)
- $1-2-2-3-3 \Rightarrow$  Modumuz 2 ve 3'dür. (*Bimodal*)
- $1-2-3 \Rightarrow$  Modumuz 1, 2 ve 3'dür. (*Trimodal*)
- $1-2-3-4 \Rightarrow$  Modumuz 1, 2, 3 ve 4'dür. (*Multimodal*)

### d) Midrange

Veri setindeki minimum değerle maximum değerin toplamının 2'ye bölünmesidir.



Mean, median, and mode of symmetric versus positively and negatively skewed data.

- a) Veri eşit olarak dağılmış. Mean, mode, median eşittir. Genellikle böyle olmaz. Çarpık olur.
- b) Pozitif çarpık data, negatif tarafta daha fazla yoğun. Mean sağ tarafta kalır ve diğerlerinden daha büyütür.
- c) Negatif çarpık data, pozitif tarafta daha fazla yoğun. Mean sol tarafta kalır ve diğerlerinden daha küçütür.

Örnek: Bir sınavda not ortalaması 50 ise simetrik data yorumu yapılabilir. 25 ise pozitif çarpık data yorumları yapılır, 75 ise negatif çarpık data yorumları yapılır.

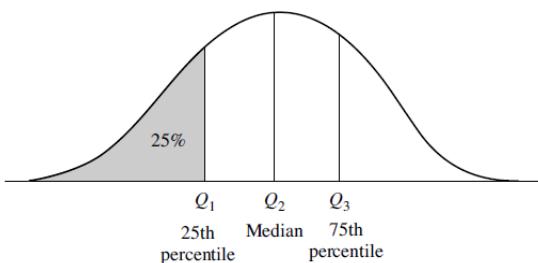
## Measuring the Dispersion of Data: Range, Quartiles, Variance, Standard Deviation, and Interquartile Range

Numeric verilerin yayılımı ve dağılımını incelemek için çeşitli istatistiksel ifadeler kullanılır. Five-number summary olarak ifade edilir ve kutu taslağı/grafiği olarak görüntülenir. Aykırı değerleri tespit etmede de kullanılan bir yöntemdir.

### Range, Quartiles and Interquartile Range (Aralık, Çeyrekler ve Çeyrekler Arası Aralık)

*Range (Açıklık) hesaplamak için:* En büyük eleman ile en küçük eleman arasındaki fark o dizinin range/açıklığını verir.

Örnek: Elimizde sıralı bir veri seti olsun:



Öyle noktalar bulalım ki veriyi ardışık eşit parçalara bölsün. Grafikteki Q noktalarının her biri veriyi 25% oranda bölgerek veri setini 4 parçaya eşit olarak ayırmış.

- Q1: Birinci çeyreklik. Verinin 25%'ini böler.
- Q2: İkinci çeyreklik. Verinin 50%'sini böler. Medyanımızı ifade eder.
- Q3: Üçüncü çeyreklik. Verinin 75%'ini böler.

$$* \underline{IQR \text{ (Interquartile Range)}} = Q_3 - Q_1$$

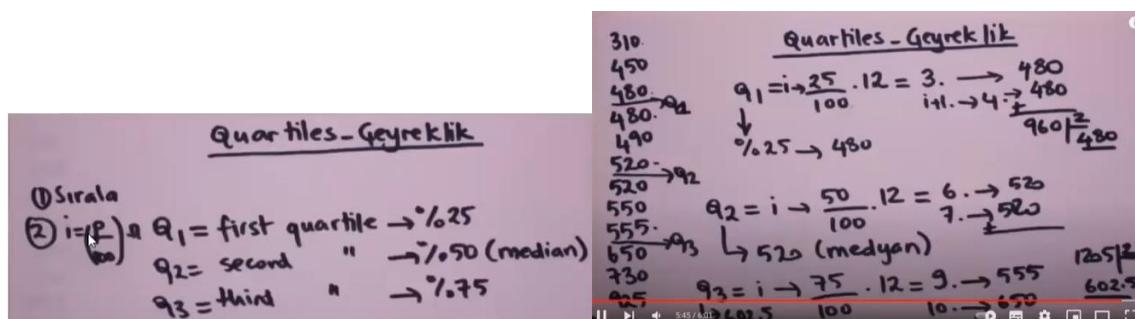
**Problem çözüm adımları:**

1-Veriyi küçükten büyüğe sırala.

$$2- i = (p/100) \times n$$

3- Q1 için p yerine 25, Q2 için p yerine 50, Q3 için p yerine 75 koyulur.

4- Her bir Q için bulunan i. ve (i+1). indisler toplanıp ikiye bölünerek Q değerleri bulunur.

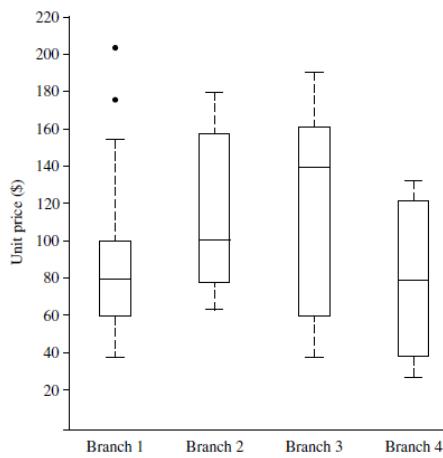


### Five-Number Summary, Boxplots and Outliers

Five-Number Summary: Serinin min değeri ve max değeri, Q1, Q2 ve Q3 değerleri.

Aykırı değerleri bulmamızı sağlayan kural: Q3 (üçüncü çeyreğin) üzerinde veya Q1 (ilk çeyreğin) altında en az  $1.5 \times \text{IQR}$  düşen değerler.

Örnek: Q3 değerimizin 60 olduğu bir veri setinde  $1.5 \times \text{IQR}$  sonucumuz 100 çıktıysa ve verimizde 160. değer var ise bu veri aykırıdır. Q1 değerimizin 70 olduğu bir veri setinde  $1.5 \times \text{IQR}$  sonucumuz 100 çıktıysa 70 geri gidilir ve 30. verimiz aykırı değer olur.



AE firmasının 4 farklı şubesinde satılan ürünlerin birim fiyatları için çizilen bir boxplot.

Büyikler minimum ve maximum değerleri ifade eder.

*Şube 1 için:* Ürün minimum 40\$ maksimum 155\$ satılmış. Birinci çeyrekliği (Q1) 60, ikinci çeyrekliği/medyanı (Q2) 80, Üçüncü çeyrekliği (Q3) 100'dür. IQR değeri  $100 - 60 = 40$ 'dır. Aykırı değerler küçük nokta ile ifade edilen değerlerdir.

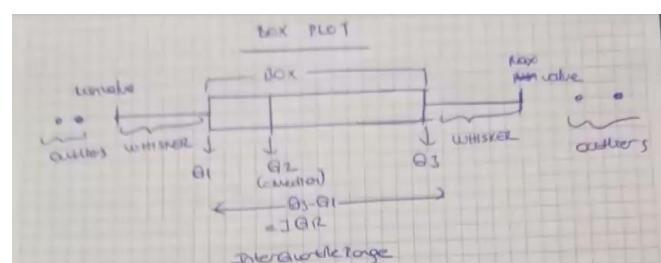
Örnek: Verilen bir veri setinden medyan bulma, çeyreklik ve çeyrekler arası açıklık(IQR) hesaplama, box plot çıkarma işlemi ve outlier/aykırı değer bulma:

$$\begin{aligned}
 & \text{Ex: } 22, 25, 27, 19, 33, 64, 23, 17, 20, 18 \\
 & \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \\
 & \quad 17, \quad 17 \quad 18 \quad 19 \quad 20 \quad \downarrow 22 \quad 23 \quad 25 \quad 33 \quad 64 \\
 & \quad \downarrow \qquad \downarrow \qquad \uparrow \text{Med} \uparrow \quad \frac{20+22}{2} = \frac{42}{2} = 21 \quad \downarrow \quad \text{Outlier} \\
 & \quad \text{Min} \qquad Q_1 \qquad \underbrace{\text{Med}}_{Q_2=21} \qquad \qquad \qquad Q_3 \qquad \downarrow \\
 & \quad N+1 = \frac{9+1}{2} = 5 \qquad \text{Median}
 \end{aligned}$$
  

$$\begin{aligned}
 & N+1 = 5+1 = 6 \quad \text{Check for outliers,} \\
 & 1. \text{ higher outliers} \\
 & \quad = Q_3 + [1.5 \cdot IQR] \rightarrow 25 + [1.5 + 7] \rightarrow \text{any data} > 35.5 \\
 & \quad IQR = 25 - 18 = 7 // \quad = 35.5 \quad \text{higher outlier,} \\
 & 2. \text{ low outliers} \\
 & \quad = Q_1 - [1.5 \cdot IQR] = 18 - [1.5 + 7] \quad \left\{ \begin{array}{l} Q_1 = 8 \\ Q_2 = 21 \\ Q_3 = 25 \end{array} \right. \quad \text{outlier} = 64 // \\
 & \quad = 18 - 10.5 \quad \left\{ \begin{array}{l} \min = 17 \\ \max = 33 \end{array} \right. \\
 & \quad = 7.5 \quad \text{Outliers}
 \end{aligned}$$

Medyan 21 ise Q2 de 21 dir.

low outlier yok.



## Variance and Standard Deviation

Veri yayılım ölçüleridir. Datanın nasıl dağıldığını gösterir. Standart sapma düşük ise, ortalamaya yani medyana yakın verilere sahip olduğumuz yorumlanır.

*Standart Sapma hesaplama formülü:*

The variance of  $N$  observations,  $x_1, x_2, \dots, x_N$ , for a numeric attribute  $X$  is

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 = \left( \frac{1}{N} \sum_{i=1}^N x_i^2 \right) - \bar{x}^2, \quad (2.6)$$

where  $\bar{x}$  is the mean value of the observations, as defined in Eq. (2.1). The standard deviation,  $\sigma$ , of the observations is the square root of the variance,  $\sigma^2$ .

Sigmanın kare kökü standart sapmayı verir.  $X$  üzerindeki çizgi ortalamayı ifade eder.

$\Sigma$  = 0 ise bütün gözlem değerleri aynıdır. Örneğin tüm işçilerin salary değeri aynıdır.

Örnek:

Burada önce çok ufak bir anakütle veri serisi için standart sapma hesaplaması gösterilmektedir. Bu seri bir inşaat firmasının yabancılaraya yaptığı aylık daire satış sayılarını göstermektedir ve veri serisi şudur: { 5, 2, 11, 12, 3, 6 }.

1. Önce bir aritmetik ortalama  $\bar{x}$  şöyle hesaplanır:

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} ..$$

Burada  $i$  her veriye verilen sıra numarasıdır yani  $i=1,2,3,\dots,6$ . Yani

$$x_1 = 5$$

$$x_2 = 2$$

$$x_3 = 11$$

$$x_4 = 12$$

$$x_5 = 3$$

$$x_6 = 6$$

Bu halde  $N = 6$  olup veri büyüklüğü veya anakütle hacmidir.

$$\bar{x} = \frac{1}{6} \sum_{i=1}^6 x_i \quad N \text{ yerine } 6$$

$$\bar{x} = \frac{1}{6} (x_1 + x_2 + x_3 + x_4 + x_5 + x_6)$$

$$\bar{x} = \frac{1}{6} (5 + 2 + 11 + 12 + 3 + 6)$$

$$\bar{x} = 6.5 \quad \text{Bu aritmetik ortalamadır.}$$

2. Standart sapma  $\sigma$  değerini bulma:

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} .$$

$$\sigma = \sqrt{\frac{1}{6} \sum_{i=1}^6 (x_i - \bar{x})^2} \quad N \text{ yerine } 6$$

$$\sigma = \sqrt{\frac{1}{6} \sum_{i=1}^6 (x_i - 6.5)^2} \quad \bar{x} \text{ yerine } 6.5$$

$$\sigma = \sqrt{\frac{1}{6} [(5 - 6.5)^2 + (2 - 6.5)^2 + (11 - 6.5)^2 + (12 - 6.5)^2 + (3 - 6.5)^2 + (6 - 6.5)^2]}$$

$$\sigma = \sqrt{\frac{1}{6} [(-1.5)^2 + (-4.5)^2 + (4.5)^2 + (5.5)^2 + (-3.5)^2 + (-0.5)^2]}$$

$$\sigma = \sqrt{\frac{1}{6} (2.25 + 20.25 + 20.25 + 30.25 + 12.25 + 0.25)}$$

$$\sigma = \sqrt{\frac{85.5}{6}}$$

$$\sigma = \sqrt{14.25}$$

$\sigma = 3.77$  Bu standart sapma değeri olur.

Bu sonucun dikkati çekecek bir yanı verilerin tam sayı olmasına rağmen standart sapmanın (ve aynı şekilde aritmetik ortalamanın) kesirli olmasıdır.

Bu hesaplamayı daha kolaylaştırmak için şu formül kullanılabilir:

$$\sigma = \sqrt{\frac{1}{n} \left( \sum_{i=1}^n x_i^2 - n\bar{x}^2 \right)}.$$

1. Önce bir aritmetik ortalama  $\bar{x}$  hesaplanır:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i .$$

$$\bar{x} = \frac{1}{6} (5 + 2 + 11 + 12 + 3 + 6)$$

$$\bar{x} = 6.5 \quad \text{Bu aritmetik ortalamadır.}$$

2. Sonra toplam kareler bulunur:

$$\sum (x_i)^2 = 5^2 + 2^2 + 11^2 + 12^2 + 3^2 + 6^2$$

$$\sum (x_i)^2 = 25+4+121+144+9+36$$

$$\sum (x_i)^2 = 339$$

3. Bunlar formüle konulur:

Yani  $\sum (x_i)^2 = 339 \quad \bar{x} = 6.5 \quad n = 6$  formüle girer:

$$\sigma = \sqrt{\frac{1}{6} (339 - 6 \times 6.5^2)}$$

$$\sigma = \sqrt{\frac{1}{6} (339 - 253.5)}$$

$$\sigma = \sqrt{\frac{1}{6} (85.5)}$$

$$\sigma = \sqrt{14.25}$$

$\sigma = 3.77$  Bu standart sapma değeridir.

## Graphic Displays of Basic Statistical Descriptions of Data

Bu kısımda bazı temel istatistiksel tanımlamaların grafiksel görüntülerine bakacağız. Bunlar:

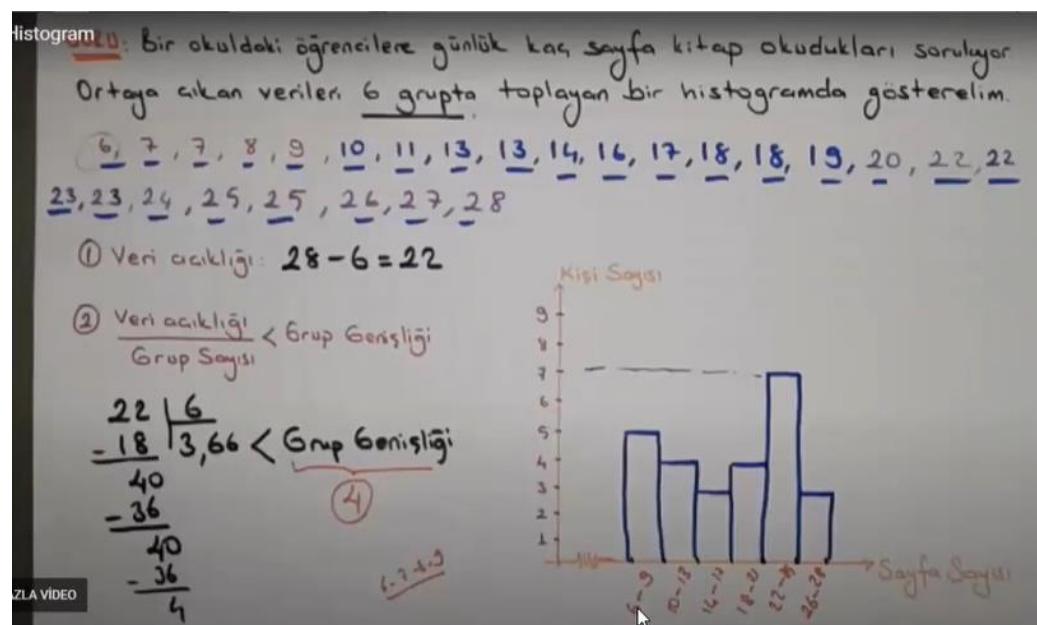
- Quantile plots
- Quantile–quantile plots
- Histograms
- Scatter plots

### a) Histograms

Veri sayısı çok fazla ve bütün verileri grafikte göstermek mümkün değilse veriler gruplandıktan sonra grafiğe aktarılabilir. Bu şekilde oluşturulan sütun grafiğine “histogram” adı verilir. Histogramı çıkarabilmek için sırasıyla aşağıdaki işlemler yapılır:

- Veri grubunun açıklığı bulunur. (En büyük veri – En küçük veri)
- Grup sayısı belirlenir. (Problem içerisinde belirtilir)
- Grup genişliği hesaplanır.
- $[Veri \ Açıklığı / Grup \ Sayısı] < Grup \ Genişliği$

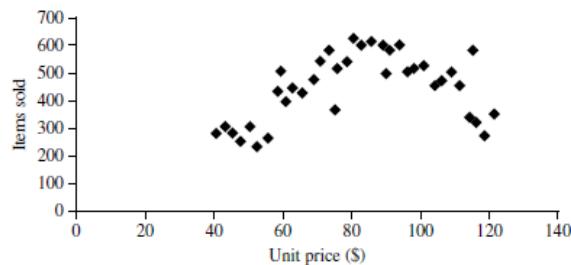
Örnek:



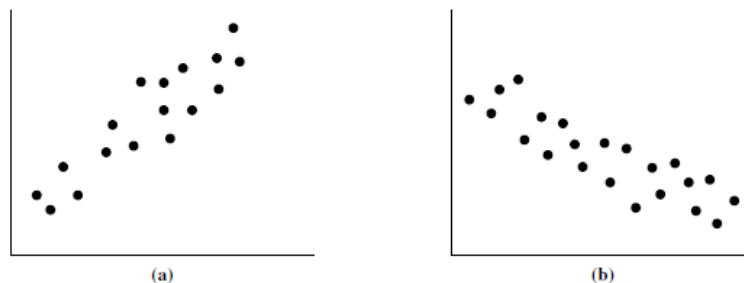
### b) Scatter plots (Saçılım grafiği)

İki attribute arasında bir ilişki varsa bunu tespit etmemizi sağlar. *Correlations* 3 adet değer alabilir:

- *Pozitif* = İki attribute arasında doğru orantılı bir ilişki vardır anlamına gelir.
- *Negatif* = İki attribute arasında ters orantılı bir ilişki vardır anlamına gelir.
- *Sıfır* = İki attribute arası ilişki yok anlamına gelir.



**Figure 2.7** A scatter plot for the Table 2.1 data set.



**Figure 2.8** Scatter plots can be used to find (a) positive or (b) negative correlations between attributes.



**9** Three cases where there is no observed correlation between the two plotted attributes in each of the data sets.

## Measuring Data Similarity and Dissimilarity

Nesneler arasında benzerliği veya farklılığı belirlemek için bazı yöntemler/algoritmalar kullanılır. Bunlardan bazıları:

- Clustering (küme içindekiler benzer dışındakiler değil)
- Outlier analysis (diğerleri ile benzemeyen verileri bulan kümeleme kullanan bir yöntem)
- Nearest-neighbor classification

### Data Matrix versus Dissimilarity Matrix

İki nesnenin birbirine olan benzerliğini bulabilmek için çeşitli uzaklık formüllerinden yararlanırız. Bu kısımda bu formüllerden bahsedeceğiz.

Bir nesnenin tek boyutu var ise “single attribute” olarak tanımlanır. Bu kısımda birden çok özelliği sahip nesneleri inceleyeceğiz.

**Data matrix** (or *object-by-attribute structure*): This structure stores the  $n$  data objects in the form of a relational table, or  $n$ -by- $p$  matrix ( $n$  objects  $\times p$  attributes):

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}. \quad (2.8)$$

“two-mode matrix” olarak ifade edilir.

Satırlar objeyi, sütunlar objenin attribute’unu temsil eder.

**Dissimilarity matrix** (or *object-by-object structure*): This structure stores a collection of proximities that are available for all pairs of  $n$  objects. It is often represented by an  $n$ -by- $n$  table:

$$\begin{bmatrix} 0 & & & & \\ d(2, 1) & 0 & & & \\ d(3, 1) & d(3, 2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n, 1) & d(n, 2) & \cdots & \cdots & 0 \end{bmatrix}, \quad (2.9)$$

“one-mode matrix” olarak ifade edilir.

Nesne  $i$  ve  $j$  arasındaki farklılığı belirtmek için  $d(i, j)$  ifadesini kullanırız. ( $d$  = different)

$i$  ve  $j$  birbirlerine benzerse/yakınsa sıfıra yakın bir değer çıkar. Uzaksa/farklıysa sıfırdan büyük bir değer çıkar.

Satır ve sütunlar objeyi temsil eder.  $n$  tane obje olduğu varsayıılır.

$d(2, 1) \Rightarrow 2.$  nesnenin 1. nesneye olan farklılık/uzaklık değerini temsil eder.

*Neden köşegenler 0?*  $\Rightarrow$  Nesnelerin kendi arasındaki uzaklığı temsil ettikleri için. Örneğin birinci nesnenin birinci nesneye olan uzaklığı 0'dır.  $[d(i, i) = 0]$

### Similarity Matrix

$$sim(i, j) = 1 - d(i, j)$$

### Proximity Measures for Nominal Attributes (Nominal Attribute için fark hesaplama)

Çeşitli attribute’larımız vardı. Bunlardan biri nominal attribute’du. Bu kısımda bu attributeler arasındaki yakınlık/uzaklık ölçüsünü hesaplayacağız.

*hair\_color* bir nominal attribute’dur. Birden fazla değer alabilir. Bu değerleri numerik olarak sembolize edebiliriz. Örneğin red için 1, yellow için 2 ve green için 3’ü kullanabiliriz.

“How is dissimilarity computed between objects described by nominal attributes?”

The dissimilarity between two objects  $i$  and  $j$  can be computed based on the ratio of mismatches:

$$d(i, j) = \frac{p - m}{p}, \quad (2.11)$$

$m \Rightarrow$  Eşleşme (match) sayısıdır. (Yani,  $i$  ve  $j$ ’nin aynı durumda olduğu özniteliklerin sayısı)

$p \Rightarrow$  Nesneleri tanımlayan özniteliklerin (attribute’ların) toplam sayısıdır.

### Örnek-1:

A Sample Data Table Containing Attributes of Mixed Type

Object Identifier	test-1 (nominal)	test-2 (ordinal)	test-3 (numeric)
1	code A	excellent	45
2	code B	fair	22
3	code C	good	64
4	code A	excellent	28

Elimizde yukarıda verilen tablo var. Obj kismı nesneleri temsil eder, attribute olarak kabul edilip hesaplanmaz. Üç adet attribute'umuz var. Simdilik yalnızca Test-1 için konuşalım:

*test-1 nominal attribute tipindedir. test-1 için dissimilarity matrix hesaplama adımları:*

1. Formülü ve tabloyu inceleyelim:

$d(i,j) = \frac{p-m}{p}$				$p = \text{Number of attributes}$
Colour	Code	Grading	Coding	$m = \text{Number of match b/w i & j}$
Blue		A	Code A	1
Red		B	Code B	2
Green		C	Code B	3
Green		A	Code C	4

Example

Id	Test Result			
	1	2	3	4
1	Code A			
2		Code B		
3			Code C	
4				Code A

$\uparrow \text{if I want to know which data is similar or dissimilar}$

2. Nesnelerin yerlerini ve aralarındaki farklılığı ifade eden matrisi kuralım:

What is the dissimilarity between two object				
First, we create dissimilarity matrix.				
1	0			
2	$d(2,1)$	0		
3	$d(3,1)$	$d(3,2)$	0	
4	$d(4,1)$	$d(4,2)$	$d(4,3)$	0

$$\begin{bmatrix} 0 \\ d(2,1) & 0 \\ d(3,1) & d(3,2) & 0 \\ d(4,1) & d(4,2) & d(4,3) & 0 \end{bmatrix}$$

3. d noktalarının değerlerini formülü uygulayarak bulalım:

$$d(2,1) = \frac{1-0}{1} = 1, \quad p=1, m=0$$

$$d(3,1) = \frac{1-0}{1} = 1, \quad p=1, m=0$$

$$d(4,1) = \frac{1-1}{1} = 0, \quad p=1, m=1$$

$$d(4,2) = \frac{1-0}{1} = 1, \quad p=1, m=0$$

$$d(4,3) = \frac{1-0}{1} = 1, \quad p=1, m=0$$

$$d(3,2) = \frac{1-0}{1} = 1, \quad p=1, m=0$$

4. Sonucu değerlendirelim:

All object are dissimilar to each other except 4 and 1,,  
[1]

Örnek 2: İki adet attribute için dissimilarity matrix bulalım:

	Attribute 1	Attribute 2
1	20	AA
2	40	BB
3	20	AA
4	30	CC

	2	3	4
2	0		
3	d(3,1)	d(3,2)	0
4	d(4,1)	d(4,2)	d(4,3)

$$d(2,1) = \frac{2-0}{2} = 1 \quad m=0, p=2$$

$$d(3,1) = \frac{2-2}{2} = \frac{0}{2} = 0 \quad m=2, p=2 \quad (\text{two match})$$

$$d(3,2) = \frac{2-0}{2} = \frac{2}{2} = 1, \quad m=0, p=2$$

$$d(4,1) = \frac{2-0}{2} = \frac{2}{2} = 1 \quad m=0, p=2$$

$$d(4,2) = \frac{2-0}{2} = \frac{2}{2} = 1 \quad m=0, p=2 //$$

	1	2	3	4
1	0			
2	1	0		
3	0	1	0	
4	1	1	1	0

Birinci nesnenin birinciye ve üçüncü nesnenin birinciye olan uzaklığı 0'dır. Yani birinci ve üçüncü nesne hariç kalan nesneler birbirlerinden farklıdır.

### Proximity Measures for Binary Attributes (Binary Attribute için fark hesaplama)

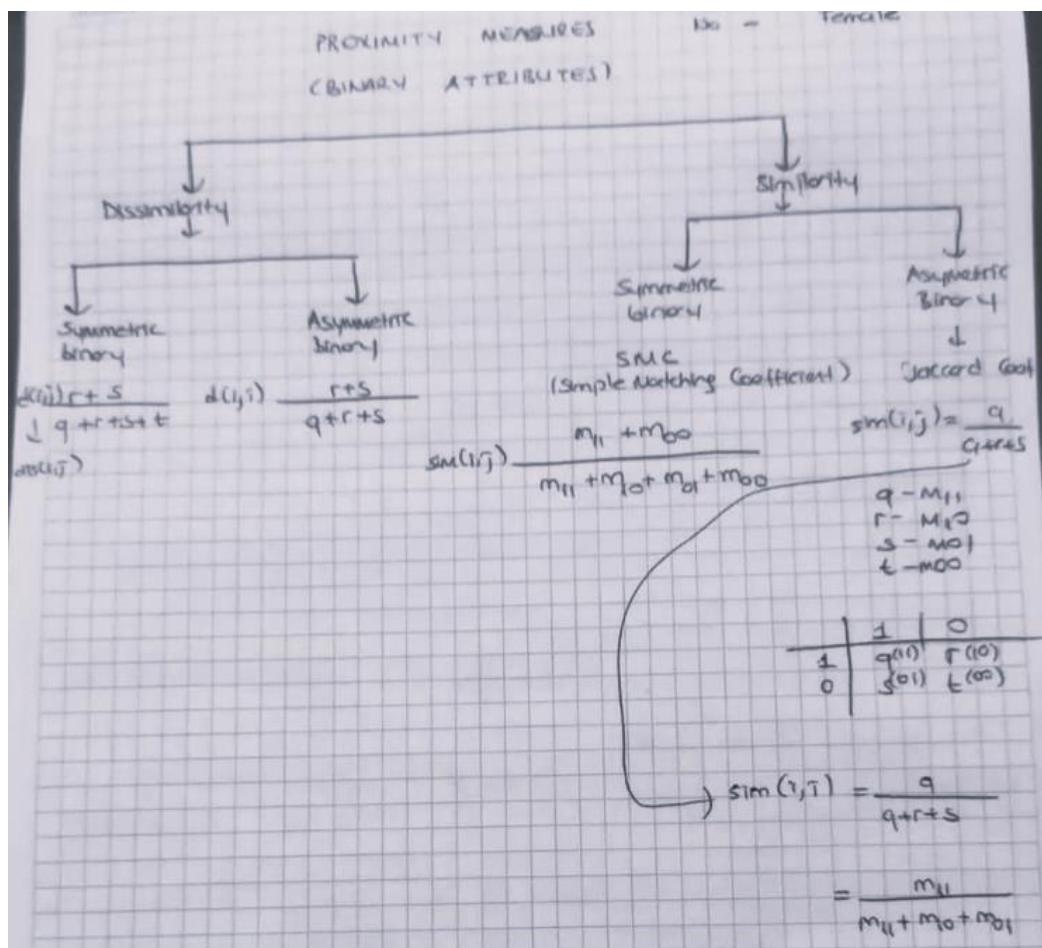
Binary attribtues yalnızca iki değer alır. (Erkek ya da kadın, 0 ya da 1)

İki binary attribute arasındaki farklılığı hesaplama adımlarına bakalım:

		<b>Object <i>j</i></b>	
		1	0
<b>Object <i>i</i></b>	1	<i>q</i>	<i>r</i>
	0	<i>s</i>	<i>t</i>
	sum	<i>q+s</i>	<i>r+t</i>
			<i>p</i>

- $q = i$  ve  $j$ 'nin 1 olma durumu ( $i-j$  arasında 1'e eşit olan attribute sayısı)
- $t = i$  ve  $j$  nin 0 olma durumu
- $r = i$ 'nin 1  $j$ 'nin 0 olma durumu.
- $s = i$ 'nin 0  $j$ 'nin 1 olma durumu.

Formüller:



### Örnek-1:

## Relational Table Where Patients Are Described by Binary Attributes

### Asimetrik data için:

	T <sub>00</sub>	T <sub>01</sub>	T <sub>10</sub>	T <sub>11</sub>	T <sub>12</sub>	T <sub>13</sub>	T <sub>14</sub>	T <sub>15</sub>	T <sub>16</sub>
Jack	1	0	1	0	0	0	0	0	0
Jim	1	0	1	0	1	0	0	0	0
Mary	1	1	0	0	0	0	0	0	0

Let's assume that  
assume.  
we have binary attributes.

we find out, how which  
object similar or dissimilar.

Assume Att. =  $\frac{r+s}{q+r+s} = \frac{m_{10} + m_{01}}{m_{11} + m_{10} + m_{01}}$

→ This is the dissimilarity matrix of given dataset

	Jack	Jim	Mary
Jack	0		
Jim	0.33	0	
Mary	0.67	0.75	0

	Jack	T <sub>00</sub>	T <sub>01</sub>	T <sub>10</sub>	T <sub>11</sub>	T <sub>12</sub>	T <sub>13</sub>	T <sub>14</sub>	T <sub>15</sub>	T <sub>16</sub>
Jack	1	0	1	0	0	0	0	0	0	0
Jim	1	0	1	0	1	0	0	0	0	0

$m_{10} \rightarrow 0$   
 $m_{01} \rightarrow 1$   
 $m_{11} \rightarrow 2$

$d(Jim, Jack) = \frac{0+1}{2+0+1} = \frac{1}{3} = 0.33$

$d(Mary, Jack) = \frac{1+1}{1+1+1} = \frac{2}{3} = 0.67$

$d(Mary, Jim) = \frac{2+1}{1+2+1} = \frac{3}{4} = 0.75$

	Jack	T <sub>00</sub>	T <sub>01</sub>	T <sub>10</sub>	T <sub>11</sub>	T <sub>12</sub>	T <sub>13</sub>	T <sub>14</sub>	T <sub>15</sub>	T <sub>16</sub>
Jack	1	0	1	0	0	0	0	0	0	0
Mary	1	1	0	0	0	0	0	0	0	0

$m_{10} = 1$   
 $m_{01} = 1$   
 $m_{11} = 1$

	Jim	T <sub>00</sub>	T <sub>01</sub>	T <sub>10</sub>	T <sub>11</sub>	T <sub>12</sub>	T <sub>13</sub>	T <sub>14</sub>	T <sub>15</sub>	T <sub>16</sub>
Jim	1	0	1	0	1	0	0	0	0	0
Mary	1	1	0	0	0	0	0	0	0	0

$m_{10} = 2$   
 $m_{01} = 1$   
 $m_{11} = 1$

### Örnek-2: Simetrik data için:

Exam, check that, these data are the symmetric binary -	
Symmetric Binary	Jack   1 0 1 0 0 0 Jim   1 0 1 0 1 0
$d(i,j) = \frac{r+s}{q+r+s+t}$	$m_{10} = 0$ $m_{01} = 1$ $m_{11} = 2$ $m_{00} = 3$
$d(i,j) = \frac{m_{10} + m_{01}}{m_{11} + m_{10} + m_{01} + m_{00}}$	
$d(jim,jack) = \frac{0+1}{2+0+1+3} = \frac{1}{6}$	
$d(jack,mory) = \frac{1+1}{1+1+1+3} = \frac{2}{6} = \frac{1}{3}$	Jack   1 0 1 0 0 0 Mory   1 0 1 0 0 0
$d(mory,jim) = \frac{2+1}{1+2+1+2} = \frac{3}{6} = \frac{1}{2}$	$m_{10} = 1$ $m_{01} = 1$ $m_{11} = 1$ $m_{00} = 3$ Jack   1 0 1 0 0 0 Mory   1 1 0 0 0 0
	Mory   1 0 1 0 0 0 Jim   1 1 0 0 0 0
	Jack   1 0 1 0 0 0 Jim   1 0 1 0 1 0 Mory   1 1 0 0 0 0

### Örnek 2.1:

$x = (1, 0, 0, 0, 0, 0, 0, 0, 0)$	$y = (0, 0, 0, 0, 0, 1, 0, 0, 1)$	{ if this data is symmetric binary . }
<u>SCM</u>		
$sim(x,y) = \frac{0+7}{0+1+2+7} = \frac{7}{10} = 0.7$	<u>SCM</u>	
	$sim(i,j) = \frac{m_{11} + m_{00}}{m_{11} + m_{10} + m_{01} + m_{00}}$	
if this data asymmetric ,		
$x = (1, 0, 0, 0, 0, 0, 0, 0, 0)$		
$y = (0, 0, 0, 0, 0, 0, 0, 0, 1)$		
A Symmetric Binary		
Jaccard Coef = $\frac{1}{q+r+s} = \frac{m_{11}}{m_{11} + m_{10} + m_{01}}$		
$sim(x,y) = \frac{0}{0+1+1} = \frac{0}{2} = 0\%$		

## Proximity Measures for Ordinal Attributes (Ordinal Attribute için fark hesaplama)

Ordinal attribute anlamlı sıraya sahip verilere sahiptir. Öncelikle normalize etmemiz gereklidir. Bunun için şu formül uygulanır:

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}.$$

String veriler numaralar ile sembolize edilerek işlemler yapılmalıdır.

Örnek:

İlk olarak değerleri 0 ve 1 aralığında belirtmek için normalizasyon yaparız:

Subject		PROXIMITY MEASURE (ORDINAL ATTRIBUTES)		
Data Matrix				
Object Identifier	TEST 2 (ordinal)			
1	Excellent	-3		STEP 1) count states, here=3 (=mf) (Fair, Good, Excellent)
2	Fair	-1		STEP 2) replace each ordinal data after by rank
3	Good	2		Fair → 1, Good → 2, Excellent → 3,
4	Excellent	-3		STEP 3) Normalize the ranking $Z_{if} = \frac{R_{if} - 1}{mf - 1}$

we have four object - we have to normalize

1	Exc	Exc	Exc	1
2	Fair	Fair	Fair	0
3	Good	Good	Good	0.5
4	Exc	Exc	Exc	1

$d_{(x,4)} = \frac{3-1}{3-1} = \frac{0}{2} = 0$   $\text{R}_{if}$  ranking of attr.

$d_{(x,2)} = \frac{2-1}{3-1} = \frac{1}{2} = 0.5$

$d_{(x,3)} = \frac{3-1}{3-1} = \frac{2}{2} = 1$

İkinci olarak uzaklık hesaplanır (manhattan pist formülü ile):

$$\text{Manhattan Dist} = |x_1 - y_1| + |x_2 - y_2| + \dots + |x_n - y_n|$$

$$d(x,4) =$$

Distance Matrix		
	1 2 3 4	
1	0	
2	1 0 0	
3	0.5 0.5 0	
4	0 1 0.5 0	

$d(2,1) = |0-1| = 1$

$d(3,1) = |0.5-1| = 0.5$

$d(4,1) = |1-1| = 0$

$d(4,2) = |1-0| = 1$

$d(4,3) = |1-0.5| = 0.5$

Ordinal tipte datanız var ise dissimilarity hesabı için şu formül kullanılır:  $\text{sim}(i, j) = 1 - d(i, j)$

## Dissimilarity of Numeric Data: Minkowski Distance

Numeric attribute tanımlı objelerin farkını bulmak için:

- Euclidean
- Manhattan
- Supressum Distance

formüllerinden yararlanacağız.

The most popular distance measure is **Euclidean distance** (i.e., straight line or “as the crow flies”). Let  $i = (x_{i1}, x_{i2}, \dots, x_{ip})$  and  $j = (x_{j1}, x_{j2}, \dots, x_{jp})$  be two objects described by  $p$  numeric attributes. The Euclidean distance between objects  $i$  and  $j$  is defined as

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{ip} - x_{jp})^2}. \quad (2.16)$$

Another well-known measure is the **Manhattan (or city block) distance**, named so because it is the distance in blocks between any two points in a city (such as 2 blocks down and 3 blocks over for a total of 5 blocks). It is defined as

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|. \quad (2.17)$$

Both the Euclidean and the Manhattan distance satisfy the following mathematical properties:

**Non-negativity:**  $d(i, j) \geq 0$ : Distance is a non-negative number.

**Identity of indiscernibles:**  $d(i, i) = 0$ : The distance of an object to itself is 0.

The **supremum distance** (also referred to as  $L_{max}$ ,  $L_\infty$  norm and as the Chebyshev distance) is a generalization of the Minkowski distance for  $h \rightarrow \infty$ . To compute it, we find the attribute  $f$  that gives the maximum difference in values between the two objects. This difference is the supremum distance, defined more formally as:

$$d(i, j) = \lim_{h \rightarrow \infty} \left( \sum_{f=1}^p |x_{if} - x_{jf}|^h \right)^{\frac{1}{h}} = \max_f^p |x_{if} - x_{jf}|. \quad (2.19)$$

The  $L^\infty$  norm is also known as the *uniform norm*.

Örnek:

	x	y
P1	0	2
P2	2	0
P3	3	1
P4	5	1

Elimizde 1 veri seti, 4 adet de objemiz var. (p1-p2-p3-p4)

Her objeye iki adet attribute tanımlanmış. (x ve y)

	P1	P2	P3	P4	P5
P1	0				
P2	2,8	0			
P3	3,2	1,4	0		
P4	5,1	3,2	2,0	0	

Tabloyu oluşturduk köşegenleri 0 olarak belirttik. (kalan değerleri şuan için boş varsayıyalım)

$$d(P_2, P_1) = \sqrt{(2-0)^2 + (0-2)^2} = \sqrt{2^2 + 2^2} = \sqrt{8} = 2\sqrt{2}$$

$$d(P_3, P_1) = \sqrt{(3-0)^2 + (1-2)^2} = \sqrt{3^2 + 1^2} = \sqrt{9+1} = \sqrt{10} = 3,16$$

$$d(P_4, P_1) = \sqrt{(5-0)^2 + (1-2)^2} = \sqrt{5^2 + 1^2} = \sqrt{26} = 5,1$$

Euclidean ile noktaları bulduk.

2) Manhattan

$$d(P_2, P_1) = |2-0| + |0-2| \\ = 2+2 \\ = 4//$$

$$d(P_3, P_1) = |3-0| + |1-2| \\ = 3+1 = 4//$$

$$d(P_4, P_1) = |5-0| + |1-2| \\ = 5+1 \\ = 6//$$

$$d(P_3, P_2) = |3-2| + |0-1| \\ = 1+1 = 2//$$

$$d(P_4, P_3) = |5-3| + |1-1| \\ = 2+0 = 2//$$

Manhattan ile noktaları bulduk.

	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>
P <sub>1</sub>	0			
P <sub>2</sub>	4	0		
P <sub>3</sub>	4	2	0	
P <sub>4</sub>	6	4	2	0

Manhattan tablosu

## SUPPRESSUM DISTANCE

$d(P_0, P_1) = \max \{  2-0 , 10-21 \}$	$\frac{P_2}{P_3}$
$= \max \{ 2, 2 \}$	$\underline{\underline{P_4}}$
$= 2,$	
$d(P_3, P_1) = \max \{  3-0 ,  1-2  \}$	
$= \max \{ 3, 1 \}$	
$d(P_3, P_2) = \max \{  3-2 ,  1-0  \} = \max \{ 1, 1 \}$	
$d(P_4, P_1) = \max \{  5-0 ,  1-2  \}$	
$= \max \{ 5, 1 \}$	
$= 5,$	
$d(P_4, P_2) = \max \{  5-2 ,  1-0  \}$	
$= \max \{ 3, 1 \}$	
$= 3,$	

	A1	A2
P1	0	2
P2	2	0
P3	3	1
P4	5	1

	P1	P2	P3	P4
P1	0	1		
P2	2	0		
P3	3	1	0	
P4	5	3	2	0

### Dissimilarity for Attributes of Mixed Types

Bir veri seti içerisinde birden çok farklı tipte attribute içerebilir. Genelde bir veri seti böyle olur. Öncelikle tüm attribute tipleri için her attribute'a uygun formül ile değerler bulunur ve bu değerler ile karışık veri setimiz için nesneler arası farklılık formülümüzü uygularız:

$$d(i, j) = \frac{\sum_{f=1}^p \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^p \delta_{ij}^{(f)}},$$

Örnek:

Wow, solve numerical attribute

Data Matrix			
	Nominal	Ordinal	Numerical
i	TEST 1	TEST 2	TEST 3
1	CodeA	Excellent	65
2	CodeB	Fair	22
3	CodeC	Good	64
4	CodeD	Excellent	28

j	TEST 1	TEST 2	TEST 3
1	1	45	
2	2	22	
3	3	64	
4	4	28	

Dissimilarity Matrix of Nominal

①

	1	2	3	4
1	0			
2	1	0		
3	1	1	0	
4	0	1	1	0

② Dissimilarity matrix of ordinal

	1	2	3	4
1	0			
2	1	0		
3	0.5	0.5	0	
4	0	1.0	0.5	0

$d(4,3) = \frac{1}{2}(28 - 64)$

$d(4,3) = \frac{1}{2}(64 - 22)$

real

How to normalize these values,  
so that can be mapped [0,1]

Use Formula

$$d_{ij} = \frac{|x_{if} - x_{jf}|}{max - min}$$

$\underline{=}$

$\rightarrow$  There is no difference between  
some objects

		1	2	3	4	
		1	0			
diral	1	0				
	2	0.55	0			
	3	0.45	1	0		
	4	0.40	0.4486	0		

$$d(2,1) = \frac{|64 - 22|}{64 - 22} = \frac{23}{42} = 0.55$$

$$d(3,1) = \frac{|64 - 45|}{64 - 22} = 0.45$$

$$d(4,1) = \frac{|28 - 45|}{64 - 22} = \frac{17}{42} = 0.40$$

$$d(4,2) = |28 - 22|$$

How to put all together (mixed)

Formula for mixed attribute

$$d(i,j) = \frac{\sum_{f=1}^P \delta_{if} d_{if}}{\sum_{f=1}^P \delta_{if}}$$

$\delta_{if} = 0$ , if  $x_{if}$  or  $x_{jf}$  missing  
 or  $x_{if} = x_{jf} = 0$  and f is  
 asymmetric binary  
 $\delta_{if} = 1$  otherwise.  
 $\sum \delta_{if} = 1$ ,  $\rightarrow$  because there is no missing value between f and i.

$$d(2,1) = \frac{(1*1) + (1*1) + (1*0.55)}{1+1+1} = \frac{2+0.55}{3} = \frac{2.55}{3} = 0.85$$

$$d(3,1) = \frac{(1*1) + (1*0.5) + (1*0.45)}{1+1+1} = 0.65$$

dc

		1	2	3	4	
		1	0			
dc	1	0				
	2	0.55	0			
	3	0.45	0.4486	0		
	4	0.40	0.71	0.79	0	

# **Chapter 3: Data Preprocessing**

## **□ Data Preprocessing: An Overview**

- Data Quality: Why Preprocess the Data?
- Major Tasks in Data Preprocessing

## **□ Data Cleaning**

- Missing Values
- Noisy Data

## **□ Data Integration**

- Entity Identification Problem---
- Redundancy and Correlation Analysis
- Tuple Duplication
- Data Value Conflict Detection and Resolution

## **□ Data Reduction**

- Overview of Data Reduction Strategies
- Wavelet Transforms
- Principal Components Analysis
- Attribute Subset Selection
- Regression and Log-Linear Models: Parametric Data Reduction
- Histograms
- Clustering
- Sampling
- Data Cube Aggregation

## **□ Data Transformation and Data Discretization**

- Data Transformation Strategies Overview
- Data Transformation by Normalization
- Discretization by Binning
- Discretization by Histogram Analysis
- Discretization by Cluster, Decision Tree, and Correlation Analyses
- Concept Hierarchy Generation for Nominal Data

## Data Preprocessing: An Overview

### Data Quality: Why Preprocess the Data?

Günümüz veri tabanları gürültülü, kayıp ve uyumsuz datalara karşı hassastır. Bu tür datalara “*low quality data*” denir. Düşük kaliteli bir dataya veri madenciliği teknikleri uygulandığında istediğimiz sağlam ve güvenilir modeli elde edemeyiz. Preprocessing işlemi low-quality datanın nasıl high quality hale getirileceğini inceler ve uygular.

Data preprocessing için dört yöntem/teknik mevcuttur:

- *Data cleaning* (Verideki gürültü ve uyumsuzluk giderilir)
- *Data integration* (Birçok farklı kaynaktan gelen veri datawarehouse’da birleştirilir.)
- *Data Reduction* (Verinin fazlalıklarını yok ederek veya kümleme ile size/boyutunu azaltır.)
- *Data transformations* (Datanın daha küçük aralığa sahip olunması sağlanır. --Normalizasyon)

Bir işlemde birden fazla data preprocessing uygulaması yapılabilir. Veri temizleme yöntemi yanlış datayı temizlemek için aynı zamanda data transformation yöntemi de içerebilir. Örneğin doğum tarihlerini yaygın formata çevirme işlemi data transformation ve eski formatı sildiği için de bir data cleaning uygulamasıdır.

Veriyi kaliteli hale getirmeden önce veriyi tanımak gereklidir. Bu sebeple data preprocessing yapmadan önce datayı tanıtmamız gereklidir. (chapter-2 de bu kısımdan bahsettiğimizde)

Veri kalitesini oluşturan/tanımlayan faktörler:

- Accuracy (Doğruluk)
- Completeness (Tam olması)
- Consistency (Tutarlılık)
- Timeliness (Zamanlılık)
- Believability (Güvenilebilirlik)
- Interpretability (İnanabilirlik)

Şirketinizin veri ambarını incelemek zorundasınız. Eksik, alışılmadık, gürültülü, tutarsız ve hatalı veriler olabilir. Bazı kayıtların attribute’ları olmayabilir. Gerçek dünyaya hoşgeldiniz... Gerçek dünya dataları *kırılgan*, eksik ve tutarsız olmaya yatkındır. Preprocessing tekniklerini bilip uygulamak bu noktada büyük önem taşiyacaktır. Bu teknikler verinin kalitesini yükseltir.

## Major Tasks in Data Preprocessing

*Data cleaning*, datalardaki tutarsızlığı çözümlemeli, aykırı değerleri tespit etmeli veya yok etmeli, gürültülü değerleri gidermeli ve kayıp değerleri doldurmalıdır.

Veri madenciliği sonuçlarını tehlkiye atmaksızın veri kümemizin size/boyutunu azaltabilir miyiz? Verilerimiz daha küçük ve açıklayıcı attribute'lara sahip olabilir mi? Evet... *Data reduction* kısmı bu sorulara cevap veriyor.

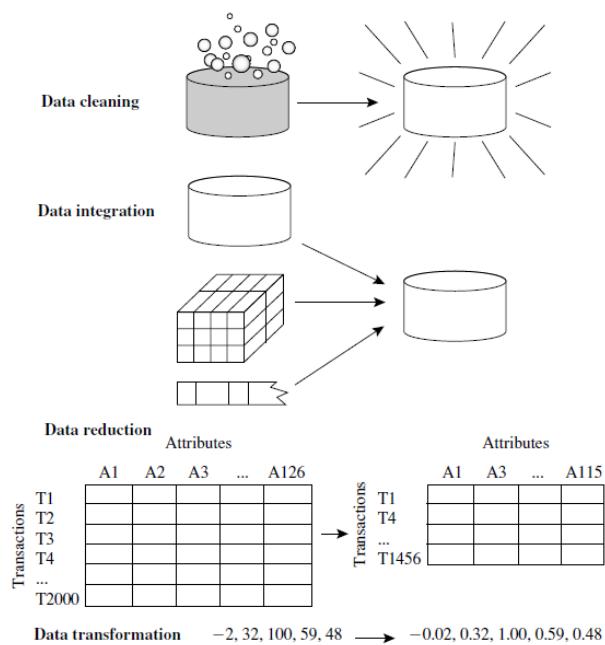
Data reduction iki kısma ayrılır:

- Dimensionality reduction

Boyun indirgemedede, orijinal verilerin küçültülmüş veya "sıkıştırılmış" bir temsilini elde etmek için veri kodlama şemaları uygulanır. Örnekler arasında *veri sıkıştırma teknikleri* [örneğin, dalgaçık dönüşümleri (wavelet transform) ve temel bileşenler analizi (pca)], *öznitelik alt kümese seçimi* (örneğin, ilgisiz niteliklerin kaldırılması) ve *öznitelik oluşturma* (örneğin, orijinal kümeden daha kullanışlı özniteliklerin küçük bir kümесinin türetildiği yer) yer alır.

- Numerosity reduction

Sayısallığın azaltılmasında, veriler parametrik modeller (regression...) veya parametrik olmayan modeller (histograms, clusters, sampling, data aggregation...) kullanılarak daha küçük alternatifleri ile değiştirilir.



Forms of data preprocessing.

- *Data cleaning* => Kirli olan veri temizlendi.
- *Data integration* => Farklı tipteki veriler aynı çatı altında birleşti.
- *Data reduction* => Her bir kayıttan 126 tane attribute (A) var. Data reduction teknikleriyle verinin size/boyutu küçültüldü. (A126 to A115)
- *Data transformation* => Veriler normalizasyon gibi tekniklerle daha küçük belirli bir aralığa çekildi.

## Data Cleaning

Aykırı değerleri tespit ederek ve tutarsız dataları düzelterek verideki kayıp değerleri doldurur ve gürültüleri giderir. Bu bölümde veri temizleme için uygulanan methodlardan bahsedeceğiz.

### Missing Values (Kayıp değerler)

Müşteridatalarını analiz etmemiz gereklidir. Birçok kayıtta birçok attribute kayıp olabilir. Aşağıdaki metodlar ile bu değerleri bulup düzeltebiliriz:

**1. Ignore the tuple:** Genelde sınıf etiketi olmadığı zaman yapılır. Bir data birçok attribute'a sahip ise içerisinde kayıp olan bazı attribute'ları görmezden gelebiliriz. Ancak bir data yalnızca birkaç attribute'a sahip ise bu metodu kullanmak uygun olmayacağından emin olunmalıdır.

**2. Fill in the missing value manually:** Kayıp attribute'u el ile doldur. Zaman alıcı ve genellikle uygun olmayan bir metodudur.

**3. Use a global constant to fill in the missing value:** Eksik değeri doldurmak için global bir sabit kullanılır. Tüm kayıp attribute'lar aynı sabit değer ile değiştirilir. Örneğin müşteri gelirleri veri setindeki kayıp değerler "unknown" olarak değiştirilir. Bu metod da bir çözüm sunmaz.

**4. Use a measure of central tendency for the attribute (the mean or median) to fill in the missing value:** Kayıp değeri doldurmak için merkezi eğilim ölçütleri kullanılır. Simetrik dağılım var ise (mean, median ve mode çakışık ise) mean, çarpık dağılım var ise median uygula ve kayıp tüm değerleri bu değerler ile doldur.

**5. Use the attribute mean or median for all samples belonging to the same class as the given tuple:** Verilen kayıtta bütün örnekler aynı class'a sahip ise mean ve medyanı kullanmamız istenir. Yani medyan ve mean hesaplanıp bu değerleri kayıp değerlere yerleştirebilmek için tüm özelliklerin aynı class'a sahip olduğunu bilmemiz istenir.

**6. Use the attribute mean or median for all samples belonging to the same class as the given tuple:** Kayıp değerler şu yöntemi veya yöntemleri kullanarak bulunur:

- Regression
- Bayesian formülü kullanan inference-based tools'lar
- Decision tree induction (kayıp olmayan değerleri kullanarak tahmin etmemizi sağlar)

## Sonuç

Burada altıncı olan metodumuz eksik değerleri bulup gidermek için oldukça uygunudur. Diğer metodlar ile karşılaştırıldığında daha sağlıklı bir yöntemdir.

## Noisy Data

"Noise/Gürültü: Ölçülebilen bir değişkende bir varyansın rastgele bir hata olduğunu söylemesi.

Chapter-2 de box-plot ve scatter plots gibi temel istatistik tanımlarından bahsetmiştik. Bu tekniklerle aykırı değerler tespit edilebiliyordu.

Örneğin price gibi bir numeric attribute'a sahip olalım. Burada bir gürültü tespit ettik ve bu gürültüyü nasıl giderebiliriz? *Veri düzeltme teknikleri* ile...

### **Veri düzeltme teknikleri (Data Smoothing Techniques)**

**a) Binning:** Sıralanmış veri değerlerini komşularının değerlerine danışarak düzeltir. Veri sıralandıktan sonra Binning tekniği uygulamak için 3 adet yöntem vardır: *Mean*, *Median* ve *Boundaries*.

Sorted data for *price* (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34

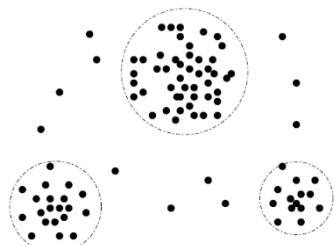
Partition into (equal-frequency) bins:
Bin 1: 4, 8, 15
Bin 2: 21, 21, 24
Bin 3: 25, 28, 34
Smoothing by bin means:
Bin 1: 9, 9, 9
Bin 2: 22, 22, 22
Bin 3: 29, 29, 29
Smoothing by bin boundaries:
Bin 1: 4, 4, 15
Bin 2: 21, 21, 24
Bin 3: 25, 25, 34

1. Veri seti eşit olarak bin'lere ayrıldı.
2. Her bin için ortalama değerler (means) yazıldı. (medyan da olabilir)
3. Her bir bin değeri en yakın boundry değeri ile değişir. (min ve max değerler tespit edildikten sonra bin içerisindeki her bir değere min veya max değerinden hangisine yakın olduğu sorulur ve bu cevaba göre min veya max değerler yerleştirilir.)

### **b) Regression:**

- Linear regression
- Multiple linear regression

**c) Outlier Analysis:** Clustering ile aykırı değerler bulunur. Bu kısımda kümelemeye dahil olamayan veriler oluşur ve bu veriler aykırı olarak tanımlanır:



## Data Integration

Birçok veri kaynaklarından verinin birleşimini sağlar. Fazlalık ve tutarsızlıklarını azaltıp önleme-ye yardımcı olur. Modeldeki doğruluk ve hız artmış olur.

### Entity Identification Problem

Farklı kaynaktan gelen datalar nasıl eşleştirilebilir? İşte bu bir varlık tanıma problemi olarak ifade edilir.

*Soru:* Bilgisayar bir veri tabanından gelen cust\_number ile başka bir veri tabanından gelen customer\_Id ‘nin aynı olduğunu anlayıp nasıl eşleştirme yapabilir?

*Cevap:* Her bir attribute’un bir metadatosu olmalı. Her **metadata** her bir attribute için ayrıntılı bilgi saklar. Bu bilgilere bakılarak eşleşme sağlanır.

### Redundancy (Fazlalık) and Correlation Analysis

Bazı attribute’lar diğer attribute’lardan türetilirse “fazlalık” oluşabilir. Örneğin yıllık gelir gibi.

Bazı durumlarda veri setinde veya attribute’larda tutarsızlık oluşabilir. Örneğin bir veri kümesinde student\_Id varken diğer veri kümesinde bu attribute std\_Id olarak tanımlanmış olabilir. Attribute’ların kendi içerisinde de tutarsızlıklar olabilir. Örneğin format farklılığı gibi.

Tutarsızlığı nasıl tespit edebiliriz?

“Korelasyon Analizi” ile bu “fazlalıklar” tespit edilebilir. İki attribute arasında birinin diğerini ne kadar ve nasıl etkilediğini bulmak için çeşitli korelasyon analizlerinden yararlanılır.

Nominal datalar için  $\chi^2$  (chi-square) testi, Numeric datalar için ise korelasyon katsayısı kullanılarak bu tutarsızlıklar ve fazlalıklar tespit edilebiliyor.

#### a) $\chi^2$ Correlation Test for Nominal Data

*Frequence / Contingency Table:* İki attribute’un aynı anda veri kümesinde bulunma sayısını ifade eder.

Example 2.1’s  $2 \times 2$  Contingency Table Data

	male	female	Total
fiction	250 (90)	200 (360)	450
non_fiction	50 (210)	1000 (840)	1050
Total	300	1200	1500

*Note: Are gender and preferred\_reading correlated?*

1500 kişiye fiction or non\_fiction soruları soruldu ve cevapları cinsiyetleri ile beraber kaydedildi. Toplam 1500 satır oluştı.

1200 Kadından 200’ü fiction cevabını vermiş. Kalan 300 kişi erkek ve 250’si fiction cevabını vermiş.

Toplam 450 ficiton, 1050 non\_fiction cevabı verilmiş.

“Total” bir attribute olarak alınmadığı ve sonradan eklendiği için 2x2’lik bir tablodur/matrictir.

*Chi Square Ana Formül:*

$$\chi^2 = \sum_{i=1}^c \sum_{j=1}^r \frac{(o_{ij} - e_{ij})^2}{e_{ij}},$$

Önce ilk sütunun satırları işleme sokulur. [male sütunun fiction (250) ve non-fiction'ları (50) ]  
Ardından ikinci sütunun satırları işleme sokulur.

c = sütun değerleri

*Observe Frequency (O<sub>ij</sub>)*= Ai ve Bj'nin ortak olma durum sayısı (250 ve sonrasında 50)

*Expected Frequency (E<sub>ij</sub>)*=

$$e_{ij} = \frac{\text{count}(A = a_i) \times \text{count}(B = b_j)}{n},$$

A -> Gender

ai -> A'nın aldığı herhangi bir attribute

B-> Tercih edilen

bj-> fiction

$$e_{11} = \frac{\text{count}(male) \times \text{count}(fiction)}{n} = \frac{300 \times 450}{1500} = 90,$$

11-> Birinci satırın birinci sütunu

Bu fonksiyon O.Frequency'nin (250) E.Frequency'lerini bulmamızı sağlar.

Sütuna bakıyorsak sütun, satıra bakıyorsak satır toplamı alınıyor.

H0 hipotezi: İki attribute birbirinden bağımsız.

Özgürlik Derecesi = (row-1) x (column-1)

Alpha Derecesi = 0.05

*Tablonun Chi Square Hesaplaması Örneği:*

$$\begin{aligned} \chi^2 &= \frac{(250 - 90)^2}{90} + \frac{(50 - 210)^2}{210} + \frac{(200 - 360)^2}{360} + \frac{(1000 - 840)^2}{840} \\ &= 284.44 + 121.90 + 71.11 + 30.48 = 507.93. \end{aligned}$$

(O<sub>ij</sub>-E<sub>ij</sub>)<sup>2</sup>/E<sub>ij</sub>

Bu değerler ile iki attribute'un birbirinden bağımsız olup olmadığını nasıl anlarız?

Chi-Square Dağılım Tablosuna bakılır.

Özgürlik Derecesi 1 ise 1.satırın Alpha Derecesi ile kesiştiği yere bakılır.

Table of chi-squared distribution

Significance level Alpha	0.995	0.975	0.2	0.1	0.05	0.025	0.02	0.0
<i>Degrees of freedom</i>								
1	0	0.001	1.642	2.706	3.841	5.024	5.412	6.6
2	0.01	0.051	3.219	4.605	5.991	7.378	7.824	9.2
3	0.072	0.216	4.642	6.251	7.815	9.348	9.837	11.

Elde edilen chi-square değeri (507) kesişen nokta değerinden büyük olduğu için H<sub>0</sub> durumu olmaz. Yani iki attribute birbirine bağımlıdır. Böylece fazlalık bulunmuş olur. Eğer küçük olsaydı bağımsız olurdu. Fazlalık söz konusu olmazdı.

### b) Correlation Coefficient for Numeric Data

Nominal attribute'lar için "chi-square" testi kullanılırken, Numeric attribute'lar için ise "korelasyon katsayıısı" kullanılır.

$$r_{A,B} = \frac{\sum_{i=1}^n (a_i - \bar{A})(b_i - \bar{B})}{n\sigma_A\sigma_B} = \frac{\sum_{i=1}^n (a_i b_i) - n\bar{A}\bar{B}}{n\sigma_A\sigma_B},$$

r= Korelasyon kat sayısı

A ve B = İncelenen attribute'lar

Sonuç daima -1 ile +1 arasında çıkmalı.

Sonuç 0 dan büyük ise A ve B pozitif korelasyondur, yani A'nın değeri B arttıkça artıyor.

Sonuç 0 dan küçük ise A ve B negatif korelasyondur, yani A ile B arasında zıt bir ilişki vardır.

Sonuç 0 ise A ve B birbirinden bağımsızdır. Kaldırılamaz, kullanmamız gereklidir.

Değer yükseldikçe yani 1'e yaklaşıkça iki attribute arası ilişki bağlılığı yüksek olur. Bu durumda iki attribute'dan birini çıkarsak da olur.

Değer azaldıkça yani -1'e yaklaşıkça iki attribute arası ilişki bağlılığı zayıf olur. Bu durumda iki attribute'dan biri yine çıkarılabilir çünkü arasındaki bağlılık az.

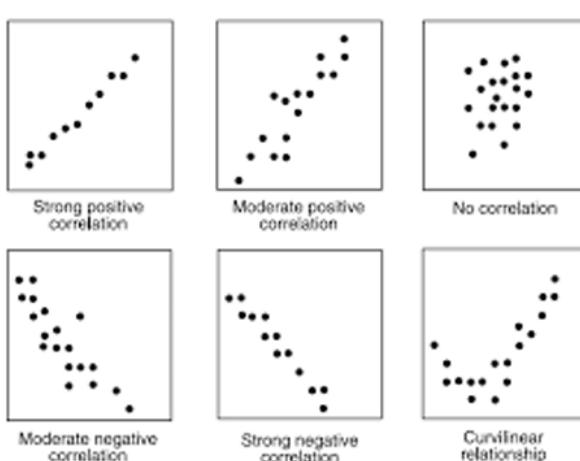
**There are three types of correlation: positive, negative, and none (no correlation).**

- Positive Correlation: as one variable increases so does the other. ...
- Negative Correlation: as one variable increases, the other decreases. ...
- No Correlation: there is no apparent relationship between the variables.

So, we plot data from two variables x and y and then we look at the pattern and see which of the following graph can be the best match then we have an evidence or lack of any relationship.

In the below image the first two from top right suggest a positive linear relationship that we call positive correlation, the top right shows no pattern therefore there is no correlation between the 2 variables

The two graphs from bottom left shows a negative correlation meaning as one increases the other one decreases and finally the bottom right suggests a curve type of relationship first decreases and then increases.



### c) Covariance of Numeric Data

İki attribute'un birbiri ile ne kadar değişebildiğini kovaryans ile ölçeriz.

İki numeric attribute olsun A ve B.

$$E(A) = \bar{A} = \frac{\sum_{i=1}^n a_i}{n} \quad E(B) = \bar{B} = \frac{\sum_{i=1}^n b_i}{n}.$$

A için (veya B) Beklenen Değer yani A'nın ortalama değeri, A attribute'u altındaki tüm değerlerinin toplamının gözlem sayısına bölümüdür.

\*A ve B arasındaki Kovaryansı Bulma:

$$\text{Cov}(A, B) = E((A - \bar{A})(B - \bar{B})) = \frac{\sum_{i=1}^n (a_i - \bar{A})(b_i - \bar{B})}{n}.$$

\*Kovaryans'ın sigmaların çarpımına bölümü **Korelans Katsayıısı**'nı verir:

$$r_{A,B} = \frac{\text{Cov}(A, B)}{\sigma_A \sigma_B},$$

Örnek:

Stock Prices for AllElectronics and HighTech

Time point	AllElectronics	HighTech
t1	6	20
t2	5	10
t3	4	14
t4	3	5
t5	2	5

Time point hesaplamaya katılmaz.

AllElectronics ve HighTech attribute'lari arasındaki kovaryansı bulalım:

1) Ortalamalar alınır:

$$E(\text{AllElectronics}) = \frac{6 + 5 + 4 + 3 + 2}{5} = \frac{20}{5} = \$4$$

$$E(\text{HighTech}) = \frac{20 + 10 + 14 + 5 + 5}{5} = \frac{54}{5} = \$10.80.$$

2) Kovaryans bulma formülü kullanılır:

$$\begin{aligned} \text{Cov}(\text{AllElectronics}, \text{HighTech}) &= \frac{6 \times 20 + 5 \times 10 + 4 \times 14 + 3 \times 5 + 2 \times 5}{5} - 4 \times 10.80 \\ &= 50.2 - 43.2 = 7. \end{aligned}$$

$$[(6-4) \times (20-10.8)] + [(5-4) \times (10-10.8)] + \dots / 5 = 7$$

Sonuç: Pozitif kovaryans mevcut. Strong rel var. İki attribute birbirine bağımlı. Çıkartılabilir.

## **Tuple Duplication**

Attributelar arasındaki fazlılık tespitine ek olarak, aynı kaydın birden fazla olma durumudur. Bir veri kümesinde aynı kayıtların birden fazla olması durumudur.

## **Data Value Conflict Detection and Resolution**

Bir üniversitede A-F veya 1-10 notlandırma sistemi olabilir. İki farklı sistemi kullanan iki üniversitenin bu verilerini aldığımızda bir farklılıkla karşılaşırız. Tutarlılık söz konusu olur.

## **Data Reduction**

Veri azaltmaya neden ihtiyaç duyulur? İncelenenek veri kümesi çok büyük olduğundan dolayı bu büyük miktardaki verilerin analizi uzun bir zaman alabilir. Bu sebeple Data Reduction teknigi ile orijinal datanın bütünlüğünü koruyarak veriyi küçültmemiz gereklidir.

### **Overview of Data Reduction Strategies**

İncelenmekteden olan rasgele değişkenlerin veya attribute'ların sayısını azaltma yöntemidir.

Veri azaltım (Data Reduction) stratejileri 3'e ayrılır:

#### **1) Dimensionality Reduction:**

a) *Wavelet transform*

b) *Principal components analysis (pca):*

Orijinal datanın daha küçük bir alanda dönüştürülmesi sağlayan boyut azaltma methodlarıdır.

c) *Attribute selection:* İlgisiz, zayıf ilgili attribute'lerin tespitini ve kaldırılması işlemini gerçekleştiren boyut azaltma methodudur.

#### **2) Numerosity Reduction:**

Orijinal datanın veri hacmini verinin daha küçük bir alternatif formu ile değiştiren metoddur.

a) *Parametric:* Verinin gerçek değeri yerine parametrelerini tutma yöntemidir. Regresyon ve log-linear model bu yönteme örnek olarak verilebilir.

b) *Nonparametric:* Histogram, clustering, sampling ve data cube aggregation kullanarak datayı azaltan yöntemdir.

#### **3) Data Compression**

Veri sıkıştırıldığında, orijinal verilerin küçültülmüş veya "sıkıştırılmış" bir temsilini elde edecek şekilde dönüşümler uygulanır.

a) *Kayıpsız:* Eğer orijinal data sıkıştırılmış datadan tekrardan oluşturulursa ve herhangi bir data kaybı olmaz ise "kayıpsız" olarak adlandırılır.

b) *Kayıplı:* Orijinal datayı bu teknik ile küçültürsek ve veri kaybı oluşursa "kayıplı" olarak adlandırılır.

#### **1.a) Wavelet Transforms**

Sinyal işleme teknigidir. 5 elemanlı X vektörüne bu transform işlemi uygulanarak yeni bir X' vektörü oluşturuldu fakat bu vektör de 5 elemanlı. Orijinal ve transform edilen datanın büyüklüğü aynı ise burada nasıl Data Reduction söz konusu olabilir?

Transform edilen vektörde önemsiz datalar da bulunur.

Transforme edilen vektörden tekrar orijinal dataya DWT ile ulaşmak da mümkündür.

### **1.b) Principal Components Analysis**

Boyut azaltımında kullanılan gözetimsiz (unsupervised) algoritmaların biridir.

Temel bileşenleri bularak verinin boyutunu küçültmeye yarayan istatistikler bir süreçtir.

İletişim kanalları ve görüntü işleme gibi alanlarda kullanılan bir yöntemdir.

PCA algoritmasında kullanılan bazı yaygın terimler:

**Dimensionality (Boyuyluluk):** Verilen veri setinde bulunan özelliklerin veya değişkenlerin sayısıdır. Daha kolay bir şekilde, veri kümesinde bulunan sütun sayısıdır.

**Correlation (Korelasyon):** İki değişkenin birbiriyle ne kadar güçlü bir şekilde ilişkili olduğunu gösterir. Mesela biri değişimse diğer değişken de değişir. Korelasyon değeri -1 ile +1 arasında değişir. Burada değişkenler birbiriyle ters orantılı ise -1, değişkenlerin birbiriyle doğru orantılı olduğunu +1 gösterir.

**Orthogonal (Ortogonal):** Değişkenlerin birbiriyle ilişkili olmadığını ve bu nedenle değişken çifti arasındaki korelasyonun sıfır olduğunu tanımlar.

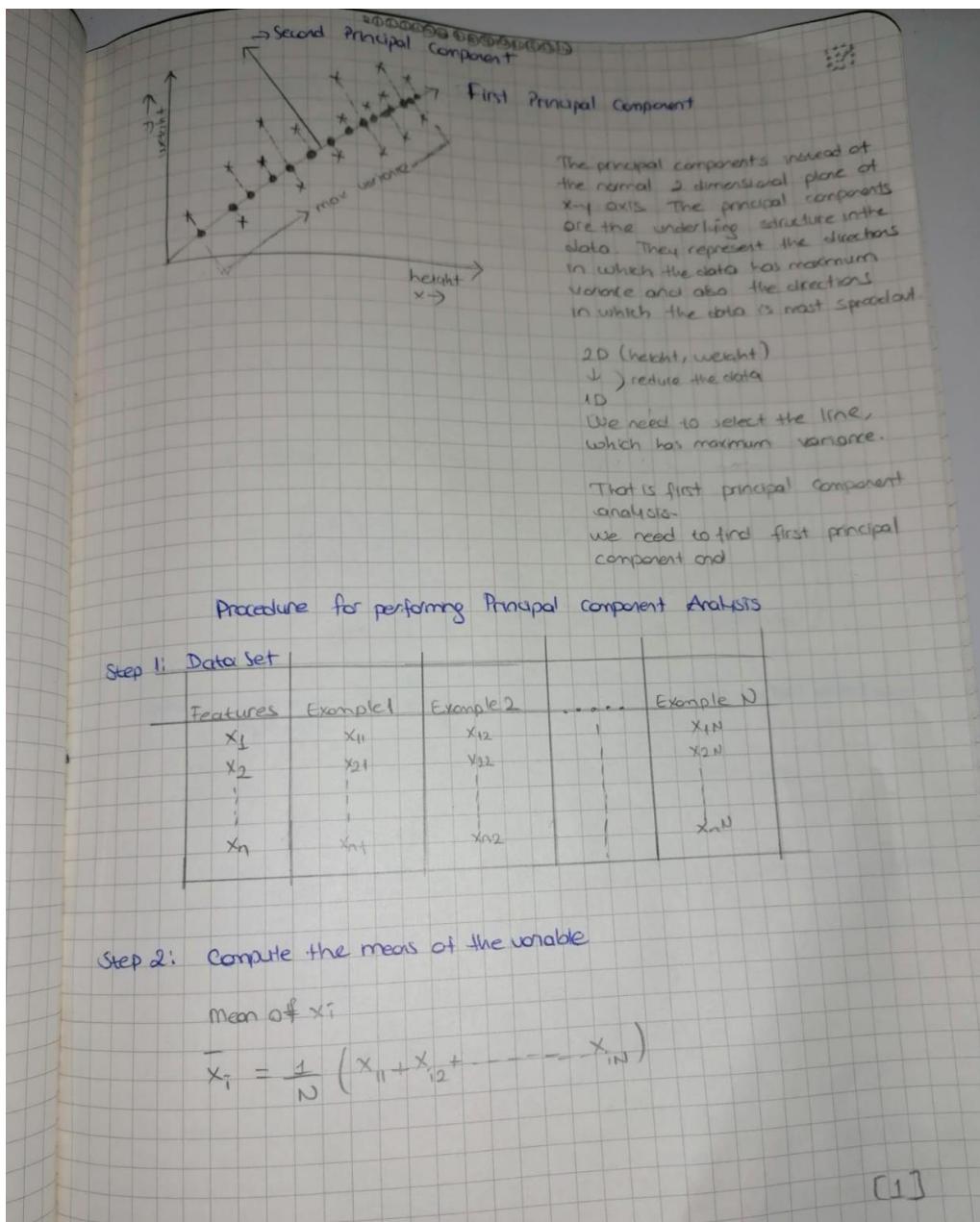
**Eigenvectors (Özvektörler):** Bir  $M$  kare matrisi varsa ve sıfırdan farklı bir  $v$  vektörü verilmişse. O zaman  $Av$ ,  $v$ 'nin skaler katı ise özvektör olacaktır.

**Covariance Matrix (Kovaryans Matrisi):** Değişken çiftleri arasındaki kovaryansı içeren bir matrise Kovaryans Matrisi denir.

Örnek: PCA Uygulamak için gereken adım ve formüller ile Problemler ve çözümleri

Adım 1: Veri setini okumak ve anlamlandırmak.

Adım 2: Değerleri toplayıp bölmek/ ortalama bulmak.



Step 3: Calculate the covariance matrix  $\Sigma$

→ Covariance of all the ordered pairs  $(x_i, x_j)$

Suppose that we have two features namely  $x_1$  and  $x_2$ , At time

we calculate  $(x_1, x_1), (x_1, x_2), (x_2, x_1), (x_2, x_2)$ , So we get  $n^2$

result for cov matrix

\* Covariance of all the ordered pairs  $(x_i, x_j)$

$$\text{cov}(x_i, x_j) = \frac{1}{N-1} \sum_{k=1}^N (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)$$

\* Construct  $N \times n$  matrix  $S$  called

$$S = \begin{bmatrix} \text{cov}(x_1, x_1) & \text{cov}(x_1, x_2) & \dots & \text{cov}(x_1, x_n) \\ \text{cov}(x_2, x_1) & \text{cov}(x_2, x_2) & \dots & \text{cov}(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(x_n, x_1) & \text{cov}(x_n, x_2) & \dots & \text{cov}(x_n, x_n) \end{bmatrix}$$

Step 4: Calculate the eigenvalues and normalized eigen vectors of the covariance matrix

\* To find eigen values, solve the equation,

$$\det(S - \lambda I) = 0$$

where  $S$  is covariance matrix,  $\lambda$  is the root of the this equation

$I$  is identity matrix

We get  $n$  roots  $\lambda_1, \lambda_2, \dots, \lambda_n$ ,

which are eigen values, such that

$$\lambda_1 > \lambda_2 > \lambda_3 > \dots > \lambda_n$$

largest to smallest

Each eigen value shows

the principal component  
so  $\lambda_1$  shows the first principal component analysis.

[2]

Adım 3: Kovaryans matrisini oluştur hesapla.

Adım 4.0: Normalize edilmiş özdeğerleri verilen formül ile hesapla.

Adım 4.1: Normalize edilmiş özvektörleri verilen formül ile hesapla. ( $S = \text{kovaryans matris}$ )

- \* For each eigen values the corresponding eigen vector is a vector

$$U = \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_n \end{bmatrix} \quad n \times 1$$

- \* To find the eigen vector, solve following equation

$$(S - \lambda' I) U = 0$$

where  $S$  is the covariance matrix,  $I$  is the identity matrix.

- \* Normalise the eigen vector

\* Divide the vector,  $U$  by its length.

i.e. normalised eigen vectors.

$$e_i = \frac{U_i}{\|U_i\|}$$

$$\text{where } \|U_i\| = \sqrt{U_{11}^2 + U_{22}^2 + \dots + U_{nn}^2}$$

- \* The unit eigen vector corresponding to the largest eigen value is the first principal component.

### Step 5: Derive new dataset

New dataset with reduced dimension is

Feature	Example 1	Example 2	...	Example N
PC1	P <sub>11</sub>	P <sub>12</sub>		P <sub>1N</sub>
PC2	P <sub>21</sub>	P <sub>22</sub>		P <sub>2N</sub>
⋮	⋮	⋮	⋮	⋮
PCn	P <sub>n1</sub>	P <sub>n2</sub>		P <sub>nN</sub>

$$P_{ij} = \sigma_i^T \begin{bmatrix} x_{1j} - \bar{x}_1 \\ x_{2j} - \bar{x}_2 \\ \vdots \\ x_{nj} - \bar{x}_n \end{bmatrix}$$

2D  
PC1, PC2

[3]

Adım 5: Yeni veri seti oluştur.

Q1: Given the following data, use PCA to reduce the dimension from 2 to 1.

Feature	Example 1	Example 2	Example 3	Example 4
x	4	8	13	7
y	11	4	5	14

Ans) Step 1: Dataset

Feat	Ex1	Ex2	Ex3	Ex4
x	4	8	13	7
y	11	4	5	14

No of features,  $n=2$   
No of samples,  $N=4$

x	y
4	11
8	4
13	5
7	14

Step 2: Computation of mean of variables

$$\bar{x} = \frac{4+8+13+7}{4} = \frac{32}{4} = 8 //$$

$$\bar{y} = \frac{11+4+5+14}{4} = 8.5 //$$

$$\text{cov}(xx) = \frac{1}{N-1} \sum_{k=1}^N (x_k - \bar{x})^2$$

Step 3: Computation of covariance matrix

ordered pairs are  
(x,x), (x,y), (y,x), (y,y)

$$\begin{matrix} x \\ y \\ \vdots \\ x \\ y \\ \vdots \\ x \\ y \end{matrix}, \begin{matrix} n \\ n^2 \\ \vdots \\ n \\ n \\ \vdots \\ n \\ n \end{matrix}$$

Covariance of all ordered pairs

$$\text{cov}(x,x) = \frac{1}{n-1} \sum_{i=1}^{n-1} (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)$$

$$= \frac{1}{4-1} \left[ (4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2 \right]$$

$$= 14 //$$

$$\text{cov}(x,y) = \frac{1}{(4-1)} (4-8)(11-8.5) + (8-8)(4-8.5) + (13-8)(5-8.5) + (7-8)(14-8.5)$$

$$= -11 //$$

$$\text{cov}(y,x) = \text{cov}(x,y)$$

$$\text{cov}(y,y) =$$

[4]

Adım 1: Veri setini incele. X ve Y attribute'unun aldığı değerleri görüyoruz ve alınan features ve samples sayılarını belirliyoruz.

Adım 2: Her değişkenin ortalamalarını bulduk.

Adım 3: Kovaryans matris hesaplanır ve yazılır.

$$\text{cov}(x,y) = \frac{1}{n-1} \left[ (11-8.5)^2 + (4-8.5)^2 + (5-8.5)^2 + (10-8.5)^2 \right]$$

ii Covariance matrix

$$S = \begin{bmatrix} \text{cov}(xx) & \text{cov}(xy) \\ \text{cov}(yx) & \text{cov}(yy) \end{bmatrix} \quad \begin{matrix} n \times n \\ 2 \times 2 \end{matrix}$$

$$S = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

Step 4: Eigen value, Eigen Vector, Normalized eigen vector

i) Eigen value

$\left\{ \begin{array}{l} \text{Calculate the eigenvalues and normalized} \\ \text{eigen vectors of the covariance matrix} \end{array} \right.$

$$\det(S - \lambda I) = 0$$

where  $S$  is the covariance matrix,  $\lambda$  is the eigen value,  $I$  is the identity matrix.

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$\lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\lambda I = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

$$S - \lambda I = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

$$S - \lambda I = \begin{bmatrix} 14-\lambda & -11-0 \\ -11-0 & 23-\lambda \end{bmatrix} = \begin{bmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{bmatrix}$$

$$\det \begin{pmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{pmatrix} = 0$$

$$(14-\lambda)(23-\lambda) - (-11) * (-11) = 0$$

$$\lambda^2 - 37\lambda + 201 = 0$$

$$\lambda_1 = 30.3849, \lambda_2 = 6.6151$$

$$\lambda_1 > \lambda_2$$

$$\lambda_1 = 30.3849 \Rightarrow \text{First principal component}$$

$$\lambda_2 = 6.6151$$

$$\rightarrow \frac{1}{2a} \sqrt{b^2 - 4ac}$$

$$\begin{matrix} a=1 \\ b=-37 \\ c=201 \end{matrix}$$

$$[\bar{x}]$$

Adım 4: Vektör ve özdeğerler hesaplanır. ( $S = \text{Cov Matris} / \text{Lambda} = \text{Eigen değeri} / I = \text{Identity Matris}$ )

2 feature olduğu için  $2 \times 2$  matris

Lambda,  $I$  ile çarpılır

$S$  de formülde yerine eklenerek matrsite çıkarma işlemi yapılır.

Determinant alınır ve sıfıra eşitlenir. Lambda değerleri birden fazla çıkar ve en büyük olan Lambda 1'dir

ii) Eigen vector of  $\lambda_1$

$$(S - \lambda_1 I) U_1 = 0$$

where  $S$  is the covariance matrix,  $\lambda_1$  is the largest eigen value  
 $I$  is the identity matrix,  $U_1$  is the eigen vector of  $\lambda_1$

$$S = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\lambda_1 I = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_1 \end{bmatrix}$$

$$\lambda_1 I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \lambda_1$$

$$S - \lambda_1 I = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix} - \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_1 \end{bmatrix}$$

$$= \begin{bmatrix} 14 - \lambda_1 & -11 - 0 \\ -11 - 0 & 23 - \lambda_1 \end{bmatrix} = \begin{bmatrix} 14 - \lambda_1 & -11 \\ -11 & 23 - \lambda_1 \end{bmatrix}$$

$$= \begin{bmatrix} 14 - \lambda_1 & -11 \\ -11 & 23 - \lambda_1 \end{bmatrix} U_1 = 0$$

$$U_1 = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$= \begin{bmatrix} 14 - \lambda_1 & -11 \\ -11 & 23 - \lambda_1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = 0$$

$$= \begin{bmatrix} (14 - \lambda_1)u_1 - 11u_2 \\ -11u_1 + (23 - \lambda_1)u_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$(14 - \lambda_1)u_1 - 11u_2 = 0$$

$$(14 - \lambda_1)u_1 - 11u_2 = 0 \checkmark$$

$$-11u_1 + (23 - \lambda_1)u_2 = 0$$

$$\frac{u_1}{11} = \frac{u_2}{23 - \lambda_1} = t$$

when  $t = 1$

$$u_1 = 11t$$

$$u_2 = 11t - 11$$

$$\text{Eigen vector } U_1 \text{ of } \lambda_1 = \begin{bmatrix} 11 \\ 11 - 11 \end{bmatrix} = \begin{bmatrix} 11 \\ 14 - 30.3849 \end{bmatrix}$$

$$= \begin{bmatrix} 11 \\ -16.3849 \end{bmatrix}$$

iii) Normalize the eigen vector  $U_1$

$$e_1 = \begin{bmatrix} \frac{11}{\sqrt{11^2 + (16.38)^2}} \\ \frac{-16.3849}{\sqrt{11^2 + (16.38)^2}} \end{bmatrix} = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix} [6]$$

Her bir lambda değeri için eigen değeri bulunur. ( $U_1$ = Lambda 1 in eigen vektörü)

Eigen vektörü normalize edilir.

$$e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$$

$e_1$  ve  $e_2$  dependent

$$e_1 = \begin{bmatrix} 0.5574 \\ 0.8313 \end{bmatrix}$$

$$e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$$

→ Step 5: Derive new dataset

we have 2 variable (xandy)

we  
from  
Our aim is to reduce 2 dimension  
to 1 dimension.

First Principal PC1 component	Ex1	Ex2	Ex3	Ex4
PC1	P <sub>11</sub>	P <sub>12</sub>	P <sub>13</sub>	P <sub>14</sub>

Let's remember Dataset:

$$\begin{array}{ccccc} x & 4 & 8 & 13 & 7 \\ y & 11 & 4 & 5 & 14 \end{array}$$

Now, we find the P<sub>11</sub>, P<sub>12</sub>, P<sub>13</sub>, P<sub>14</sub>

$$P_{11} = e_1^T \begin{bmatrix} 4-8 \\ 11-8.5 \end{bmatrix}$$

$$P_{1j} = e_i^T \begin{bmatrix} x_{ij} - \bar{x}_1 \\ x_{ij} - \bar{x}_2 \end{bmatrix}$$

$$= [0.5574 \quad -0.8303] \begin{bmatrix} -4 \\ 2.5 \end{bmatrix}$$

$$= -4.3052$$

$$P_{12} = [0.5574 \quad -0.8303] \begin{bmatrix} 8-8 \\ 4-8.5 \end{bmatrix}$$

$$P_{12} = 3.7361$$

$$P_{13} = 5.6928$$

$$P_{14} = -5.1238$$

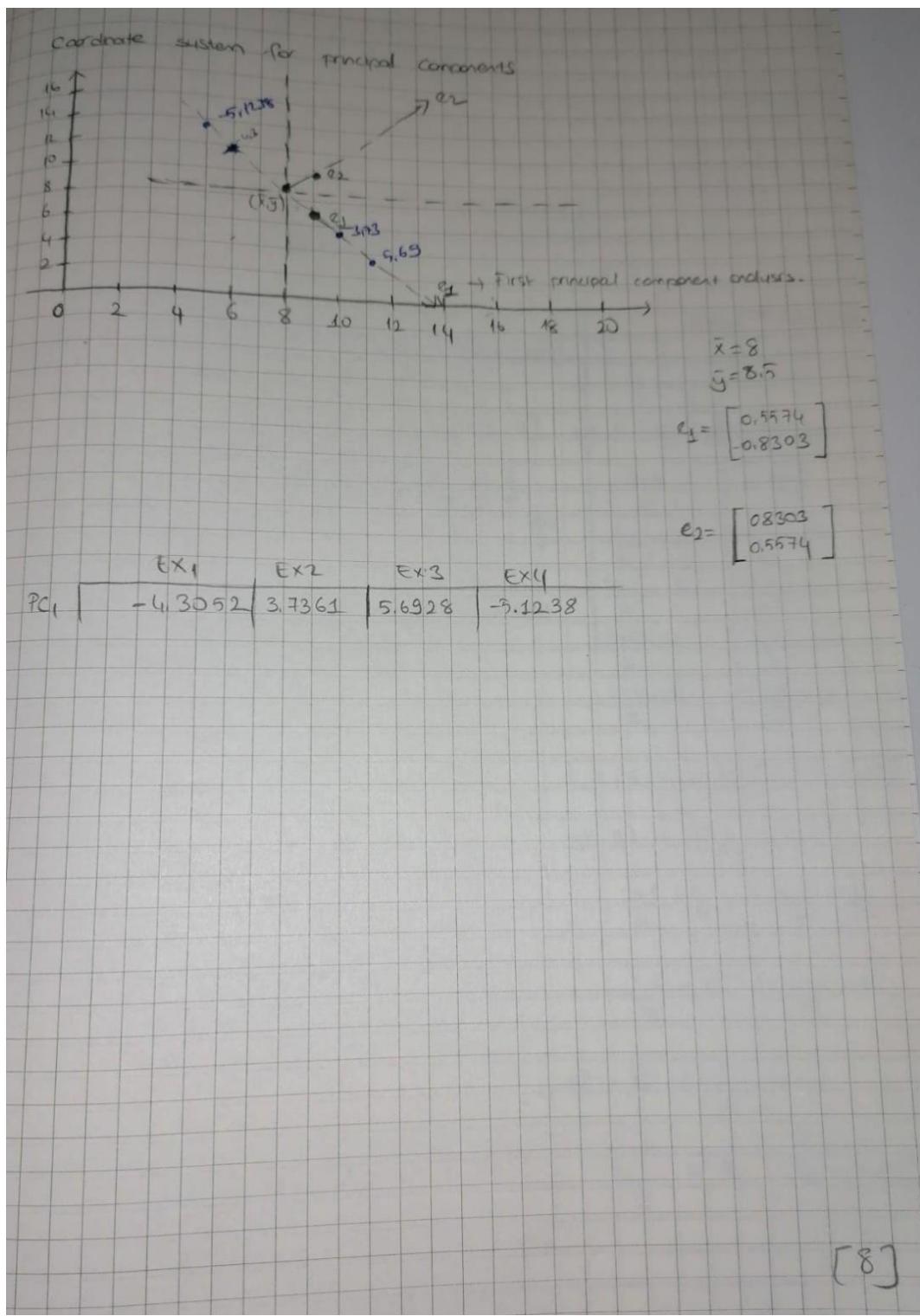
That is new dimension

PC1	Ex1	Ex2	Ex3	Ex4	Ex5
PC1	-4.3052	3.7361	5.6928	-5.1238	10.11

[7]

İkinci Eigen Vektörü bulunur.

Adım 5: Yeni bir veri kümesi oluşturulur. İki boyuta indirgedik.



Kordinat çizilir.

Datalar yeni kordinat sisteminde ifade edilir.

### 1.c) Attribute Subset Selection

Veri setimizdeki attribute sayısı yüzlerce olabilir. İlgisiz veya fazla özelliklerin veri setinde bulunmaması gereklidir. Örneğin yaş ile adres attribute'ları bazı durumlarda çok ilgisiz olabilir. İlgili özellikleri çıkarıp ilgisizleri tutmak karmaşıklığa sebep olur. Neticede, ortaya çıkan patternimiz low-quality olur.

Attribute S.S., verinin boyunu ilgisiz veya fazla attribute'leri kaldırarak veri boyutunu azaltır. Böylece verinin boyutunu azaltırken karakteristik özelliğinin de kaybolmamasını sağlar.

Exhaustive search: Orijinal niteliklerin 'iyi' bir alt kümesini nasıl bulabiliyor?" n attribute için,  $2^n$  olası altküme. Niteliklerin optimal alt kümesi için kapsamlı bir aramadır fakat engelleyici ve pahalı olabilir. Karmaşıklık ve maliyet yüksek olur.

Heuristic methods: Exh. Search yerine kullanılır. Arama yapılrken açgözlülük teknigi ile en iyi seçeneği arar. Local içinde en iyisini tercih ederek ardından globalde en iyisine ulaşmaya çalışır.

!FARK => Ex. Search kapsamlı, Heu. Meth. İse o zaman içerisinde en iyisni bulmaya çalışır.

Decision tree inşa edilebilmek için **information gain** hesaplanması gereklidir. Information gain kullanılarak attribute'ların önem derecesi bulunur. Hangi attribute en iyi, hangisi kötü gibi sorulara cevap verir.

Heuristic Metotları 4'e ayrılır:

**1) Forward Selection yöntemi:** Elimizdeki attribute setinden en iyisini bulmak için boş küme içerisine kendi aralarındaki en iyi attribute'leri bularak sırayla yerleştirme yöntemidir. A1,A4 ve A6 attribute'ları bu attribute setinde önemli olan attribute'lar olmuş.

**2) Backward Elimination yöntemi:** Elimizdeki attribute setinden en iyisini bulmak için elimizdeki tüm setten her seferde kötü olan attribute çıkarılarak en iyi sonuca ulaşılır.

**3) Decision Tree:** Node'larda yer alan attribute: 1,4,6 (önemli olanlar)

3 yöntemle de aynı attribute'lar iyi olarak sonuçlandı.

Forward selection	Backward elimination	Decision tree induction
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  Initial reduced set: $\{\}$ $\Rightarrow \{A_1\}$ $\Rightarrow \{A_1, A_4\}$ $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  $\Rightarrow \{A_1, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_4, A_5, A_6\}$ $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  <pre> graph TD     Root[A4?] -- Y --&gt; A1[A1?]     Root -- N --&gt; A6[A6?]     A1 -- Y --&gt; Class1_1((Class 1))     A1 -- N --&gt; Class2_1((Class 2))     A6 -- Y --&gt; Class1_2((Class 1))     A6 -- N --&gt; Class2_2((Class 2))   </pre> $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$

Greedy (heuristic) methods for attribute subset selection.

**Özet:** Heuristic Metot kaça ayrılır, kategorileri nelerdir?

1. Stepwise forward selection
2. Stepwise backward elimination
3. Combination of forward selection and backward elimination (önce iyi bulur sonra kötüyü çıkarır)
4. Decision tree induction ()

### Regression and Log-Linear Models: Parametric Data Reduction

Regresyon, verilen data üzerinden tahmin gerçekleştirir.

variable,  $\tilde{y}$  (called a *response variable*), can be modeled as a linear function of another random variable,  $x$  (called a *predictor variable*), with the equation

$$y = wx + b, \quad (3.7)$$

### Histograms

### Clustering

Elimizdeki verileri bu algoritma ile kümeleyerek benzer verileri tespit etme amacımızı gerçekleştirdik.

### Sampling

### Data Cube Aggregation

The diagram illustrates the process of data cube aggregation. On the left, there is a hierarchical data cube structure. The top level is labeled "Year 2010". Below it is a level labeled "Quarter" and "Sales". The bottom level is labeled "Year 2009", "Quarter", and "Sales". This is followed by levels for "Year 2008", "Year 2007", and "Year 2006", each with "Quarter" and "Sales" dimensions. An arrow points from this cube to a simplified table on the right.

		Year	Sales
2008	Q1	\$1,568,000	
2009	Q2	\$2,356,000	
2010	Q3	\$3,594,000	
	Q4		\$586,000

Sales data for a given branch of AllElectronics for the years 2008 through 2010. On the *left*, the sales are shown per quarter. On the *right*, the data are aggregated to provide the annual sales.

2008 yılında 4 satırından oluşan veri tek satırına düştü. Veri bir anlamda toplanarak azalmış oldu.

## **Data Transformation and Data Discretization**

Bu bölümde veri dönüştürme yöntemleri sunulmaktadır. Bu ön işleme adımda, veriler dönüştürülür veya birleştirilir, böylece ortaya çıkan madencilik süreci daha fazla olabilir. Verimlidir ve bulunan modellerin anlaşılması daha kolay olabilir. Veri ayrıklaştırma, bir form veri dönüşümü konusu da tartışılmaktadır.

### **Data Transformation Strategies**

- 1. Smoothing:** Veriden gürültüyü kaldırmaya çalışır. Binning, regression ve clustering yöntemlerini kullanır.
- 2. Attribute construction:** Verilen attribute kümelerinden yeni attribute'ların eklenmesidir.
- 3. Aggregation:** Aggregation operasyonları ile veri transforme edilir.
- 4. Normalization:** Verinin daha küçük bir aralığa çekilmesi.
- 5. Discretization:** Numeric attribute'leri belli bir değerle temsil ederek veriyi azaltırız.
- 6. Concept hierarchy generation for nominal data:** Üst seviye konseptlerle veriyi transform etmektedir. Örneğin sokak bilgisini şehir bilgisi olarak transform etmek.

## **Chapter 8: Classification: Basic Concepts**

### **❑ Basic Concepts**

- What Is Classification?
- General Approach to Classification

### **❑ Decision Tree Induction**

- Decision Tree Induction
- Attribute Selection Measures
- Tree Pruning
- Scalability and Decision Tree Induction
- Visual Mining for Decision Tree Induction

### **❑ Bayes Classification Methods**

- Bayes' Theorem
- Naïve Bayesian Classification

### **❑ Rule-Based Classification**

- Using IF-THEN Rules for Classification
- Rule Extraction from a Decision Tree
- Rule Induction Using a Sequential Covering Algorithm

### **❑ Model Evaluation and Selection**

- Metrics for Evaluating Classifier Performance
- Holdout Method and Random Subsampling
- Cross-Validation
- Bootstrap
- Model Selection Using Statistical Tests of Significance
- Comparing Classifiers Based on Cost–Benefit and ROC Curves

### **❑ Techniques to Improve Classification Accuracy**

- Introducing Ensemble Methods
- Bagging
- Boosting and AdaBoost
- Random Forests
- Improving Classification Accuracy of Class-Imbalanced Data

## Basic Concepts

### What Is Classification?

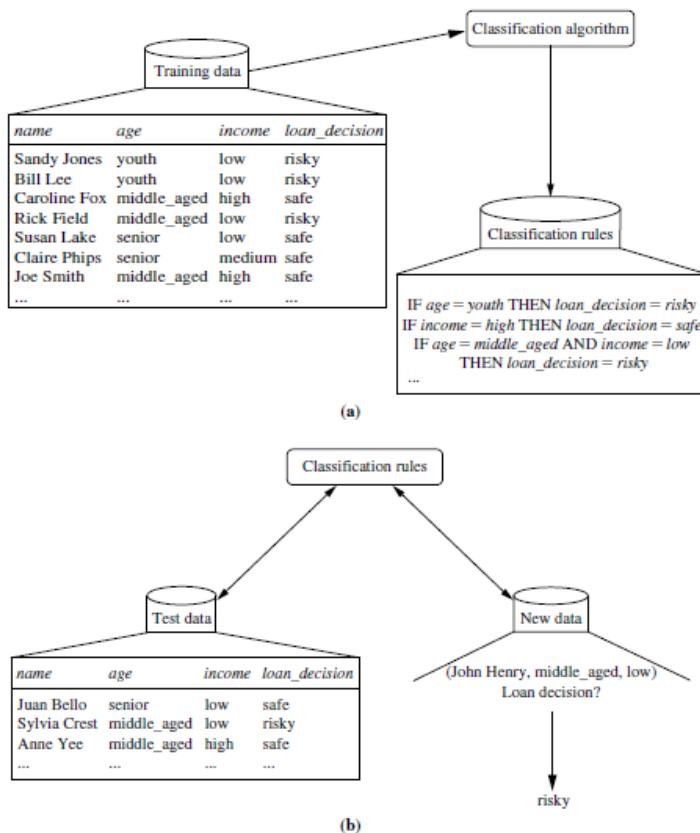
Verilerin çeşitli sebeplerden ötürü analiz edilmesi gerekebilir. Sınıflandırma, bir veri analizi yöntemidir. Kategorik/Discript değerlerin tahmininde kullanılır. Numerik verilerin analizi/tahmini ise Regresyon yöntemi ile yapılır.

### General Approach to Classification

Classification ile çözeceğimiz problemin ilk adımı için elimizde Training set olmak zorundadır. Eğitim kümesi ile model oluşturulur ve verinin sınıfı tahmin edilir.

Veri sınıflandırmasının iki adımı vardır:

- 1) *Öğrenme adımı (learning step)*: Training set ile gerçekleşir. Sınıflandırma modeli oluşturulur
- 2) *Sınıflandırma adımı (classification step)*: Verinin sınıfının tahmin edildiği aşamadır.



| The data classification process: (a) *Learning*: Training data are analyzed by a classification algorithm. Here, the class label attribute is *loan\_decision*, and the learned model or classifier is represented in the form of classification rules. (b) *Classification*: Test data are used to estimate the accuracy of the classification rules. If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples.

**Attributes:** Her bir insanın ismi, yaşı ve gelir düzeyi

**Sınıf Etiketi:** Kredi kararı (riskli mi değil mi olmak üzere iki tane sınıflandırma mevcut)

Cl. Alg. ile Eğitim kümesi kullanılarak model oluşturuldu. (a)

Test data ile bu kurallara bakılarak, modele uygulanarak yeni datalar oluşturulur veya yeni dataya bakılarak tersi.

Training datayı attribute vektörler oluşturur.

*Supervised Learning:* Her veri bir sınıfa aittir. Classification'da olduğu gibi.

*Unsupervised Learning:* Verilerin sınıfları yoktur. Clustering'de olduğu gibi.

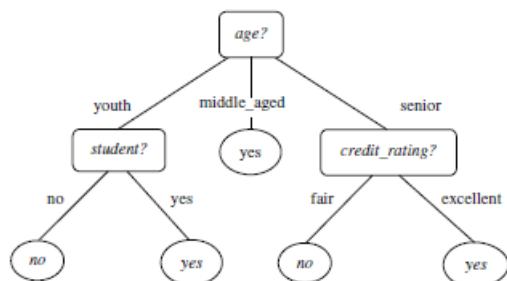
Overfit: Verinin ezberlenmesi.

Elimizdeki verilerin bir kısmı training bir kısmı da test için ayrılmalıdır. Bütün hepsi training için ayrılmamalı.

## Decision Tree Induction

Sınıflandırma Modelini oluşturabilmek için Decision Tree yöntemi kullanılır.

Karar ağacı induksiyonu, sınıf etiketli trainingden karar ağaçlarının öğrenilmesidir.



Gelen test dataya önce yaş sorulur. Cevaba göre sonuca gidilir.

Age = **root node (kök)**

Student, credit\_rating? = **internal node (iç)**

No, yes = **leaf node** (yaprak-sınıfları tahmin ediyor)

Bir karar ağacı inşa edebilmek için 3 algoritma/yol vardır:

- a) ID3
- b) C4.5
- c) CART

## ID3 yöntemi ile Desicion Tree oluşturma:

ENTROPY AND INFORMATION GAIN					
Day	Outlook	Temp	Humidity	Wind	Play Tennis
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

### ENTROPY MEASURES HOMOGENEITY OF EXAMPLES

- Entropy measures the *impurity* of a collection of examples. It depends from the distribution of the random variable  $p$ .
  - $S$  is a collection of training examples  $Entropy(S) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_-$
  - $p_+$  the proportion of positive examples in  $S$
  - $p_-$  the proportion of negative examples in  $S$

#### Examples

$$Entropy([14+, 0-]) = -14/14 \log_2(14/14) - 0 \log_2(0) = 0$$

$$Entropy([9+, 5-]) = -9/14 \log_2(9/14) - 5/14 \log_2(5/14) = 0.94$$

$$Entropy([7+, 7-]) = -7/14 \log_2(7/14) - 7/14 \log_2(7/14) = 1/2 + 1/2 = 1$$

Exit full screen (F1)

Entropy measures the impurity of a collection of examples.

It depends from the distribution of the random variable  $p$ .

Entropi, bir örnek koleksiyonunun safsızlığını ölçer.

Rastgele değişken  $p$ 'nin dağılımına bağlıdır.

Entropi 0 ise ise tüm sınıf etiketleri aynı sınıfta toplanmıştır.

Entropi 1 ise sınıf etiketlerinin sayısı birbirine eşittir.

## ENTROPY AND INFORMATION GAIN

### INFORMATION GAIN MEASURES THE EXPECTED REDUCTION IN ENTROPY

- Given entropy as a measure of the impurity in a collection of training examples, the **information gain**, is simply the expected reduction in entropy caused by partitioning the examples according to an attribute.
- More precisely, the information gain,  $Gain(S, A)$  of an attribute  $A$ , relative to a collection of examples  $S$ , is defined as,

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

- where  $Values(A)$  is the set of all possible values for attribute  $A$ , and  $S_v$  is the subset of  $S$  for which attribute  $A$  has value  $v$  (i.e.,  $S_v = \{s \in S | A(s) = v\}$ )

Cevaba en hızlı ulaşmak için en çok bilgi veren attribute (E) seçilmesidir Information Gain.

Attribute: Outlook						
<i>Values (Outlook) = Sunny, Overcast, Rain</i>						
D1	Sunny	Hot	High	Weak	No	$S = [9+, 5-]$
D2	Sunny	Hot	High	Strong	No	$Entropy(S) = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.94$
D3	Overcast	Hot	High	Weak	Yes	$S_{Sunny} \leftarrow [2+, 3-]$
D4	Rain	Mild	High	Weak	Yes	$Entropy(S_{Sunny}) = -\frac{2}{3} \log_2 \frac{2}{3} - \frac{3}{3} \log_2 \frac{3}{3} = 0.971$
D5	Rain	Cool	Normal	Weak	Yes	$S_{Overcast} \leftarrow [4+, 0-]$
D6	Rain	Cool	Normal	Strong	No	$Entropy(S_{Overcast}) = -\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4} = 0$
D7	Overcast	Cool	Normal	Strong	Yes	$S_{Rain} \leftarrow [3+, 2-]$
D8	Sunny	Mild	High	Weak	No	$Entropy(S_{Rain}) = -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} = 0.971$
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	

Mahesh Huddar

Outlook için Information Gain hesaplaması:

Tüm datanın classına bakılır: 9 no, 5 yes.

Entropi hesaplandı = 0.94

Outlookun verilerine bakılır ve her değerin entropileri hesaplanır: *sunny, overcast, rain*  
Örneğin *Sunny*'e ait 2 yes, 3 no vardır. *Sunny* entropisi = 0.971 (paydalar 5 alınır 14 değil)

Information Gain formülünde değerleri yerine yazdık ve Information Gain'i bulduk. Outlook bize 0.2464 değerinde bilgi veriyormuş.

Diğer attribute'lari da hesapladık:

Attribute: Humidity						
<i>Values (Humidity) = High, Normal</i>						
D1	Sunny	Hot	High	Weak	No	$S = [9+, 5-]$
D2	Sunny	Hot	High	Strong	No	$Entropy(S) = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.94$
D3	Overcast	Hot	High	Weak	Yes	$S_{High} \leftarrow [3+, 4-]$
D4	Rain	Mild	High	Weak	Yes	$Entropy(S_{High}) = -\frac{3}{7} \log_2 \frac{3}{7} - \frac{4}{7} \log_2 \frac{4}{7} = 0.9852$
D5	Rain	Cool	Normal	Weak	Yes	$S_{Normal} \leftarrow [6+, 1-]$
D6	Rain	Cool	Normal	Strong	No	$Entropy(S_{Normal}) = -\frac{6}{7} \log_2 \frac{6}{7} - \frac{1}{7} \log_2 \frac{1}{7} = 0.5916$
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	

$$Gain(S, Humidity) = 0.94 - \frac{7}{14} 0.9852 - \frac{7}{14} 0.5916 = 0.1516$$

Attribute: Temp						
<i>Values (Temp) = Hot, Mild, Cool</i>						
D1	Sunny	Hot	High	Weak	No	$S = [9+, 5-]$
D2	Sunny	Hot	High	Strong	No	$Entropy(S) = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.94$
D3	Overcast	Hot	High	Weak	Yes	$S_{Hot} \leftarrow [2+, 2-]$
D4	Rain	Mild	High	Weak	Yes	$Entropy(S_{Hot}) = -\frac{2}{4} \log_2 \frac{2}{4} - \frac{2}{4} \log_2 \frac{2}{4} = 1.0$
D5	Rain	Cool	Normal	Weak	Yes	$S_{Mild} \leftarrow [4+, 2-]$
D6	Rain	Cool	Normal	Strong	No	$Entropy(S_{Mild}) = -\frac{4}{6} \log_2 \frac{4}{6} - \frac{2}{6} \log_2 \frac{2}{6} = 0.9183$
D7	Overcast	Cool	Normal	Strong	Yes	$S_{Cool} \leftarrow [3+, 1-]$
D8	Sunny	Mild	High	Weak	No	$Entropy(S_{Cool}) = -\frac{3}{4} \log_2 \frac{3}{4} - \frac{1}{4} \log_2 \frac{1}{4} = 0.8113$
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	

$$Gain(S, Temp) = 0.94 - \frac{4}{14} 1.0 - \frac{6}{14} 0.9183 - \frac{4}{14} 0.8113 = 0.0289$$

Mahesh Huddar

Her bir attribute için Information Gain

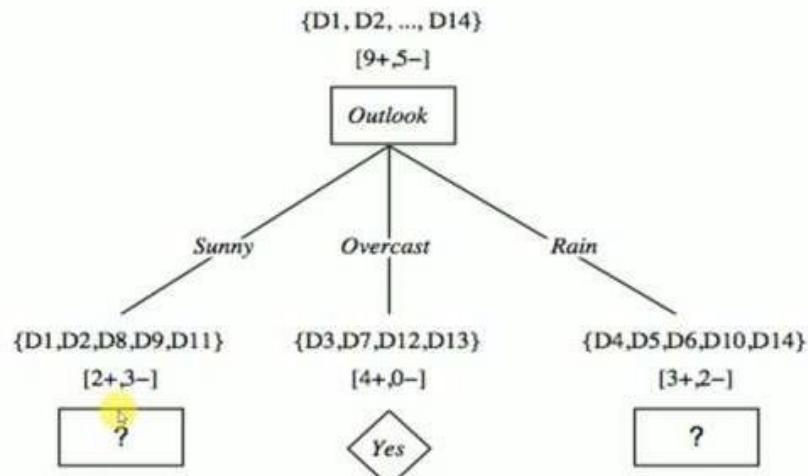
Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	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

$$Gain(S, Outlook) = 0.2464$$

$$Gain(S, Temp) = 0.0289$$

$$Gain(S, Humidity) = 0.1516$$

$$Gain(S, Wind) = 0.0478$$



Outlook, daha fazla bilgi verdiği için **ROOT NODE** oldu.

Branch'ler Outlook'un alabileceği değerler oldu. İlgili değerlerin bulunduğu satır sayıları ve sınıf etiketlerinin durumu belirtildi. Yani arama uzayı belirlenmiş oldu.

Overcast için tüm değerler aynı kayıtta (yes) toplanmıştır ve bu sebeple yaprak değeri yes olarak belirtildi.

## Sunny için Internal Node Bulma

Eski Tablo:

Day	Outlook	Temp	Humidity	Wind	Play Tennis
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

Outlook artık yok. O sütun ayrılmış oldu.

Temp için Sunny:

Day	Temp	Humidity	Wind	Play Tennis	Attribute: Temp
D1	Hot	High	Weak	No	Values (Temp) = Hot, Mild, Cool
D2	Hot	High	Strong	No	$S_{Sunny} = [2+, 3-]$ $Entropy(S_{Sunny}) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.97$
D8	Mild	High	Weak	No	$S_{Hot} \leftarrow [0+, 2-]$ $Entropy(S_{Hot}) = 0.0$
D9	Cool	Normal	Weak	Yes	$S_{Mild} \leftarrow [1+, 1-]$ $Entropy(S_{Mild}) = 1.0$
D11	Mild	Normal	Strong	Yes	$S_{Cool} \leftarrow [1+, 0-]$ $Entropy(S_{Cool}) = 0.0$

$$Gain(S_{Sunny}, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Temp) = Entropy(S) - \frac{2}{5} Entropy(S_{Hot}) - \frac{2}{5} Entropy(S_{Mild}) - \frac{1}{5} Entropy(S_{Cool})$$

$$Gain(S_{Sunny}, Temp) = 0.97 - \frac{2}{5} 0.0 - \frac{2}{5} 1.0 - \frac{1}{5} 0.0 = 0.570$$

Humidity için Sunny:

Day	Temp	Humidity	Wind	Play Tennis	Attribute: Humidity
D1	Hot	High	Weak	No	Values (Humidity) = High, Normal
D2	Hot	High	Strong	No	$S_{Sunny} = [2+, 3-]$ $Entropy(S) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.97$
D8	Mild	High	Weak	No	$S_{High} \leftarrow [0+, 3-]$ $Entropy(S_{High}) = 0.0$
D9	Cool	Normal	Weak	Yes	$S_{Normal} \leftarrow [2+, 0-]$ $Entropy(S_{Normal}) = 0.0$
D11	Mild	Normal	Strong	Yes	

$$Gain(S_{Sunny}, Humidity) = Entropy(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Humidity) = Entropy(S) - \frac{3}{5} Entropy(S_{High}) - \frac{2}{5} Entropy(S_{Normal})$$

$$Gain(S_{Sunny}, Humidity) = 0.97 - \frac{3}{5} 0.0 - \frac{2}{5} 0.0 = 0.97$$

Wind için Sunny:

Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

#### Attribute: Wind

Values (Wind) = Strong, Weak

$$S_{Sunny} = [2+, 3-]$$

$$Entropy(S) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.97$$

$$S_{Strong} \leftarrow [1+, 1-]$$

$$Entropy(S_{Strong}) = 1.0$$

$$S_{Weak} \leftarrow [1+, 2-]$$

$$Entropy(S_{Weak}) = -\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} = 0.9183$$

$$Gain(S_{Sunny}, Wind) = Entropy(S) - \sum_{v \in \{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Wind) = Entropy(S) - \frac{2}{5} Entropy(S_{Strong}) - \frac{3}{5} Entropy(S_{Weak})$$

$$Gain(S_{Sunny}, Wind) = 0.97 - \frac{2}{5} 1.0 - \frac{3}{5} 0.918 = 0.0192$$

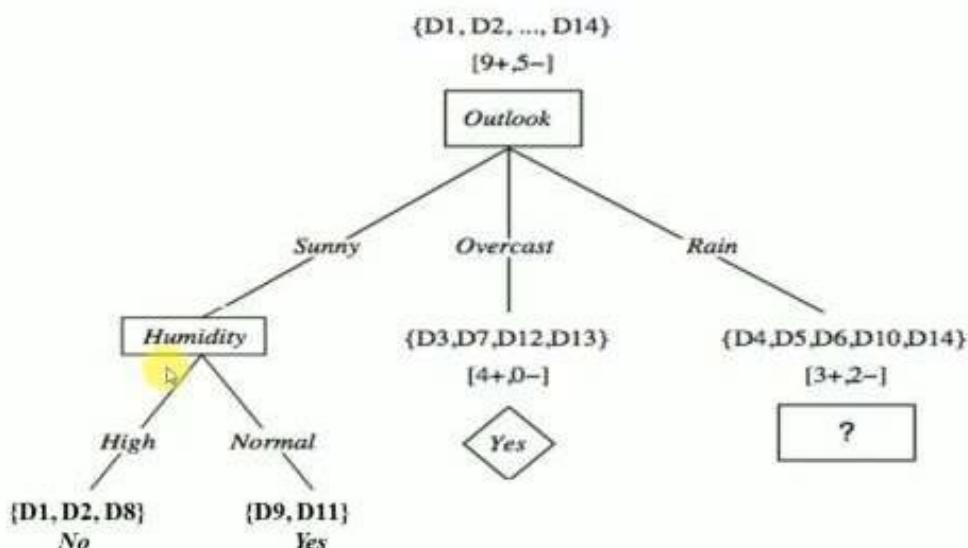
Sunny'nin 3 attribute için Information Gain'leri:

Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

$$Gain(S_{sunny}, Temp) = 0.570$$

$$Gain(S_{sunny}, Humidity) = 0.97$$

$$Gain(S_{sunny}, Wind) = 0.0192$$



Humidity daha fazla Inf. Gain'e sahip olduğu için INTERNAL NODE oldu. Branch değerleri yazıldı

### Rain için Internal Node Bulma:

Temp için Rain:

Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

#### Attribute: Temp

Values (Temp) = Hot, Mild, Cool

$$S_{Rain} = [3+, 2-]$$

$$\text{Entropy}(S_{Sunny}) = -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} = 0.97$$

$$S_{Hot} \leftarrow [0+, 0-]$$

$$\text{Entropy}(S_{Hot}) = 0.0$$

$$S_{Mild} \leftarrow [2+, 1-]$$

$$\text{Entropy}(S_{Mild}) = -\frac{2}{3} \log_2 \frac{2}{3} - \frac{1}{3} \log_2 \frac{1}{3} = 0.9183$$

$$S_{Cool} \leftarrow [1+, 1-]$$

$$\text{Entropy}(S_{Cool}) = 1.0$$

↳

$$\text{Gain}(S_{Rain}, \text{Temp}) = \text{Entropy}(S) - \sum_{v \in \{\text{Hot, Mild, Cool}\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$\text{Gain}(S_{Rain}, \text{Temp})$

$$= \text{Entropy}(S) - \frac{0}{5} \text{Entropy}(S_{Hot}) - \frac{3}{5} \text{Entropy}(S_{Mild})$$

$$- \frac{2}{5} \text{Entropy}(S_{Cool})$$

$$\text{Gain}(S_{Rain}, \text{Temp}) = 0.97 - \frac{0}{5} 0.0 - \frac{3}{5} 0.9183 - \frac{2}{5} 1.0 = 0.0192$$

↳ Gain(S\_Rain, Temp) = 0.0192

Humidity için Rain:

Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

#### Attribute: Humidity

Values (Humidity) = High, Normal

$$S_{Rain} = [3+, 2-]$$

$$\text{Entropy}(S_{Sunny}) = -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} = 0.97$$

$$S_{High} \leftarrow [1+, 1-]$$

$$\text{Entropy}(S_{High}) = 1.0$$

$$S_{Normal} \leftarrow [2+, 0-]$$

$$\text{Entropy}(S_{Normal}) = -\frac{2}{3} \log_2 \frac{2}{3} - \frac{1}{3} \log_2 \frac{1}{3} = 0.9183$$

$$\text{Gain}(S_{Rain}, \text{Humidity}) = \text{Entropy}(S) - \sum_{v \in \{\text{High, Normal}\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$\text{Gain}(S_{Rain}, \text{Humidity}) = \text{Entropy}(S) - \frac{2}{5} \text{Entropy}(S_{High}) - \frac{3}{5} \text{Entropy}(S_{Normal})$$

$$\text{Gain}(S_{Rain}, \text{Humidity}) = 0.97 - \frac{2}{5} 1.0 - \frac{3}{5} 0.9183 = 0.0192$$

↳

Wind için Rain:

Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

#### Attribute: Wind

Values (wind) = Strong, Weak

$$S_{Rain} = [3+, 2-]$$

$$\text{Entropy}(S_{Sunny}) = -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} = 0.97$$

$$S_{Strong} \leftarrow [0+, 2-]$$

$$\text{Entropy}(S_{Strong}) = 0.0$$

$$S_{Weak} \leftarrow [3+, 0-]$$

$$\text{Entropy}(S_{Weak}) = 0.0$$

$$\text{Gain}(S_{Rain}, \text{Wind}) = \text{Entropy}(S) - \sum_{v \in \{\text{Strong, Weak}\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$\text{Gain}(S_{Rain}, \text{Wind}) = \text{Entropy}(S) - \frac{2}{5} \text{Entropy}(S_{Strong}) - \frac{3}{5} \text{Entropy}(S_{Weak})$$

$$\text{Gain}(S_{Rain}, \text{Wind}) = 0.97 - \frac{2}{5} 0.0 - \frac{3}{5} 0.0 = 0.97$$

↳

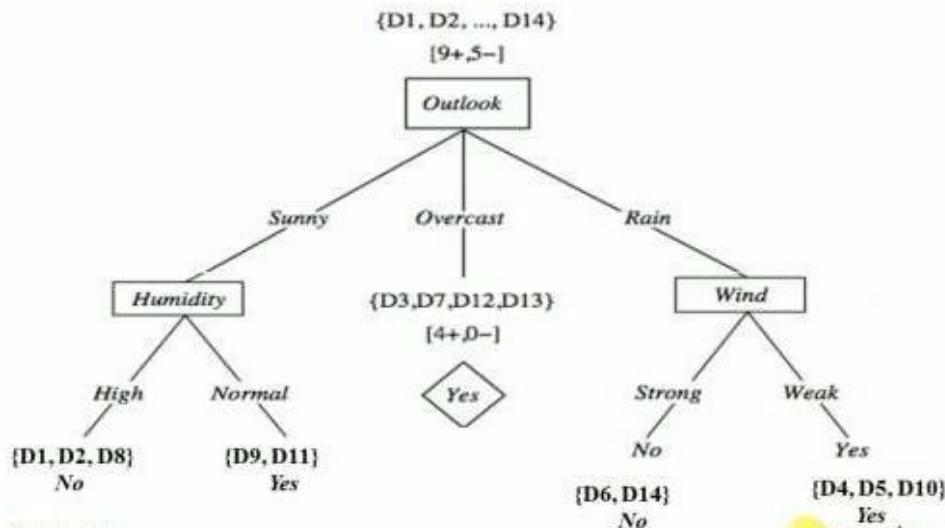
Rain'in 3 attribute için Information Gain'leri:

Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

$$Gain(S_{Rain}, Temp) = 0.0192$$

$$Gain(S_{Rain}, Humidity) = 0.0192$$

$$Gain(S_{Rain}, Wind) = 0.97$$



Wind daha fazla Inf. Gain'e sahip olduğu için INTERNAL NODE oldu. Branch değerleri yazıldı.

Wind strong olduğunda alabileceğimiz değerler D6 ve D14 satırlarında iki değer de aynı: "No" REAF NODE "No" olmuş oldu.

Wind weak olduğunda alabileceğimiz değerler D4, D5 ve D10 satırlarında üç değer de aynı: "Yes" REAF NODE "Yes" olmuş oldu.

statik ve İstatistik - Bayes Teoremi (Bayes' Theorem)

## BAYES TEOREMI

### (Bayes' Theorem)

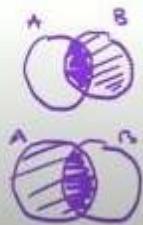
Bayes teoremi, bir olasılık değerini, bildiğimiz diğer bazı olasılık dağılımlarını kullanarak hesaplama fikridir.

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)}$$

$$P(A|B) = \frac{P(AB)}{P(B)}$$

$P(A|B) = B$  koşulu altında  $A$  nin gerçekleşme olasılığı

$P(B|A) = A$  " " "  $B$  " " "



Bayes kullanılarak bir sınıfı tahmin edeceğiz.

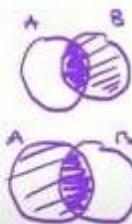
statik ve İstatistik - Bayes Teoremi (Bayes' Theorem) lik değerini, bildiğimiz diğer bazı olasılık dağılımlarını kullanarak hesaplama fikridir.

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)}$$

$$P(A|B) = \frac{P(AB)}{P(B)}$$

$P(A|B) = B$  koşulu altında  $A$  nin gerçekleşme olasılığı

$P(B|A) = A$  " " "  $B$  " " "



$P(A) = A$  nin olma olasılığı

$P(B) = B$  " " "

## **Naïve Bayes Analysis**

- Naive-Bayes (NB) technique is a supervised learning technique that uses probability-theory-based analysis.
- It is a machine-learning technique that computes the probabilities of an instance belonging to each one of many target classes, given the prior probabilities of classification using individual factors.
- Naïve Bayes technique is used often in classifying text documents into one of multiple predefined categories.

*Naive Bayes* supervised learning tekniklerinden bir tanesidir.

## **Naïve Bayes Analysis**

- NB algorithm is easy to understand and works fast.
- It also performs well in multiclass prediction, such as when the target class has multiple options beyond binary yes/no classification.
- NB can perform well even in case of categorical input variables compared to numerical variable(s)

*Multiclass prediction:* Birden fazla değerli tahmin etme işlemi. Örneği önceki örnek yes ve no.

## **Naïve Bayes Model**

- In the abstract, Naive-Bayes is a conditional probability model for classification purposes.
- The goal is to find a way to predict the class variable (Y) using a vector of independent variables i.e., finding the function  $f: X \rightarrow Y$ .
- In probability terms, the goal is to find  $P(Y|X)$ , i.e., the probability of Y belonging to a certain class X.
- Y is generally assumed to be a categorical variable with two or more discrete values.

Amaç sınıf etiketlerini tahmin etmek.

## Naïve Bayes Model

- The posterior probability (of belonging to a Class K) is calculated as a function of prior probabilities and current likelihood value, as shown in the equation,

$$p(c_k|x) = \frac{p(c_k) * p(x|c_k)}{p(x)}$$

- $p(c_k|x)$  is the posterior probability of class K, given predictor X.
  - $p(c_k)$  is the prior probability of class K.
  - $p(x)$  is the prior probability of predictor.
  - $p(x|c_k)$  is the current likelihood of predictor given class.
- 

## NAIVE BAYES CLASSIFIER Example – 2

- Estimate conditional probabilities of each attributes {color, legs, height, smelly} for the species classes: {M, H} using the data given in the table.
- Using these probabilities estimate the probability values for the new instance – (Color=Green, legs=2, Height=Tall, and Smelly=No).

No	Color	Legs	Height	Smelly	Species
1	White	3	Short	Yes	M
2	Green	2	Tall	No	M
3	Green	3	Short	Yes	M
4	White	3	Short	Yes	M
5	Green	2	Short	No	H
6	White	2	Tall	No	H
7	White	2	Tall	No	H
8	White	2	Short	Yes	H

Elimizde iki adet sınıf var: M ve H (Multiclass)

No	Color	Legs	Height	Smelly	Species
1	White	3	Short	Yes	M
2	Green	2	Tall	No	M
3	Green	3	Short	Yes	M
4	White	3	Short	Yes	M
5	Green	2	Short	No	H
6	White	2	Tall	No	H
7	White	2	Tall	No	H
8	White	2	Short	Yes	H

## NAIVE BAYES CLASSIFIER

### EXAMPLE - 2

$$P(M) = \frac{4}{8} = 0.5 \quad P(H) = \frac{4}{8} = 0.5$$

New Instance

(Color=Green, legs=2, Height=Tall, and Smelly=No)

Color	M	H
White	2/4	3/4
Green	2/4	1/4

Legs	M	H
2	1/4	4/4
3	3/4	0/4

Height	M	H
Tall	3/4	2/4
Short	1/4	2/4

Smelly	M	H
Yes	3/4	1/4
No	1/4	3/4

## NAIVE BAYES CLASSIFIER - EXAMPLE - 2

$$P(M) = \frac{4}{8} = 0.5 \quad P(H) = \frac{4}{8} = 0.5$$

Color	M	H
White	2/4	3/4
Green	2/4	1/4

Legs	M	H
2	1/4	4/4
3	3/4	0/4

Height	M	H
Tall	3/4	2/4
Short	1/4	2/4

Smelly	M	H
Yes	3/4	1/4
No	1/4	3/4

$$p(M|New\ Instance) = p(M) * p(Color = Green|M) * p(Legs = 2|M) * p(Height = tall|M) * p(Smelly = no |M)$$

$$p(M|New\ Instance) = 0.5 * \frac{2}{4} * \frac{1}{4} * \frac{3}{4} * \frac{1}{4} = 0.0117$$

$$p(H|New\ Instance) = p(H) * p(Color = Green|H) * p(Legs = 2|H) * p(Height = tall|H) * p(Smelly = no |H)$$

$$p(H|New\ Instance) = 0.5 * \frac{1}{4} * \frac{4}{4} * \frac{2}{4} * \frac{3}{4} = 0.047$$

$p(\text{Color} = \text{Green} | M) = M$  bilindiğinde rengin yeşil olma olasılığı

## NAIVE BAYES CLASSIFIER - EXAMPLE - 2

$$P(M) = \frac{4}{8} = 0.5 \quad P(H) = \frac{4}{8} = 0.5$$

Color	M	H	Legs	M	H	Height	M	H	Smelly	M	H
White	2/4	3/4	2	1/4	4/4	Tall	3/4	2/4	Yes	3/4	1/4
Green	2/4	1/4	3	3/4	0/4	Short	1/4	2/4	No	1/4	3/4

$$p(M|New\ Instance) = p(M) * p(Color = Green|M) * p(Legs = 2|M) * p(Height = tall|M) * p(Smelly = no |M)$$

$$p(M|New\ Instance) = 0.5 * \frac{2}{4} * \frac{1}{4} * \frac{3}{4} * \frac{1}{4} = 0.0117$$

$$p(H|New\ Instance) = p(H) * p(Color = Green|H) * p(Legs = 2|H) * p(Height = tall|H) * p(Smelly = no |H)$$

$$p(H|New\ Instance) = 0.5 * \frac{1}{4} * \frac{4}{4} * \frac{2}{4} * \frac{3}{4} = 0.047$$

**p(H|New Instance) > p(M|New Instance)**

**Hence the new instance belongs to Species H**

*Naive Bayes*, bir test datası geldiği zaman bu olasılık işlemleri ile yeni bir test datanın sınıfını tahmin ediyor.

**Model oluşturduk fakat modelin sağlamlığını nasıl ölçeriz?**

Confusion Matrix (Karmaşıklık/hata matrisi) ile ölçülür.

## HATA MATRİSİ (Confusion Matrix) and Accuracy

**Sınıflandırma Sistemi:**

Bir sınava hazırlanan öğrencilere ait;

- Diploma notu,
- Günlük çalışma saatı,
- Okul başarı puanı

vb. özellikler kullanılarak sınavda başarılı olup olamayacakları tahmin edilmek isteniyor.

Daha önceki sınav sonuçları ve bu sınava girenlere ait bilgiler mevcuttur.

Sıra No	Öğrenci No	Diploma Notu	Çalışma Saati	Okul Başarısı	Sonuç
1	123	85,25	3,25	75	Başarılı
2	456	77,65	2,20	66	Başarısız
3	789	65,14	1,88	59	Başarısız
4	987	78,77	2,56	56	Başarılı
5	852	67,44	2,35	67	Başarısız
6	456	91,32	4,10	78	Başarılı
...	...	...	...	...	...
400	...	...	...	...	...

Attributeler: Dip notu, Çalış saatı, Okl başarısı

Sınıf Etiketleri: Başarılı/Başarısız

TEST VERİ SETİ: 65 Başarılı ve 35 Başarısız Sınıf Etkiketi.

Test veri seti modele uygulanacak.

Makine ile yapay zekadan faydalananarak bu eğitim setinden bir model oluşturulur. Ardından test data bu modele uygulanır. Test data ile gerçek veri arasındaki farklılık kontrol edilir. Modelin doğru olup olmadığı böylece ölçülebilir. Bu modelin doğruluğunun tespiti için Karmaşıklık Matrisi kullanmamız gereklidir.

Confusion Matrix Nedir? Bir Örnekle Açıklama Accuracy Örnekle Açıklama		Hata Matrisine Doğru	
Sonuç	Tahmin	Değerlendirme	
Başarılı	Başarılı	True Positive	
Başarısız	Başarılı	False Positive	
Başarısız	Başarısız	True Negative	
Başarılı	Başarısız	False Negative	
Başarısız	Başarısız	True Negative	
Başarılı	Başarılı	True Positive	

## Model evaluation and Selection

### Train Test datasets in Machine Learning



Elimizdeki veri kümelerinin hepsi modeli oluşturmak için kullanılmaz. Bir kısmı test bir kısmı training için kullanılır ve model eğitilir. Modelin doğruluğunu ölçmek için test datası kullanılır.

**Pozitif Tuple:** İki classın içerisindeki pozitif tuple yes veya satın alınmış olması olabilir.

**Negatif Tuple:** Negatif tuple'lar no veya satın alınmamış olması olabilir.

Measure	Formula
accuracy, recognition rate	$\frac{TP+TN}{P+N}$
error rate, misclassification rate	$\frac{FP+FN}{P+N}$
sensitivity, true positive rate, recall	$\frac{TP}{P}$
specificity, true negative rate	$\frac{TN}{N}$
precision	$\frac{TP}{TP+FP}$
$F, F_1, F$ -score, harmonic mean of precision and recall	$\frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$
$F_\beta$ , where $\beta$ is a non-negative real number	$\frac{(1+\beta^2) \times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}}$

**True positives .TP/**: These refer to the positive tuples that were correctly labeled by the classifier. Let TP be the number of true positives.

**True negatives .TN/**: These are the negative tuples that were correctly labeled by the classifier. Let TN be the number of true negatives.

**False positives .FP/**: These are the negative tuples that were incorrectly labeled as positive (e.g., tuples of class buys computer D no for which the classifier predicted buys computer D yes). Let FP be the number of false positives.

**False negatives .FN/**: These are the positive tuples that were mislabeled as negative (e.g., tuples of class buys computer D yes for which the classifier predicted buys computer D no). Let FN be the number of false negatives.

		Predicted class		Total
Actual class	yes	TP	FN	P
	no	FP	TN	N
		Total	P'	N'
				P + N

¶ Confusion matrix, shown with totals for positive and negative tuples.

Satin alanlar tp + fn = P

Satin almayanlar fp + tn = N

Pozitif kayıtlar içerisinde N olduğu için başına F geldi yanlış olarak tahmin edilmiş.

TP ve TN doğru olarak tahmin edilmiş kısımlardır.

FN gerçekte pozitif tuple ancak sınıflandırıcı negatif olarak tahmin etmiş.

$$\text{accuracy} = \frac{TP + TN}{P + N}.$$

		Başarılı (Tahmin)	Başarsız (Tahmin)
Başarılı (Gerçek)	60(TP)	5(FN)	
Başarsız (Geçek)	15(FP)	20(TN)	
Başarılı (Gerçek)	60(TP)	5 (FN)	
Başarsız (Geçek)	15(FP)	20(TN)	

$$\text{Accuracy} = (\text{TP} + \text{TN}) / \text{TOPLAM}$$

$$(60+20)/100 = \% 80$$

Model 65 başarılı datanın 5 tanesini başarısız olarak tahmin etmiş.

## Confusion Matrix – Solved Example

- **Accuracy:** Overall, how often is the classifier correct?

$$\bullet \text{ Accuracy} = \frac{TN+TP}{TN+FP+FN+TP}$$

$$= \frac{45 + 95}{150} = \underline{\underline{93.33\%}}$$

	Predicted No	Predicted Yes
Actual No	TN = 45	FP = 5
Actual Yes	FN = 5	TP = 95

- **Misclassification Rate:** Overall, how often is it wrong?

$$\bullet \text{ Missclassification Rate} = \frac{FN+FP}{TN+FP+FN+TP}$$

$$= \frac{5 + 5}{150} = \underline{\underline{6.67\%}}$$

	Predicted No	Predicted Yes
Actual No	TN = 45	FP = 5
Actual Yes	FN = 5	TP = 95

- **True Positive Rate:** When it's actually yes, how often does it predict yes?

- also known as "Sensitivity" or "Recall"

$$\bullet \text{ True Positive rate} = \frac{TP}{Actual \text{ Yes}}$$

$$= \frac{95}{100} = \underline{\underline{95\%}}$$

	Predicted No	Predicted Yes
Actual No	TN = 45	FP = 5
Actual Yes	FN = 5	TP = 95

- **False Positive Rate:** When it's actually no, how often does it predict yes?

- *False Positive rate =  $\frac{FP}{Actual\ No}$*

$$= \frac{5}{50} = \underline{\underline{10\%}}$$

	Predicted No	Predicted Yes
Actual No	TN = 45	FP = 5
Actual Yes	FN = 5	TP = 95

- **True Negative Rate:** When it's actually no, how often does it predict no?

- also known as "Specificity"

- *True Negative rate =  $\frac{TN}{Actual\ No}$*

	Predicted No	Predicted Yes
Actual No	<u>TN = 45</u>	FP = 5
Actual Yes	FN = 5	TP = 95

- **True Negative Rate:** When it's actually no, how often does it predict no?

- also known as "Specificity"

- *True Negative rate =  $\frac{TN}{Actual\ No}$*

$$= \frac{45}{50} = \underline{\underline{90\%}}$$

	Predicted No	Predicted Yes
Actual No	<u>TN = 45</u>	FP = 5
Actual Yes	FN = 5	TP = 95

- **Precision:** When it predicts yes, how often is it correct?

$$\begin{aligned} \bullet \text{Precision} &= \frac{\text{TP}}{\text{Predicted Yes}} \\ &= \frac{95}{100} = 95\% \end{aligned}$$

	Predicted No	Predicted Yes
Actual No	TN = 45	FP = 5
Actual Yes	FN = 5	TP = 95

- **Prevalence:** How often does the yes condition actually occur in our sample?

$$\begin{aligned} \bullet \text{Prevalence} &= \frac{\text{Actual Yes}}{\text{Total}} \\ &= \frac{100}{150} = 66.67\% \end{aligned}$$

	Predicted No	Predicted Yes
Actual No	TN = 45	FP = 5
Actual Yes	FN = 5	TP = 95

Formüller:

$$\text{accuracy} = \frac{TP + TN}{P + N}.$$

$$\text{sensitivity} = \frac{TP}{P}$$

$$\text{specificity} = \frac{TN}{N}.$$

It can be shown that accuracy is a function of sensitivity and specificity:

$$\text{accuracy} = \text{sensitivity} \frac{P}{(P + N)} + \text{specificity} \frac{N}{(P + N)}.$$

$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{recall} = \frac{TP}{TP + FN} = \frac{TP}{P}. \quad \text{error rate} = \frac{FP + FN}{P + N}.$$

## Veri seti nasıl dengeli veya dengesiz olur?

Doğru modele ulaşabilmek için birçok veriye ihtiyaç duyulur. Ayrıca sınıflar arasındaki verinin de çok olması gereklidir.

99 elma 1 muz?! // 99 elma 80 muz

Bu elma ve muz farkında veri setimiz ilkinde dengesiz ikincisinde dengeli olur.

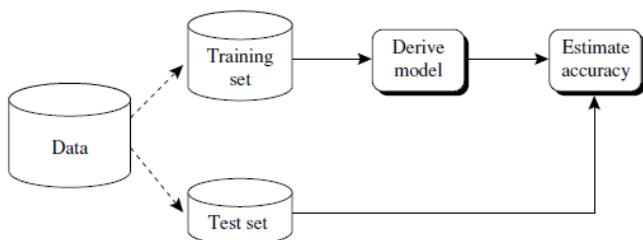
Modelin sağlamlık özellikleri:

**Speed:** Hesaplama maliyetini göstermemiz gereklidir.

**Robustness:** Modelin verilerini yani örneğin Elma ve muz sağlamken tanımaması ama gürültülü veya kayıp veriler olduğu zaman yani elma bir dilimse veya çırık ise yine tanıyalabilmesidir.

**Scalability:** Geniş mikardaki datadan bir sınıflandırıcı oluşturulabilmesidir.

**Interpretability:** Modelin karşı tarafa iyi bir şekilde ifade edinebilmesi, okunabilmesi.



I7 Estimating accuracy with the holdout method.

Doğruluğu yaparken ne kullandığımızı belirtmeliyiz:

**Holdout Method:** Verinin bir kısmının eğitim için(2/3) ve test için(1/3) tutulması

**Random Subsampling:** k kez gerçekleşen Holdout metodun bir varyasyonudur.

## CROSS VALIDATION

Rastgele olarak veriler k kez bölünür. D kümesi için Her parça D1,D2 gibi ifade edilir.

### Cross Validation in Machine Learning

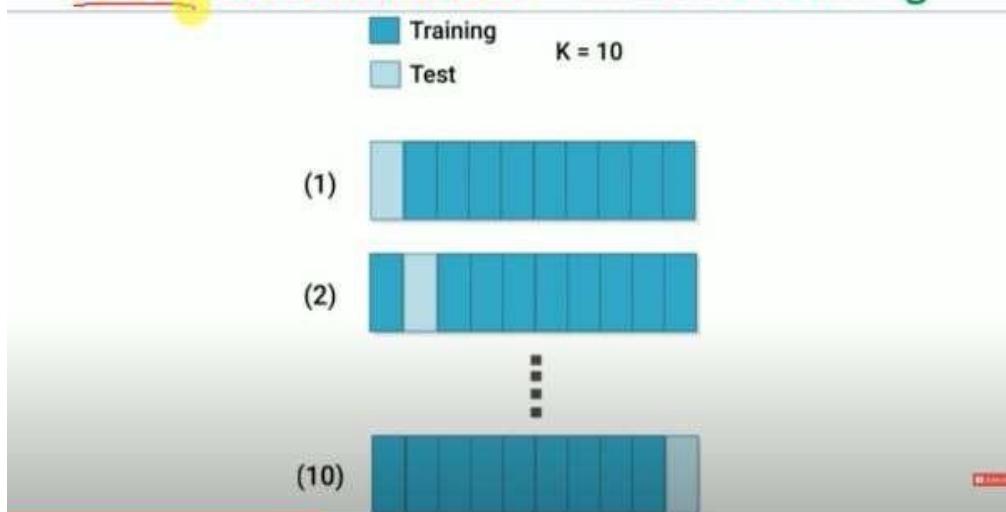
- The idea of cross-validation arises because of the *problems* with train test model or train test validation model.
- It basically wants to guarantee that the score of our model does not depend on the way we picked the train and test set.

## Types Cross Validation in Machine Learning

Following are the type of Cross Validation Techniques

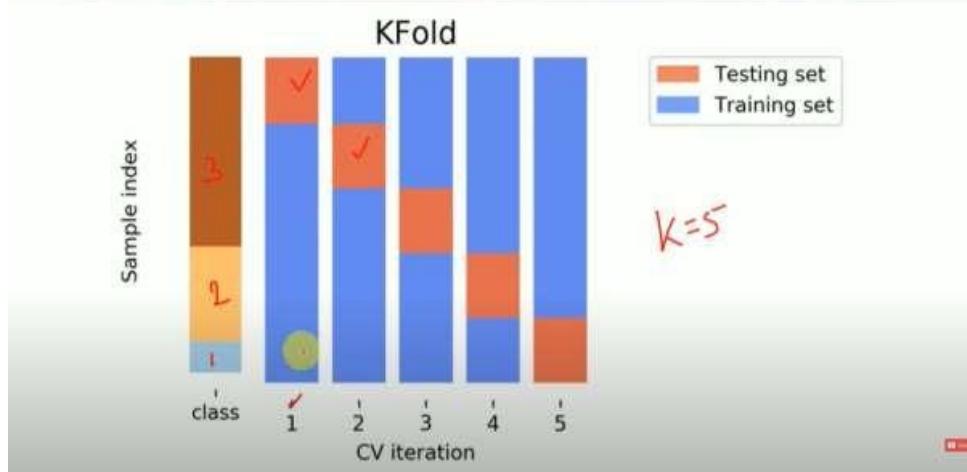
- K-folds ✓
- Stratified K-folds
- Leave-one-out
- Leave-p-out

### K-Fold Cross Validation in Machine Learning



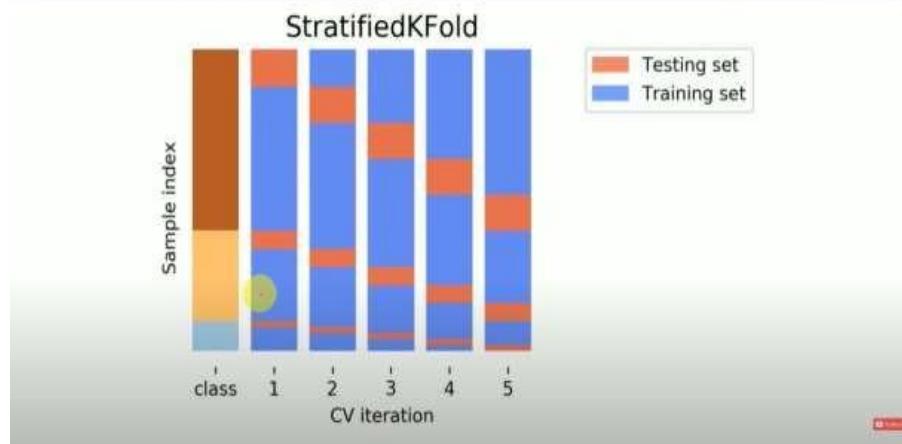
Parça parça test data ve training data . Her iterasyonda model ve accuracy oluşur.

### K-folds Cross Validation in Machine Learning



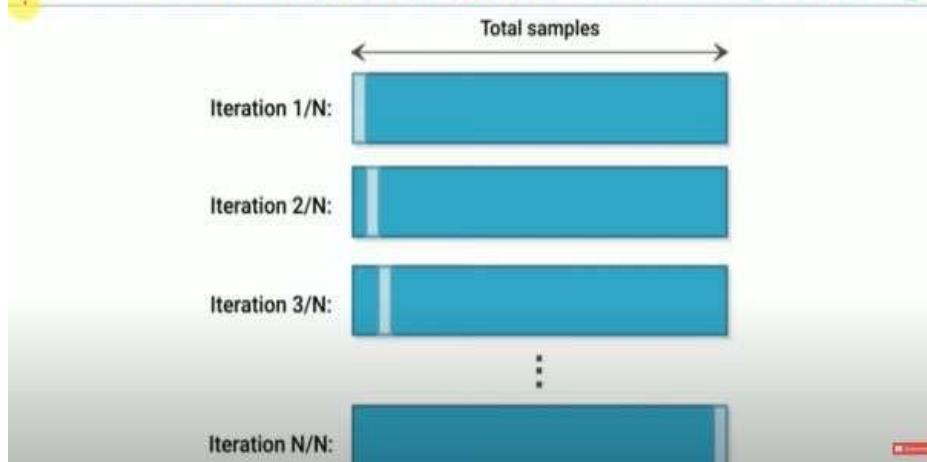
3 sınıf varmış. Mavi ile training yaptık. Test datanın yeri değişiyor k kez. Balans değil burada. 1-2-3 dengeli dağılmamış

## Stratified K-folds Cross Validation in Machine Learning



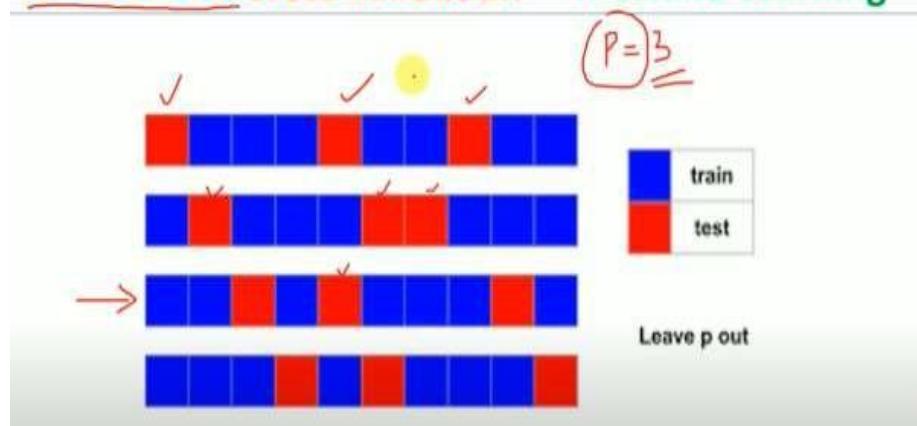
K-folds olumsuzluğunu gidermek için kullanılır. Her kısım içsn model eğitilmiş olur. Fakat hala dengesiz bir veri seti.

## Leave-one-out Cross Validation in Machine Learning



Birini test datası olarak dışarı atıp geri kalanı bu datayla eğitmektir. Maliyet çok yüksek kullanılmaz.

## Leave-P-out Cross Validation in Machine Learning



10 a bölünen datadan 3 tane test için seçilmiş. Kalan training için. Randomly seçiliyor. Aynı test datası kullanılabiliriyor yine.

# Chapter 10: Cluster Analysis: Basic Concepts and Methods

Sınıflandırma yeni datanın sınıfını bulmada kullanılırken cluster ise o verileri kümeler. Sınıf etiketi bilmez. Birçok müşteri ve attribute var ve bu algoritma ile bu müşterileri birçok kümeye/gruba gruplayabiliriz.

Biyoloji, güvenlik ve web aramaları gibi alanlarda bu algoritma kullanılır.

**Clustering Teknikleri:**

- *partitioning methods:*
- *hierarchical methods:*
- *density-based methods:*
- *grid-based methods:*

## Partitioning Clustering Methods K Means Algorithm

K-Means Clustering Algorithm - Partitioning Method - Machine Learning - Data Mining - Solved			
Data Set	C1{3}	C2{18}	(Min of 2 <sup>nd</sup> and 3 <sup>rd</sup> col)
2	1	16	C1
4	1	14	C1
10	7	8	C1
12	9	6	C2
3	0	15	C1
20	17	2	C2
30	27	12	C2
11	8	7	C2
25	22	7	C2

**Updated centroid**

$C1=(2+3+4+10)/4=4.75$      $C2=(12+20+30+11+25)/5=19.6$

Tek boyutlu data. Rastgele iki tane centroid oluşturuldu. Her datanın centroide olan uzaklığı ifade eder.

Data Set	C1( 2.5 )	C2( 16 )	Cluster No. (Min of 2 <sup>nd</sup> and 3 <sup>rd</sup> column)
2	<b>0.5</b>	14	C1
4	<b>1.5</b>	12	C1
10	7.5	<b>6</b>	C2
12	9.5	<b>4</b>	C2
3	<b>0.5</b>	13	C1
20	17.5	<b>4</b>	C2
30	27.5	<b>14</b>	C2
11	8.5	<b>5</b>	C2
25	22.5	<b>9</b>	C2

Centroid	C1	C2
Old	2.5	16
New	3	18

#### Updated centroid

$$C1 = (2+3+4)/3 = 3 \quad C2 = (10+12+20+30+11+25)/6 = 18$$

Data Set	C1( 4.75 )	C2( 19.6 )	Cluster No. (Min of 2 <sup>nd</sup> and 3 <sup>rd</sup> col)
2	<b>2.75</b>	17.6	C1
4	<b>0.75</b>	15.6	C1
10	<b>5.25</b>	9.6	C1
12	<b>7.25</b>	7.6	C1
3	<b>1.75</b>	16.6	C1
20	15.25	<b>0.4</b>	C2
30	25.25	<b>10.4</b>	C2
11	<b>6.25</b>	8.6	C1
25	20.25	<b>5.4</b>	C2

Centroid	C1	C2
Old	4.75	19.6
New	7	25

#### Updated centroid

$$C1 = (2+3+4+10+11+12)/6 = 7 \quad C2 = (20+30+25)/3 = 25$$

Her datanın yeni centroide olan uzaklığı bulundu.

Data Set	C1( 2 )	C2( 4 )	Cluster No. (Min of 2 <sup>nd</sup> and 3 <sup>rd</sup> col.)
2	<b>0</b>	2	C1
4	2	<b>0</b>	C2
10	8	<b>6</b>	C2
12	10	<b>8</b>	C2
3	<b>1</b>	1	C1
20	18	<b>16</b>	C2
30	28	<b>26</b>	C2
11	9	<b>7</b>	C2
25	23	<b>21</b>	C2

Feedback from machine( Machine itself learn from initial random centroid)

Updated centroid

$C1 = (2+3)/2 = 2.5, \quad C2 = (4+10+12+20+30+11+25)/7 = 16$

Clustering Algorithm, Partitioning Method, Machine Learning, Data Mining, Solved			
Data	C1(7)	C2(25)	
Set	(Min of 2 <sup>nd</sup> and 3 <sup>rd</sup> col)		
2	5	23	C1
4	3	21	C1
10	3	15	C1
12	5	13	C1
3	4	22	C1
20	13	5	C2
30	23	5	C2
11	4	14	C1
25	18	0	C2

**Updated centroid**

$C1 = (2+3+4+10+11+12)/6 = 7$      $C2 = (20+30+25)/3 = 25$

Bulunan yeni centroid eski centroidle aynıysa durulur.

## Hierarchical Clustering Algorithms & Agglomerative Clustering

### DENDOGRAM OLUŞTURMA

Apply Agglomerative with Single Linkage on following data.																																									
Data	A	B	C	D	E																																				
Value	1	3	5	6	9																																				
[Use absolute distance]																																									
Adjacency matrix creation based on distance type.																																									
Original:																																									
<table border="1"> <tr> <td>A</td><td>0</td><td></td><td></td><td></td><td></td></tr> <tr> <td>B</td><td>2</td><td>0</td><td></td><td></td><td></td></tr> <tr> <td>C</td><td>4</td><td>2</td><td>0</td><td></td><td></td></tr> <tr> <td>D</td><td>5</td><td>3</td><td>1</td><td>0</td><td></td></tr> <tr> <td>E</td><td>8</td><td>6</td><td>4</td><td>3</td><td>0</td></tr> <tr> <td></td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td></tr> </table>						A	0					B	2	0				C	4	2	0			D	5	3	1	0		E	8	6	4	3	0		A	B	C	D	E
A	0																																								
B	2	0																																							
C	4	2	0																																						
D	5	3	1	0																																					
E	8	6	4	3	0																																				
	A	B	C	D	E																																				
<b>Imp. Note:</b> For any type linkage, always pick up <b>minimum distance</b> from matrix																																									
We require to keep two adjacency matrices. Don't change to original matrix.																																									

**a) Single Linkage: Minimum Function**

In the original matrix, **C & D** are located closed to each other at distance 1. Merge them into single cluster.

A	0				
B	2	0			
C	4	2	0		
D	5	3	1	0	
E	8	6	4	3	0
	A	B	C	D	E

A	0			
B	2	0		
CD			0	
E	8	6		0
A	B	CD	E	

$\min[(A,C), (A,D)]$	$\min[(B,C), (B,D)]$	$\min[(E,C), (E,D)]$
$\min[4,5]$	$\min[2,3]$	$\min[4,3]$
4	2	3

A	0			
B	2	0		
CD	4	2	0	
E	8	6	3	0
A	B	CD	E	

Uzaklıklar için:

4 ve 5 arasında min 4

2 ile 3 arasında min 2

4 ile 3 arasında min 3

A	0			
B	2	0		
CD	4	2	0	
E	8	6	3	0
A	B	CD	E	

A	0			
B	2	0		
CD	4	2	0	
E	8	6	3	0
A	B	CD	E	

AB	0		
CD		0	
E		3	0
AB	CD	E	

$\min[(CD,A), (CD,B)]$	$\min[(E,A), (E,B)]$
$\min[4,2]$	$\min[8,6]$
2	6

AB	0		
CD	2	0	
E	6	3	0
AB	CD	E	

CD A olan uzaklııyla B olan uzaklığının min 2

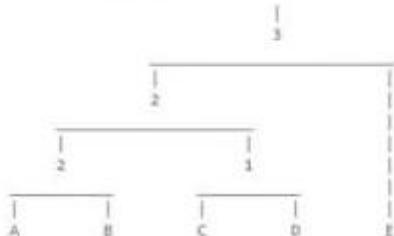
AB	0		
CD	2	0	
E	6	3	0
	AB	CD	E

AB	0		
CD	2	0	
E	6	3	0
	AB	CD	E

min[ {E,AB}, {E,CD} ]
min[6,3]
3

ABCD	0		
E	6	0	
	ABCD	E	

ABCD	0		
E	3	0	
	ABCD	E	



Dendrogram

CD 1 seviyesinde birleşti

AB 2 seviyesinde birleşti

#### b) Complete Linkage: Maximum Function

In the original matrix, **C & D** are located closed to each other at distance 1. Merge them into single cluster.

A	0				
B	2	0			
C	4	2	0		
D	5	3	1	0	
E	8	6	4	3	0
	A	B	C	D	E

A	0				
B	2	0			
CD	2	0	0		
E	8	6	0	0	
	A	B	CD	E	

max[ (A,C), (A,D) ]	max[ (B,C), (B,D) ]	max[ (E,C), (E,D) ]
max[4,5]	max[2,3]	max[4,3]
5	3	4

A	0				
B	2	0			
CD	5	3	0		
E	8	6	4	0	
	A	B	CD	E	

Maximum kullanılır.

A	0			
B	2	0		
CD	5	3	0	
E	8	6	4	0
	A	B	CD	E

A	0			
B	2	0		
CD	4	2	0	
E	8	6	3	0
	A	B	CD	E

$\max[\{CD, A\}, \{CD, B\}]$	$\max[\{E, A\}, \{E, B\}]$
$\max[5, 3]$	$\max[8, 6]$
5	8

AB	0			
CD		0		
E	3	0		
	AB	CD	E	

AB	0			
CD	5	0		
E	8	3	0	
	AB	CD	E	

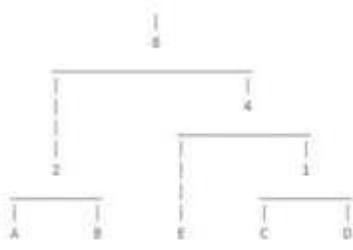
AB	0			
CD	5	0		
E	8	3	0	
	AB	CD	E	

AB	0			
CD	5	0		
E	8	3	0	
	AB	CD	E	

AB	0			
CDE		0		
	AB	CD	E	

$\max[\{AB, CD\}, \{AB, E\}]$
$\max[5, 8]$
8

AB	0			
CDE	8	0		
	AB	CDE		



Dendrogram

... =

### c) Average Linkage: Average Function

In the original matrix, C & D are located closed to each other at distance 1. Merge them into single cluster.

A	0				
B	2	0			
C	4	2	0		
D	5	3	1	0	
E	8	6	4	3	0
	A	B	C	D	E

A	0				
B	2	0			
CD				0	
E	8	6		0	
A	B	CD		E	

(A) (C,D)	(B) (C,D)	(E) (C,D)
avg[(A,C), (A,D)]	avg[(B,C), (B,D)]	avg[(E,C), (E,D)]
avg[4,5]	avg[2,3]	avg[4,3]
4.5	2.5	3.5

A	0				
B	2	0			
CD	4.5	2.5	0		
E	8	6	3.5	0	
A	B	CD		E	

Ortalama kullanılır.

AB	0			
CD	3.5	0		
E	7	3.5	0	
	AB	CD	E	

AB	0			
CD	3.5	0		
E	7	3.5	0	
	AB	CD	E	

ABCD	0			
E	5.25	0		
	ABCD	E		

(E) (ABCD)
avg[(E,A),(E,B),(E,C),(E,D)]
avg[8,6,4,3]

ABCD	0			
E	5.25	0		
	ABCD	E		

Dendrogram

