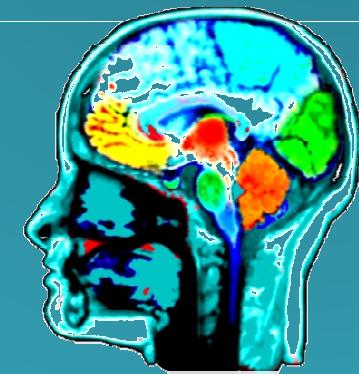




# Introduction To Data Mining

Isfahan University of Technology (IUT)  
Bahman 1401



## Introduction

---

Dr. Hamidreza Hakim  
[hamid.hakim.u@gmail.com](mailto:hamid.hakim.u@gmail.com)

# Content

---

---

References

Grading

Data

What is Data mining?

Why Data mining

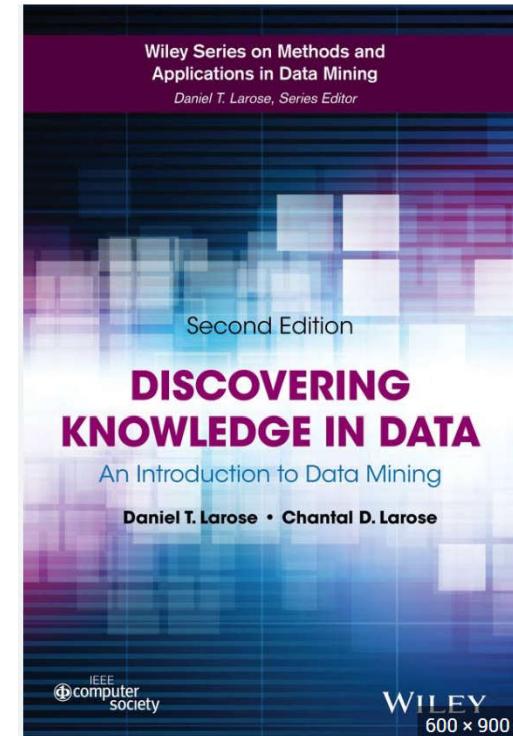
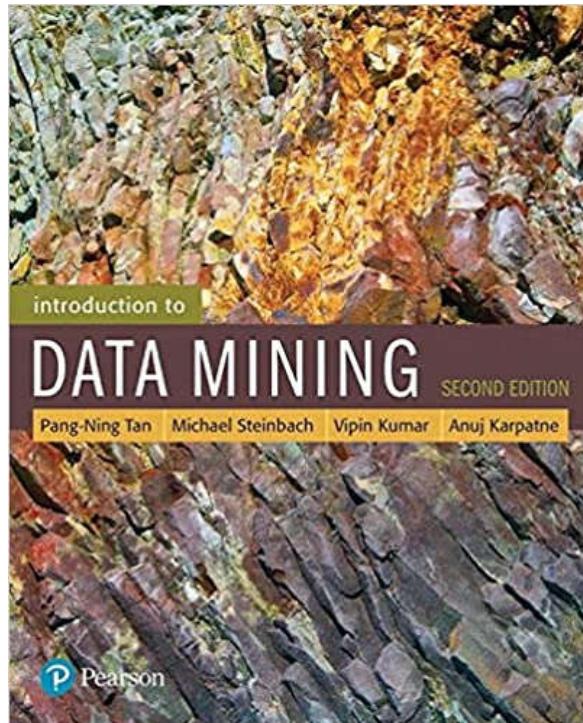
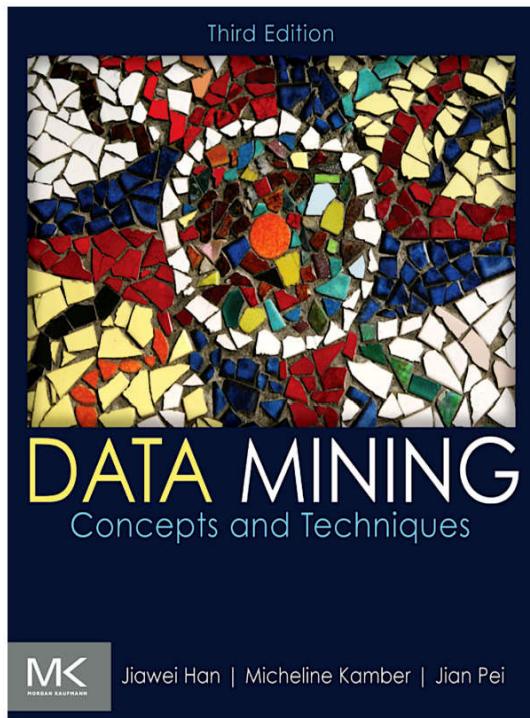
Multi-Dimensional View of Data Mining

Some Examples

# References

---

---



# Grades

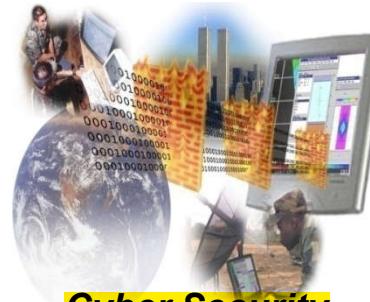
---

---

- Project and Presentation: 2 points
- Exercises: 5 points
- Exams and Quizzes: 13 points

# Large-scale Data is Everywhere!

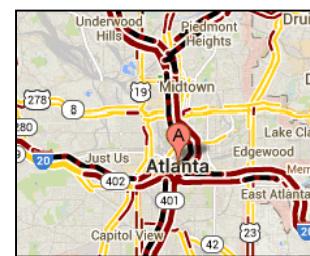
- There has been enormous data growth in both **commercial** and **scientific** databases due to advances in **data generation** and **collection** technologies
- New mantra
  - Gather **whatever** data you can whenever and **wherever** possible.
- Expectations
  - Gathered data will have **value** either for the purpose collected or for a purpose not envisioned.



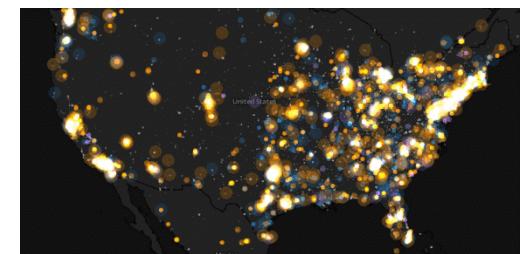
**Cyber Security**



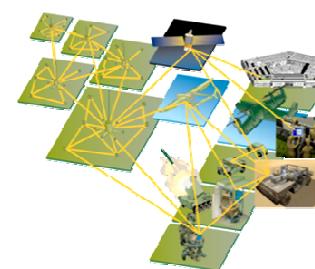
**E-Commerce**



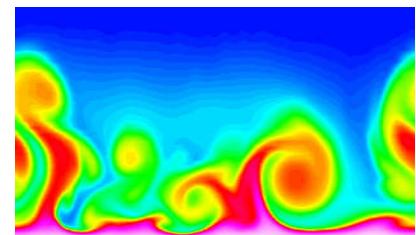
**Traffic Patterns**



**Social Networking: Twitter**



**Sensor Networks**



**Computational Simulations**

# Why Data Mining? Commercial Viewpoint

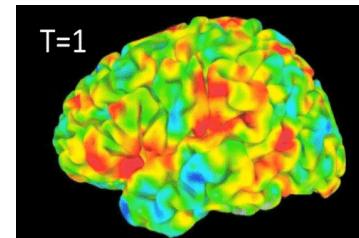
- Lots of data is being collected and warehoused
  - Web data
    - ◆ Google has Peta Bytes of web data
    - ◆ Facebook has billions of active users
  - purchases at department/grocery stores, e-commerce
    - ◆ Amazon handles millions of visits/day
  - Bank/Credit Card transactions
- Computers have become cheaper and more powerful
- Competitive Pressure is Strong
  - Provide better, customized services for an edge (e.g. in Customer Relationship Management)



داده ها با سرعت بسیار زیاد جمع آوری و ذخیره می شوند  
- سنسورهای از راه دور در ماهواره ناسا EOSDIS بیش از پنتابایت داده های علوم زمین در سال را باقیگانی می کند  
- تلسکوپ هایی که آسمان را اسکن می کنند داده های بررسی آسمان داده های fMR از مغز  
- داده های بیولوژیکی با کارایی بالا  
- شبیه سازی های علمی

# Why Data Mining? Scientific Viewpoint

- Data collected and stored at enormous speeds
  - Remote sensors on a satellite
    - ◆ NASA EOSDIS archives over petabytes of earth science data/year
  - Telescopes scanning the skies
    - ◆ Sky survey data
  - High-throughput biological data
  - Scientific simulations
    - ◆ terabytes of data generated in a few hours
- Data mining helps scientists
  - in automated analysis of massive datasets
  - In hypothesis formation

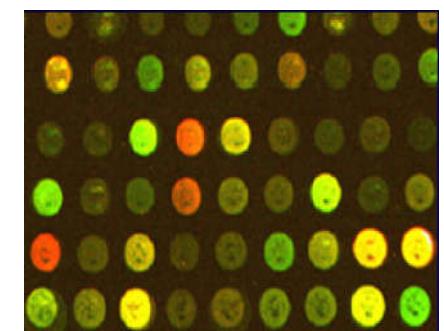


fMRI Data from Brain

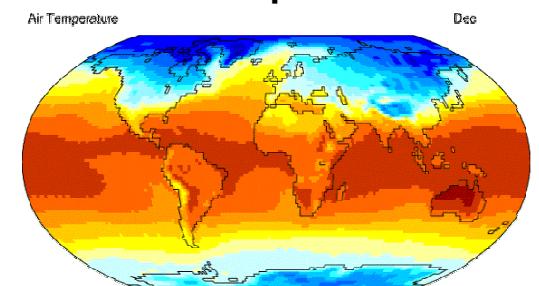


Sky Survey Data

پردازش داده های مغزی



Gene Expression Data



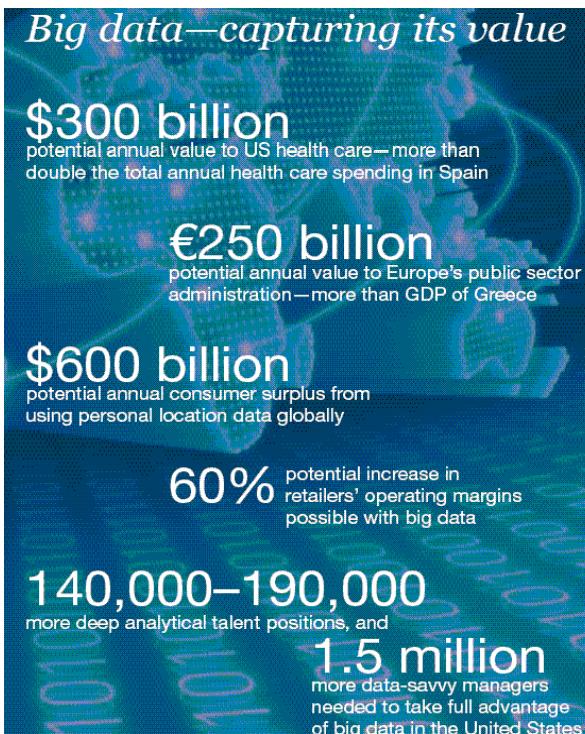
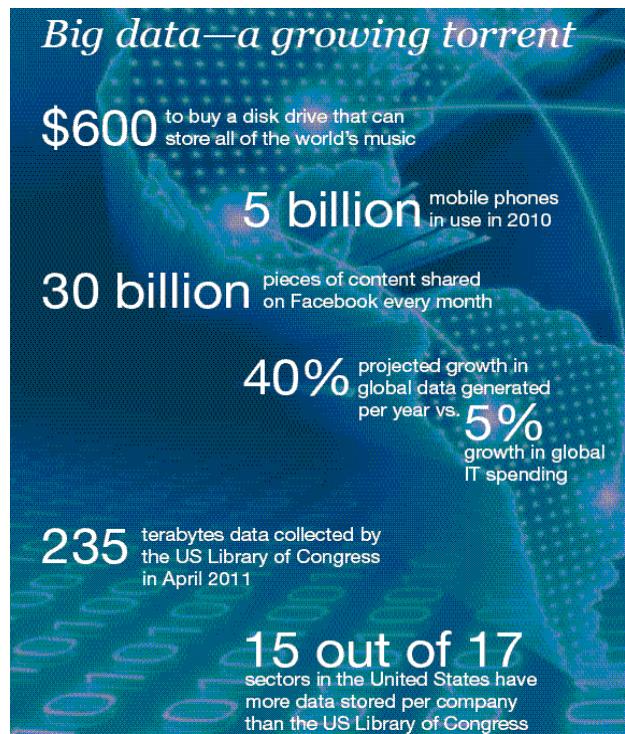
Surface Temperature of Earth

# Great opportunities to **improve productivity** in all walks of life

McKinsey Global Institute

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

کمک برای بهبود مسائل



# Great Opportunities to Solve Society's Major Problems

بهبود مراقبت های بهداشتی و کاهش هزینه ها

یافتن منابع انرژی جایگزین/ساز

پیش بینی تاثیر تغییرات آب و هوا

کاهش گرسنگی و فقر با افزایش تولیدات کشاورزی

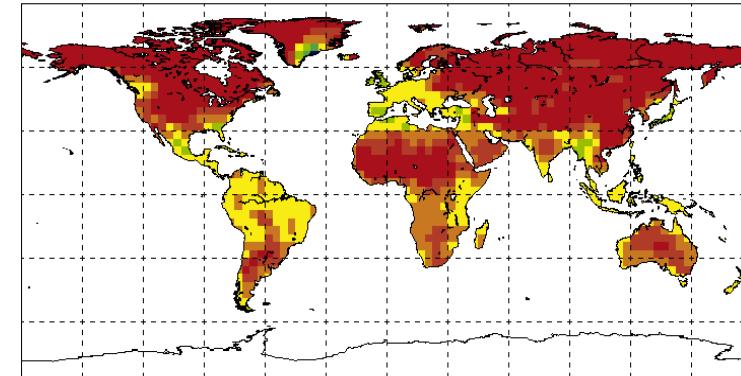


Improving health care and reducing costs



Finding alternative/ green energy sources

CCCma/A2a January to January Mean Temperature (degrees C) 2080s relative to 1961–90



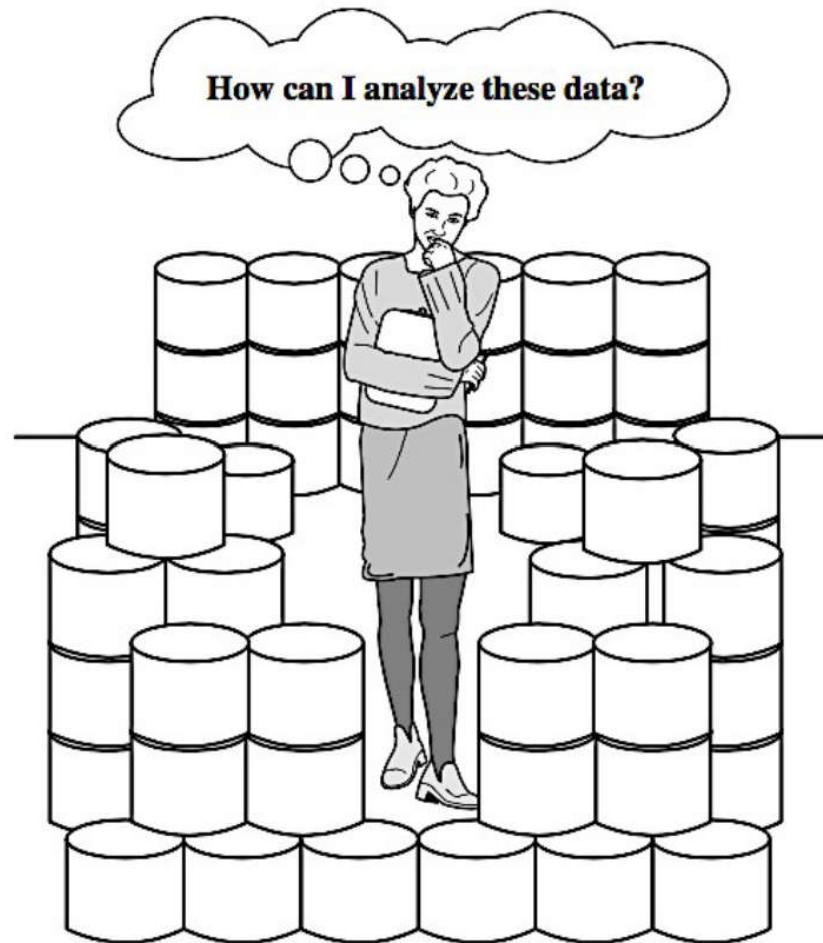
Predicting the impact of climate change



Reducing hunger and poverty by increasing agriculture production

## We Are Drowning In Data, But Starving For Knowledge!

ما در انبوهای از داده‌ها و اطلاعات غرق شدیم در حالی  
که تشنگی دانش هستیم  
داده کاوی: ما یه سری داده داریم که میخاییم ازش  
ارزش افزوده استثبات کنیم  
ما قرار نیست دیتا رو پیدا کنیم ما میخاییم دانش را پیدا  
کنیم (دانش‌های ارزشمند) استخراج میشه



The world is data rich but information poor.

# What is Data Mining?

## ● Many Definitions

- Non-trivial extraction of implicit, previously unknown and potentially useful information from data
- Exploration & analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns.

استخراج غیر ضروری اطلاعات ضمنی، قبل ناشناخته و بالقوه مفید از  
داده ها  
- کاوش و تجزیه و تحلیل، با ابزار های خودکار یا نیمه خودکار، مقادیر  
زیادی داده به منظور کشف الگوهای معنادار.

استخراج آنالیز و کشف اطلاعات از داده  
ها به وسیله ای الگوریتم های اتوماتیک  
و شبه اتوماتیک برای استخراج الگو از  
داده ها

داده کاوی یک اتفاق نیست، یک فرایند  
است در پردازش داده ها ما با فرایندی از  
پردازش روبرو هستیم  
مثل فرایند یا چرخه ای تولید نرم افزار

(Knowledge or Data) Mining!!!



Data mining—searching for knowledge (interesting patterns) in data.

# What is Data Mining?

---

- Data mining,
  - is the **process** of **uncovering patterns** and other **valuable information** from **large data sets**.

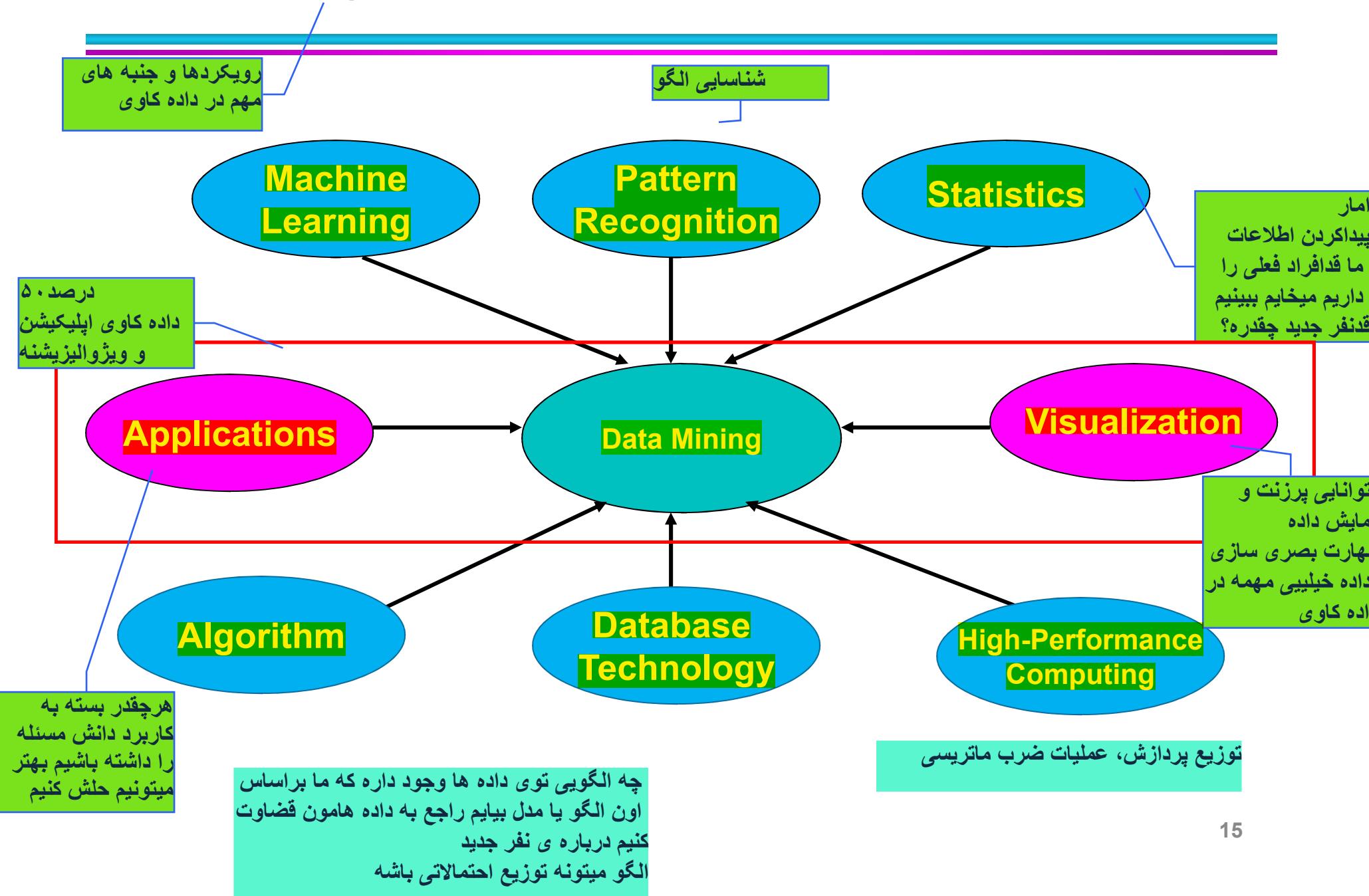
فرآیند کشف الگوهای و سایر اطلاعات ارزشمند از مجموعه داده های بزرگ است

- Alternative names

**Knowledge discovery (mining) in databases (KDD)**,  
**knowledge extraction**, **data/pattern analysis**, **data archeology**, **data dredging**, **information harvesting**,  
**business intelligence**, etc.

نام های دیگر دیتا ماینینگ

# Data Mining: Confluence of Multiple Disciplines



# Human role in data mining?

نقش انسان در پردازش داده ها

- Berry and Linoff, in their 1997 book gave the following definition for data mining:
  - “Data mining is the process of exploration and analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns and rules”.
- Three years later, in their Mastering Data Mining book, they mentioned that,
  - “If there is anything we regret, it is the phrase ‘by automatic or semiautomatic means’ . . . because we feel there has come to be too much focus on the automatic techniques and not enough on the exploration and analysis. This has misled many people into believing that data mining is a product that can be bought rather than a discipline that must be mastered.”

انسان باید به طور فعال در هر مرحله از  
داده کاوی مشارکت داشته باشد

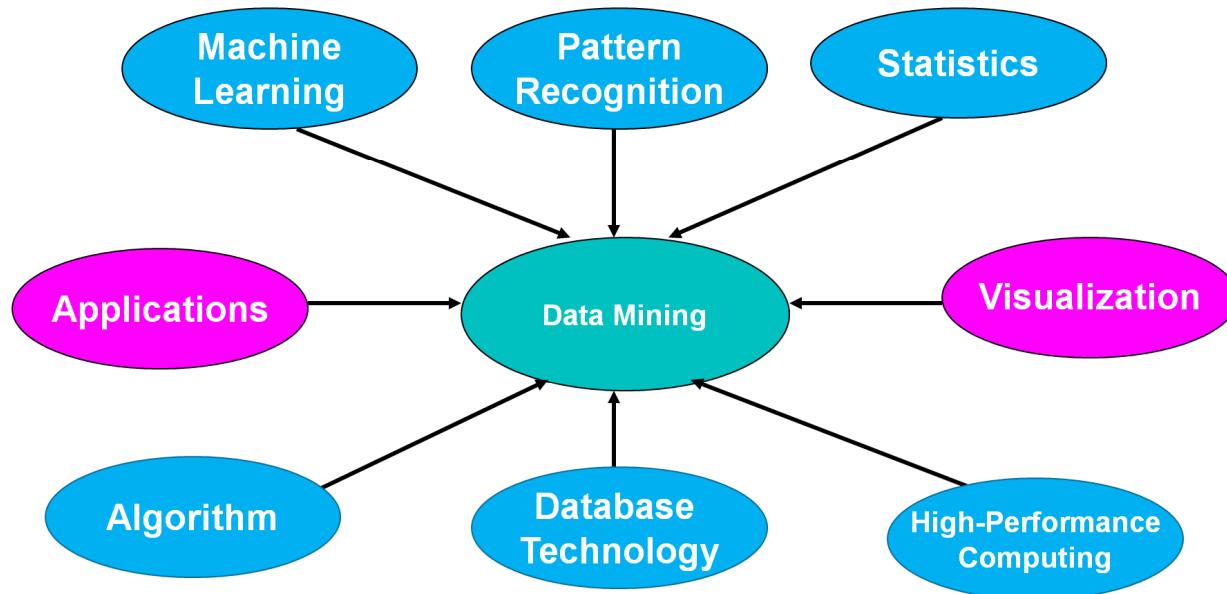
Human need to be actively involve in every phase of data mining.

این موضوع بسیاری از مردم را گمراх کرده است که باور کنند داده کاوی  
محصولی است که می توان خرید کرد نه رشته ای که باید در آن تسلط یافت.

دیتا ماینینگ یک رویه و رویکرد است که  
ارام ارام حاصل میشه و هزینه بر است و  
برای این مهارت ها نیاز به انسان داریم

# Human role in data mining?

---



**Human need to be actively involve in every phase of data mining.**

# Question?

---

---

Your Name, ID, Major

Q1: What do you think Data Mining is?

Q2: What project have you done so far that you think is most relevant to Data Mining?

Not necessarily research project; can be your course project or any hackathon event you participated in.

Q3: What do you expect to learn from this course?

Descriptive data mining and predictive data mining are two different approaches to analyzing data.

Descriptive data mining is the process of analyzing data to describe what has happened in the past. It involves discovering patterns, trends, and relationships in historical data sets. The goal of descriptive data mining is to gain a better understanding of the data and identify insights that can be used to improve decision-making.

Predictive data mining, on the other hand, is focused on using historical data to make predictions about future events or behavior. It involves building models that can be used to forecast outcomes based on historical data. The goal of predictive data mining is to use historical data to uncover patterns and relationships that can be used to make accurate predictions about future events.

In summary, while descriptive data mining helps in understanding what has happened in the past, predictive data mining focuses on predicting what is likely to occur in the future based on historical data.

# Multi-Dimensional View of Data Mining

---

---

- **Data to be mined**
  - Database data (extended-relational, object-oriented, heterogeneous), data warehouse, transactional data, stream, spatiotemporal, time-series, sequence, text and web, multi-media, graphs & social and information networks
- **Knowledge to be mined (or: Data mining functions)**
  - Characterization, discrimination, association, classification, clustering, trend/deviation, outlier analysis, ...
  - Descriptive vs. predictive data mining
  - Multiple/integrated functions and mining at multiple levels
- **Techniques utilized**
  - Data warehousing (OLAP), machine learning, statistics, pattern recognition, visualization, high-performance computing, etc.
- **Applications adapted**
  - Retail, telecommunication, banking, fraud analysis, bio-data mining, stock market analysis, text mining, Web mining, etc.

# Data Mining Tasks

- Prediction Methods

متدها برای پیش‌بینی

- Use some variables to predict unknown or future values of other variables.

- Description Methods

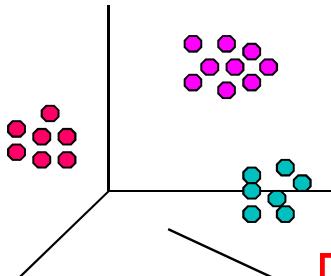
متدها برای توصیف  
یه حجم زیادی از دیتا داریم میخام  
ببینیم تو ش چه خبره؟

- Find human-interpretable patterns that describe the data.

روش های پیش بینی  
- از برخی متغیرها برای پیش بینی مقادیر ناشناخته یا آینده متغیرهای دیگر استفاده کنید.  
روش های توصیف  
- الگوهای قابل تفسیر برای انسان را پیدا کنید که داده ها را توصیف می کند.

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

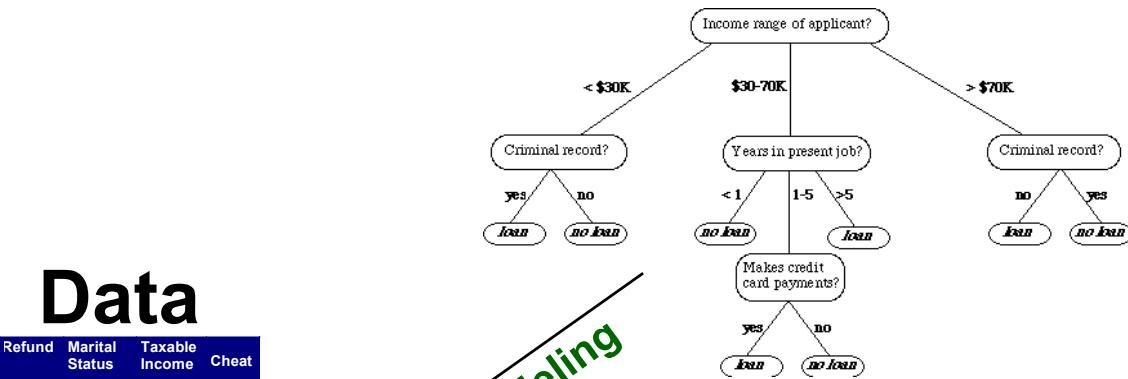
# Data Mining Tasks ...



Clustering

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

Association Rules



Predictive Modeling

Anomaly Detection

# Association Rule Discovery: Definition

---

---

Given a set of records each of which contain some number of items from a given collection

Produce dependency rules

which will

predict occurrence of an item based on occurrences of other items.

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

Rules Discovered:

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

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

# Association Analysis: Applications

## ● Market-basket analysis

- Rules are used for sales promotion, shelf management, and inventory management

حوزه‌ی مخابرات

تحلیل سبد بازار  
- قوانین برای ارتقای فروش، مدیریت قفسه و  
مدیریت موجودی استفاده می‌شود

## ● Telecommunication alarm diagnosis

- Rules are used to find combination of alarms that occur together frequently in the same time period

حوزه‌ی پزشکی

تشخیص دزدگیر مخابراتی  
- از قوانین برای یافتن ترکیبی از آلام‌هایی که  
اغلب در یک دوره زمانی با هم رخ می‌دهند  
استفاده می‌شود

## ● Medical Informatics

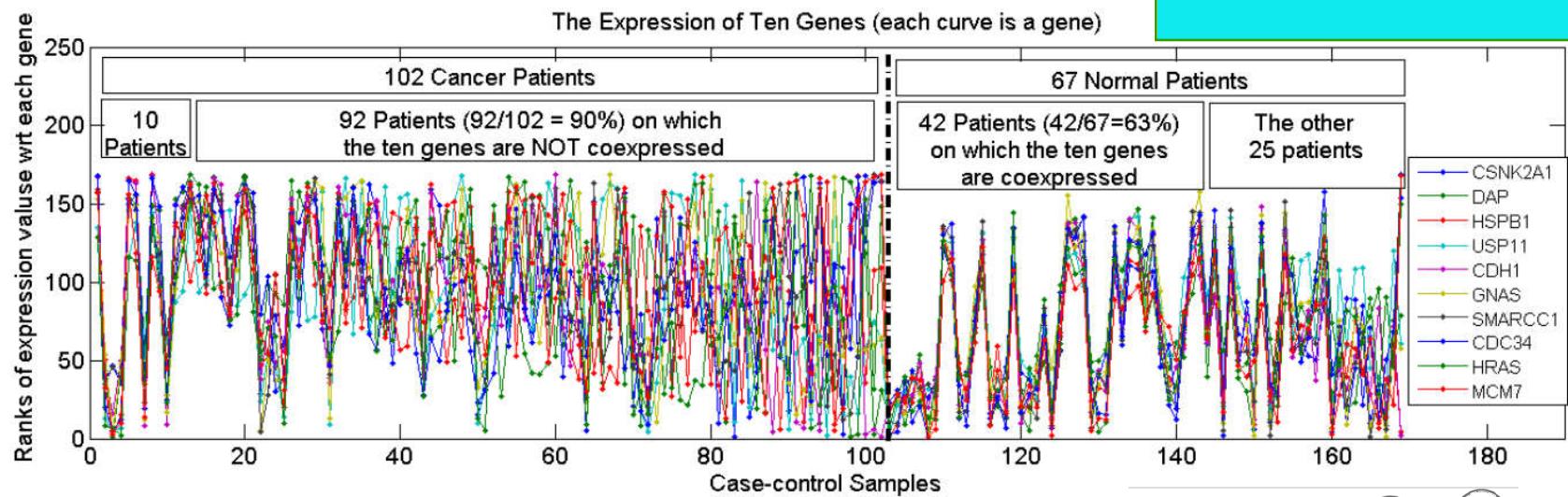
- Rules are used to find combination of patient symptoms and test results associated with certain diseases

انفورماتیک پزشکی  
- قوانین برای یافتن ترکیبی از علائم بیمار و نتایج  
آزمایش مرتبط با بیماری‌های خاص استفاده می‌  
شود

# Association Analysis: Applications

- An Example Subspace Differential Coexpression Pattern from lung cancer dataset

نمونه ای از الگوی هم بیان دیفرانسیل  
زیرفضایی از مجموعه داده سرطان ریه

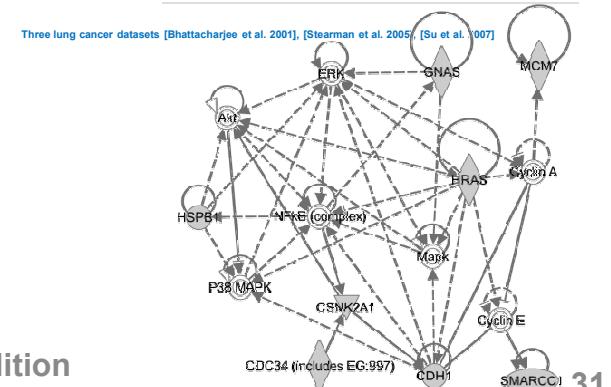


Enriched with the TNF/NFB signaling pathway  
which is well-known to be related to lung cancer  
P-value:  $1.4 \times 10^{-5}$  (6/10 overlap with the pathway)

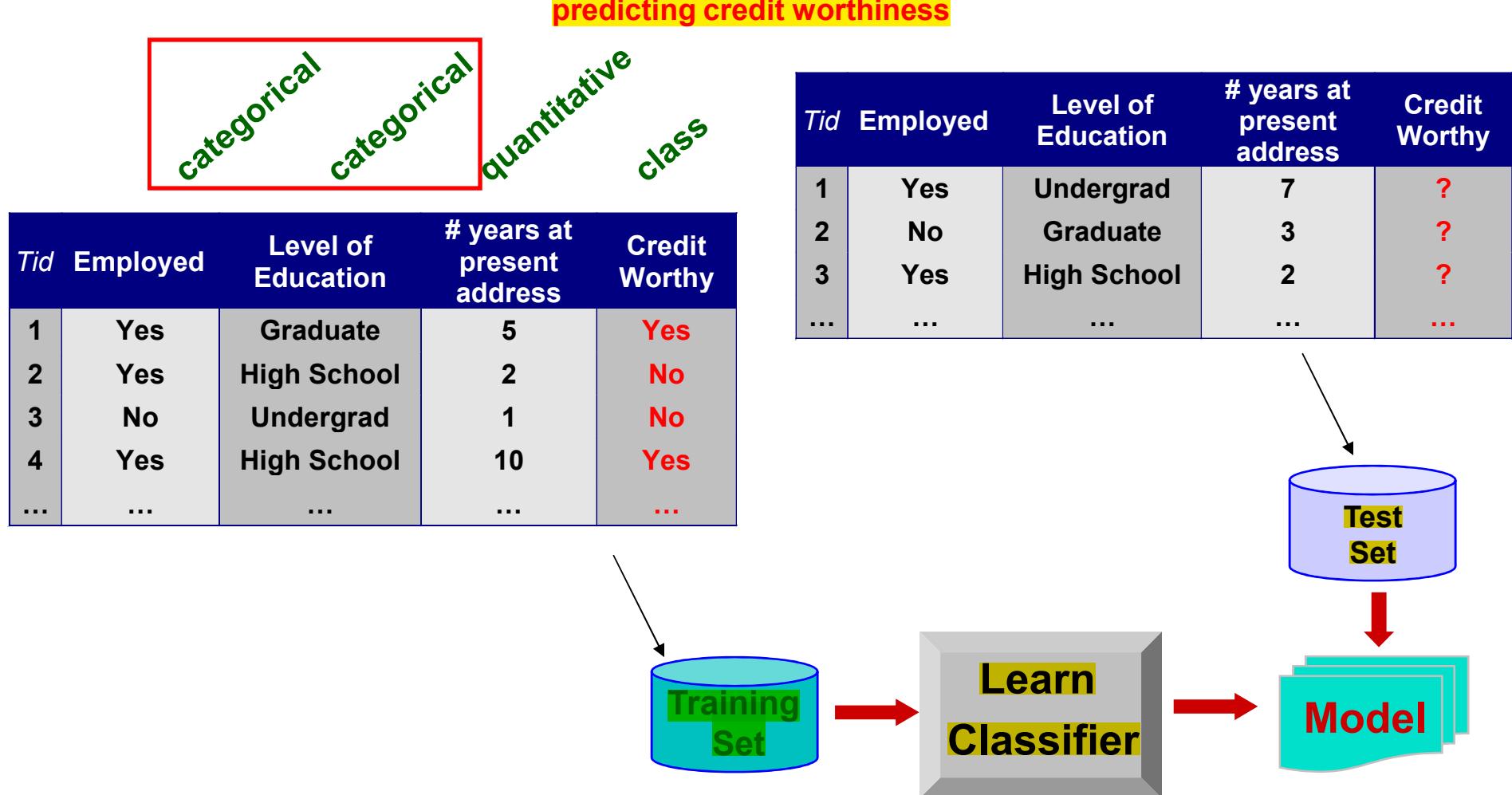
[Fang et al PSB 2010]

09/09/2020

Introduction to Data Mining, 2nd Edition  
Tan, Steinbach



# Classification Example

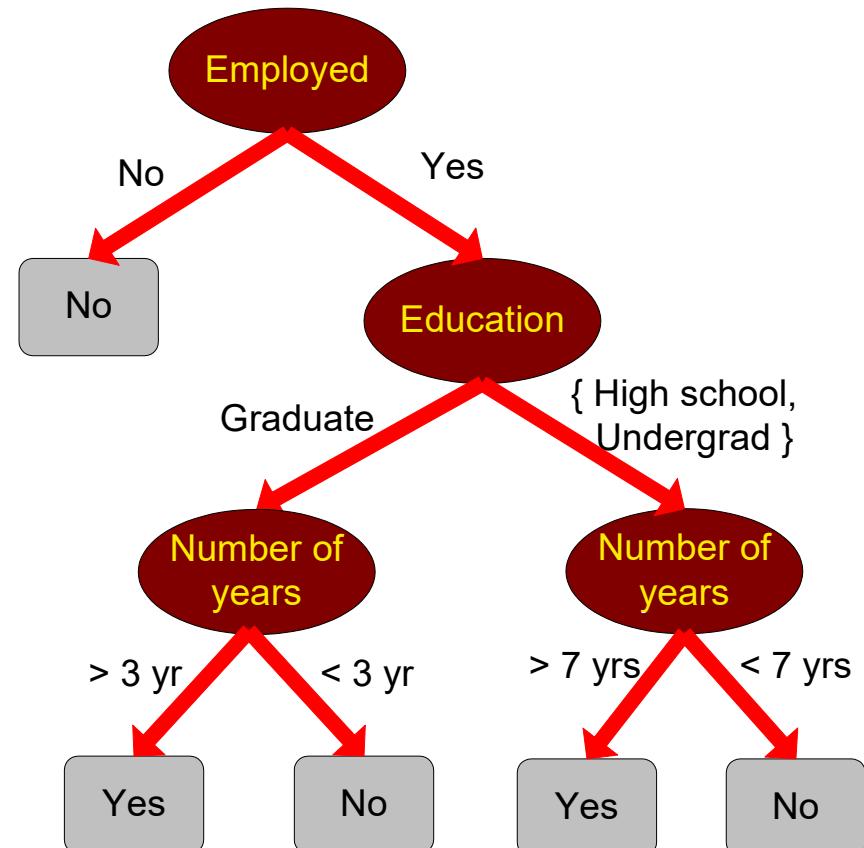


# Predictive Modeling: Classification

- Find a **model** for **class attribute** as a **function** of the values of other attributes

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

Model for predicting credit worthiness



# Classification: Application 1

## Fraud Detection

- **Goal:** Predict **fraudulent cases** in credit card **transactions**.  
پیش بینی موارد کلاهبرداری در معاملات  
کارت اعتباری
- **Approach:**
  - ◆ Use credit card **transactions** and the **information** on its **account-holder** as **attributes**.
    - **When** does a customer buy, **what** does he buy, **how often** he pays on time, etc
  - ◆ Label **past transactions** as **fraud** or **fair** transactions. This forms the **class attribute**.
  - ◆ **Learn a model** for the class of the transactions.
  - ◆ Use this model to **detect fraud** by observing credit card transactions on an account.

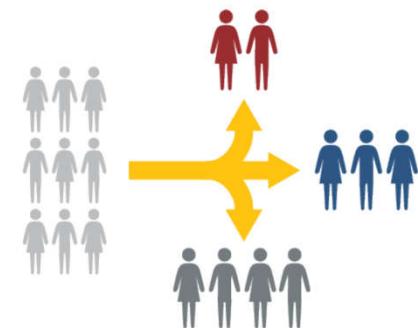


# Classification: Application 2

پیش‌بینی ریزش برای مشتریان تلفن

Churn prediction for telephone customers

- **Goal:** To predict whether a customer is likely to be lost to a competitor.
- **Approach:**
  - ◆ Use detailed record of transactions with each of the past and present customers, to find attributes.
    - How often the customer calls, where he calls, what time-of-the day he calls most, his financial status, marital status, etc.
  - ◆ Label the customers as loyal or disloyal.
  - ◆ Find a model for loyalty.

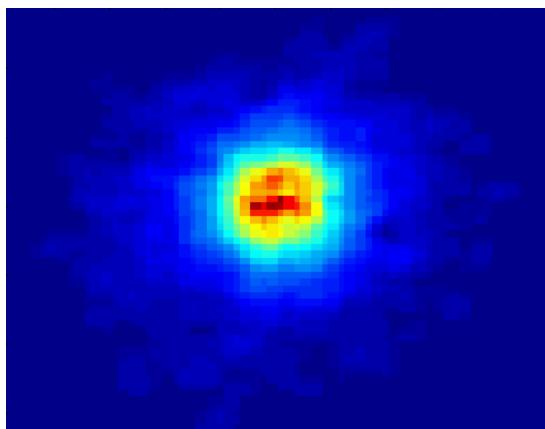


From [Berry & Linoff] Data Mining Techniques, 1997

# Classifying Galaxies

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

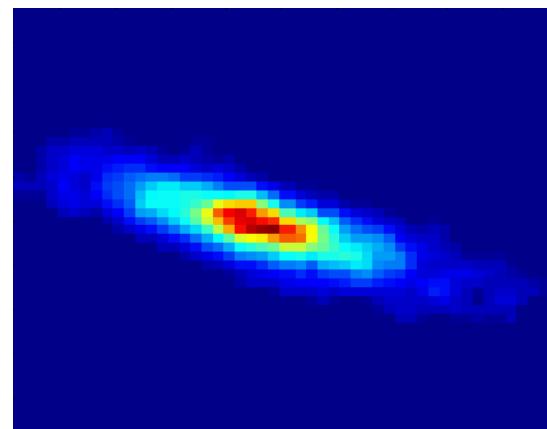
*Early*



**Class:**

- Stages of Formation

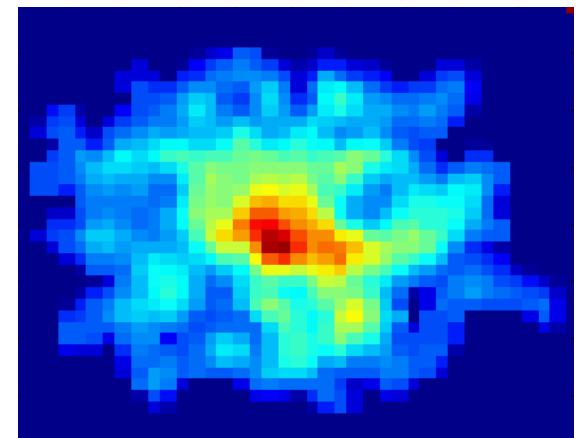
*Intermediate*



**Attributes:**

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

*Late*



**Data Size:**

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

# Classification: Application 3

## Sky Survey Cataloging

- **Goal:** To predict class (star or galaxy) of sky objects, especially visually faint ones, based on the telescopic survey images (from Palomar Observatory).

– 3000 images with 23,040 x 23,040 pixels per image.

- **Approach:**

- ◆ Segment the image.
- ◆ Measure image attributes (features) - 40 of them per object.
- ◆ Model the class based on these features.
- ◆ Success Story: Could find 16 new high red-shift quasars, some of the farthest objects that are difficult to find!

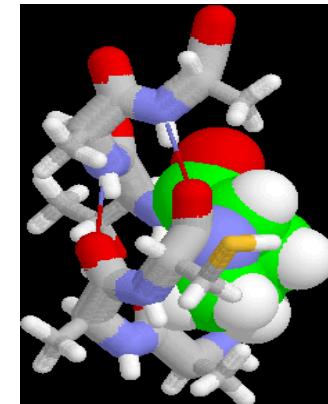
فهرست نویسی بررسی آسمان  
– هدف: پیشینی کلاس (ستاره یا کهکشان) اجرام آسمان، به ویژه آنهایی که از نظر بصری کمتر هستند، بر اساس تصاویر بررسی تلسکوپی (از رصدخانه پالومار)

# Examples of Classification Task

- **Classifying land covers** (water bodies, urban areas, forests, etc.) using satellite data
- **Categorizing news** stories as finance, weather, entertainment, sports, etc
- Identifying intruders in the cyberspace
- Predicting tumor cells as benign or malignant
- **Classifying secondary structures of protein** as alpha-helix, beta-sheet, or random coil



طبقه بندی پوشش های زمین (آب، مناطق شهری، جنگل ها و غیره) با استفاده از داده های ماهواره ای  
طبقه بندی اخبار به عنوان امور مالی، آب و هوا، سرگرمی، ورزش و غیره  
شناسایی مزاحمان در فضای مجازی  
پیش بینی سلول های تومور به عنوان خوش خیم یا بد خیم  
طبقه بندی ساختار های ثانویه پروتئین به عنوان آلفا مارپیچ، بتاسیت یا سیم پیچ تصادفی



# Regression

مسائل پیش‌بینی و  
کار با متغیرهای  
تصادفی پیوسته

قیمت یک خانه  
شاخص الودگی هوا  
گزارش زمان

پیش‌بینی مقدار یک متغیر با ارزش پیوسته بر اساس  
مقادیر سایر متغیرها، (با فرض یک مدل خطی یا  
غیرخطی وابستگی).

- Predict a value of a given continuous valued variable based on the values of other variables, (assuming a linear or nonlinear model of dependency.)
- Extensively studied in statistics, neural network fields.
- Examples:
  - Predicting sales amounts of new product based on advertising expenditure.
  - Predicting wind velocities as a function of temperature, humidity, air pressure, etc.
  - Time series prediction of stock market indices.

پدیده‌هایی که بعد زمانی دارند

سری زمانی قیمت یک کالا  
اگه بخایم پیش‌بینی داشته باشیم درباره‌ی  
اینده‌ی اون کالا مساله‌ی از جنس رگرسن  
است

پیش‌بینی میزان فروش محصول جدید بر اساس هزینه‌ی تبلیغاتی.

- پیش‌بینی سرعت باد به عنوان تابعی از دما، رطوبت، فشار هوا و غیره.

- پیش‌بینی سری زمانی شاخص‌های بورس

# Clustering Task

Finding **groups of objects**

such that the **objects in a group**

will be **similar** (or **related**) to one another and

**different from** (or **unrelated to**) the **objects in other groups.**

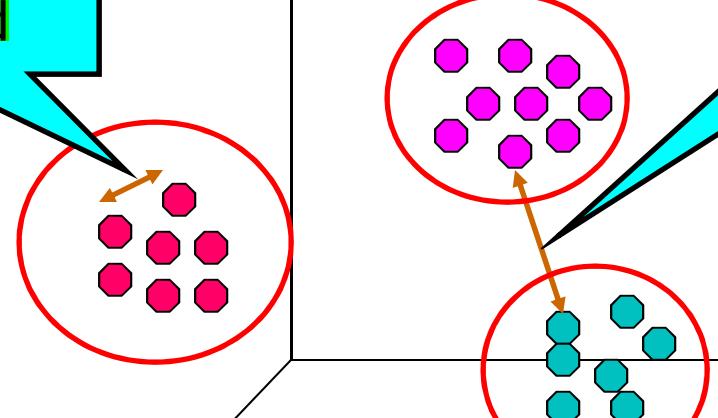
یک شاخصی از فاصله میسازه و تلاش میکنه  
فاصله‌ی اشیایی که توی یک خوشبندی هستند از  
هم کم باشه و فاصله‌ی دو شی در دو خوشبندی  
متفاوت زیاد باشه

فاصله‌ی درون خوشبندی

فاصله‌ی پرون خوشبندی

Intra-cluster  
distances are  
minimized

Inter-cluster  
distances are  
maximized



# Clustering: Application 1

تقسیم بندی بازار:

- هدف: تقسیم بازار به زیرمجموعه های متمایز از مشتریان که در آن هر زیرمجموعه ای ممکن است به عنوان هدف بازار انتخاب شود تا با یک آمیخته بازاریابی مجزا به آن دست یابید.

## Market Segmentation:

- **Goal:** subdivide a market into distinct subsets of customers where any subset may conceivably be selected as a market target to be reached with a distinct marketing mix.
  
- **Approach:**
  - ◆ Collect different attributes of customers based on their geographical and lifestyle related information.
  - ◆ Find clusters of similar customers.
  - ◆ Measure the clustering quality by observing buying patterns of customers in same cluster vs. those from different clusters.

رویکرد: ویژگی های مختلف مشتریان را بر اساس اطلاعات جغرافیایی و سبک زندگی آنها جمع آوری کنید.  
خوشه هایی از مشتریان مشابه را بیابید.  
کیفیت خوشه بندی را با مشاهده الگوهای خرید مشتریان در همان خوشه در مقابل مشتریان خوشه های مختلف اندازه گیری کنید.

# Clustering: Application 2

خوشه بندی اسناد:

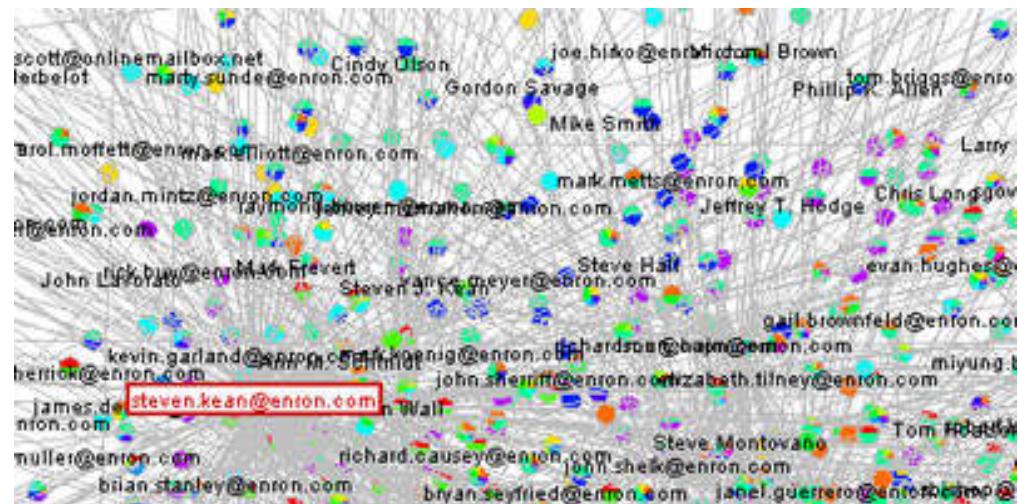
- هدف: یافتن گروه هایی از اسناد مشابه یکدیگر بر اساس اصطلاحات مهم موجود در آنها.

- رویکرد: برای شناسایی اصطلاحات رایج در هر سند. یک معیار تشابه را بر اساس فراوانی اصطلاحات مختلف تشکیل دهد. از آن برای خوشه بندی استفاده کنید.

## Document Clustering:

- **Goal:** To find groups of documents that are similar to each other based on the important terms appearing in them.
- **Approach:** To identify frequently occurring terms in each document. Form a similarity measure based on the frequencies of different terms. Use it to cluster.

Enron email dataset



# Applications of Cluster Analysis

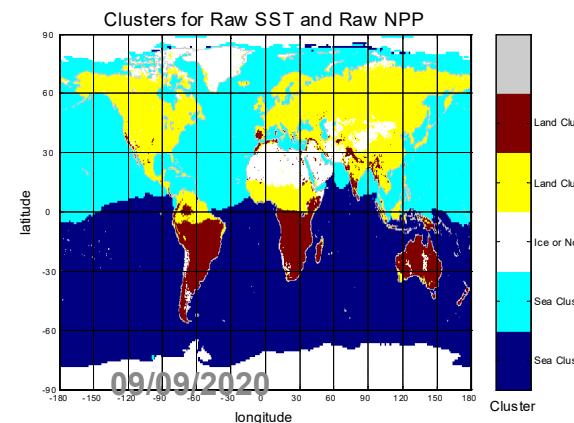
## ● Understanding

- Custom profiling for targeted marketing
- Group related documents for browsing
- Group genes and proteins that have similar functionality
- Group stocks with similar price fluctuations

نمادهای بورس

## ● Summarization

- Reduce the size of large data sets



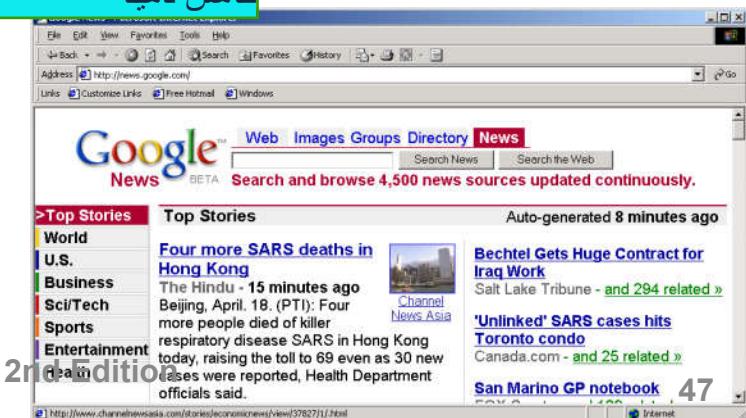
Use of K-means to partition Sea Surface Temperature (SST) and Net Primary Production (NPP) into clusters that reflect the Northern and Southern Hemispheres.

Introduction to Data Mining, 2nd Edition  
Tan, Steinbach



دراگ کردن  
- پروفایل سفارشی برای بازاریابی  
هدفمند  
- گروه بندی اسناد مرتبط برای مرور  
- گروه بندی ژن ها و پروتئین هایی که عملکرد مشابهی دارند  
- گروه بندی سهام با نوسانات قیمتی  
مشابه  
خلاصه سازی  
- اندازه مجموعه داده های بزرگ را کاهش دهد

- hael Eisen



# Deviation/Anomaly/Change Detection

Detect significant deviations from normal behavior

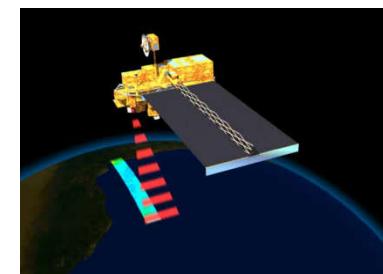
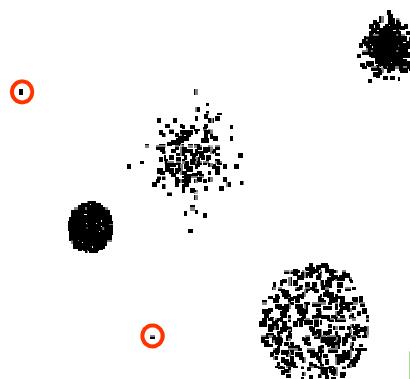
Applications:

- Credit Card Fraud Detection
- Network Intrusion Detection
- Identify anomalous behavior from sensor networks for monitoring and surveillance.
- Detecting changes in the global forest cover.

در یک مساله‌ی دیتماینینگ ما یه سری رکوردهای داده ای داریم که هر رکوردی یه سری اطلاعات به ما میده و قراره به کمک این رکوردها ما یه مساله‌ای رو حل کنیم مثل تایید هویت یک ادم برای اختصاص وام ما سن و تحصیلات و شغل و اطلاعات دیگرش را داشتیم و ستون آخرمون این بود ایا طرف معتبر بوده یا نه پس ما یه چارچوب برای دیتابیس‌مون داریم

تشخیص نفوذ در شبکه‌ها

نواحی غیرعادی در تصاویر ماهواره‌ای



انحرافات قابل توجه از رفتار عادی را تشخیص دهید

برنامه‌های کاربردی:

- تشخیص تقلب در کارت اعتباری
- تشخیص نفوذ شبکه
- شناسایی رفتار غیرعادی از شبکه‌های حسگر برای نظارت.
- تشخیص تغییرات در پوشش جنگلی جهانی

# Major Issues in Data Mining (1)

- **Mining Methodology**

- Mining **various** and **new kinds of knowledge**(New Question)
- Mining knowledge in **multi-dimensional space** (Traffic[#+Speed])
- Data mining: An **interdisciplinary effort** (bug mining)
- Boosting the power of discovery in a **networked environment** (hybrid)
- Handling **noise**, **uncertainty**, and **incompleteness** of data
- Pattern **evaluation** and pattern- or constraint-guided mining  
(ADHD types details)

- **User Interaction**

- **Interactive mining**(Search Engine(Similar))
- Incorporation of **background knowledge**(Multi Judge)
- **Presentation** and **visualization** of data mining results  
(Better Presentations)

روش شناسی کاوش کردن

- استخراج انواع مختلف و جدید دانش (سوال جدید)
- دانش ماینینگ در فضای چند بعدی
- داده کاوی: یک تلاش بین رشته ای (باگ کاوی)
- تقویت قدرت کشف در محیط شبکه ای (هیبرید)
- مدیریت نویز، عدم قطعیت و ناقص بودن داده ها
- ارزیابی الگو مبتنی بر محدودیت
- تعامل با کاربر
- استخراج تعاملی (موتور جستجو (مشابه))
- ترکیب دانش پیشینه (چند داور)
- ارائه و بصری سازی نتایج داده کاوی

# Major Issues in Data Mining (2)

- Efficiency and Scalability

(running time of a data mining algorithm must be predictable)

- Efficiency and scalability of data mining algorithms
- Parallel, distributed, stream, and incremental mining methods(.)

- Diversity of data types

- Handling complex types of data (Time Series)
- Mining dynamic, networked, and global data repositories(Social Network)

- Data mining and society

- Social impacts of data mining (The social Dilemma)
- Privacy-preserving data mining(Fanavard)
- Invisible data mining(Search Engine)

کارایی و مقیاس پذیری  
(زمان اجرای یک الگوریتم داده کاوی باید قابل پیش بینی باشد)  
- کارایی و مقیاس پذیری الگوریتم های داده کاوی  
- روش های استخراج موازی، توزیعی، جریانی و افزایشی  
- تنوع انواع داده ها  
- مدیریت انواع پیچیده داده ها (سری های زمانی)  
- استخراج مخازن داده پویا، شبکه ای و جهانی (شبکه اجتماعی)  
- داده کاوی و جامعه  
- اثرات اجتماعی داده کاوی (معضل اجتماعی!)  
- داده کاوی با حفظ حریم خصوصی (فناورد)  
- داده کاوی نامرئی (موتور جستجو)

## Where to Find References? DBLP, CiteSeer, Google

---

---

Data mining and KDD (SIGKDD)

- Conferences: ACM-SIGKDD, IEEE-ICDM, SIAM-DM, PKDD, PAKDD, etc.
- Journal: Data Mining and Knowledge Discovery, KDD Explorations, ACM TKDD"

Database systems (SIGMOD)

- Conferences: ACM-SIGMOD, ACM-PODS, VLDB, IEEE-ICDE, EDBT, ICDT, DASFAA
- Journals: IEEE-TKDE, ACM-TODS/TOIS, JIIS, J. ACM, VLDB J., Info. Sys., etc. "

AI & Machine Learning

- Conferences: Machine learning (ML), AAAI, IJCAI, COLT (Learning Theory), CVPR, NIPS, etc.
- Journals: Machine Learning, Artificial Intelligence, Knowledge and Information Systems, IEEE-PAMI, etc.

## **Where to Find References? DBLP, CiteSeer, Google**

---

---

Web and IR

Conferences: SIGIR, WWW, CIKM, etc.

Journals: WWW: Internet and Web Information Systems

..

Statistics

Conferences: Joint Stat. Meeting, etc.

Journals: Annals of statistics, etc.

..

Visualization

Conference proceedings: CHI, ACM-SIGGraph, etc.

Journals: IEEE Trans. visualization and computer graphics, etc.

# Exercise2

---

---

- One data mining case example from **Kaggle** in the following format

Case	Feature #	Train Sample #	Test Sample #
------	-----------	----------------	---------------

# The social Dilemma

 **The Social Dilemma**  
2020 · Documentary/Docudrama · 1h 34m

[Overview](#) [Watch movie](#) [Reviews](#) [Cast](#) [Trailers & clips](#) [Quotes](#)

<https://www.thesocialdilemma.com> ::

**The Social Dilemma**  
From the creators of Chasing Ice and Chasing Coral, **The Social Dilemma** blends documentary investigation and narrative drama to disrupt the disrupters, ...  
[The Dilemma](#) · [The Film](#) · [Take a social media reboot](#) · [Take Action](#)

**Cast** >

Tristan Harris Jaron Lanier Skyler Gisondo Tim Kendall  
Ben Kara Hayward Cassandra Sophia Hammons Isla

**Watch movie** [EDIT SERVICES](#)

 Watch now [Subscription](#)  Already watched  Want to watch

**About**

7.6/10 [IMDb](#) 85% [Rotten Tomatoes](#) 78% [Metacritic](#)

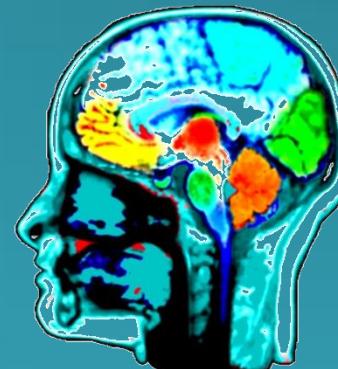
93% liked this film    
Google users

Tech experts from Silicon Valley sound the alarm on the dangerous impact of social networking, which Big



# Introduction To Data Mining

Isfahan University of Technology (IUT)  
Bahman 1401



## Review of Probability Theory

Dr. Hamidreza Hakim  
[hamid.hakim.u@gmail.com](mailto:hamid.hakim.u@gmail.com)

# Content

---

---

Elements of Probability

Random variables

Two random variables

Bayes Rule

The sample space is the set of all possible outcomes of a random experiment, while the event space is the set of all possible events that can occur in the experiment. In other words, the sample space contains all the individual outcomes that could happen, while the event space consists of sets of those outcomes that we might be interested in.

For example, let's consider the experiment of flipping a coin once. The sample space is {heads, tails}, which represents all possible outcomes of the coin flip. The event space consists of all possible subsets of the sample space, including:

The event "getting heads", which consists of the subset {heads}.

The event "getting tails", which consists of the subset {tails}.

The event "getting either heads or tails", which consists of the entire sample space {heads, tails}.

The event "not getting heads", which is the complement of the event "getting heads" and consists of the subset {tails}.

As you can see, the event space is simply a collection of subsets of the sample space, each representing a particular event that we might be interested in.

Sure, here's an example of probability:

Suppose we roll a fair six-sided die. The sample space is the set of all possible outcomes, which in this case is {1, 2, 3, 4, 5, 6}. The event space consists of all subsets of the sample space. For example, the event "rolling an even number" consists of the subset {2, 4, 6}.

Now, let's say we want to calculate the probability of rolling a 3 or higher. This corresponds to the event {3, 4, 5, 6}. There are four outcomes in this event, and six possible outcomes overall, so the probability of rolling a 3 or higher is 4/6 or 2/3.

Another example of an event might be "rolling a prime number", which consists of the subset {2, 3, 5}. The probability of rolling a prime number is 3/6 or 1/2.

# Elements of Probability

احتمال یه تابع است که  
تاویژگی زیر را داشته  
باشه: تابعی که بتوانه  
فضای حالت هامون را به  
مقداری ببره که همیشه  
مثبت است

- Sample space  $\Omega$ : the set of all the outcomes of an experiment

فضای نمونه ای  
مثلثا برای قد انسان ها  
میشه یه متغیر پیوسته

- Event space  $F$ : a collection of possible outcomes of an experiment.  $F \subseteq \Omega$ .

- Probability measure: a function  $P: F \rightarrow R$  that satisfies the following properties:

فضای پیشامد  
میشه گفت یه  
زیرمجموعه ای از  
فضای نمونه است  
مثلابازه ای خاصی  
از قد انسان ها

- $P(A) \geq 0 \forall A \in F$
- $P(\Omega) = 1$
- If  $A_1, A_2, \dots$  are disjoint events, then

که همه ای اعضای مجموعه مون  
وی ورودی اون تابع باشه، مقدار  
یک را به ما بده

انداختن تاس، ۶تا حالت داره و گستته  
است و سکه انداختن دوتا حالت داره پس  
همه ای حالت هایی که از مایش ما میتوانه  
داشته باشه میشه فضای نمونه ای

احتمال مجموع

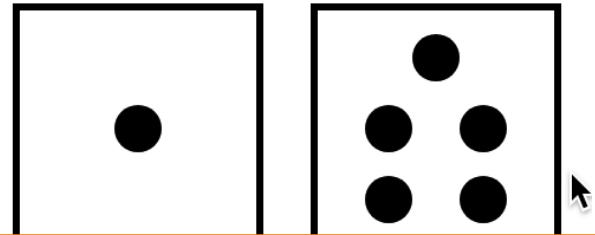
$$P(\bigcup_i A_i) = \sum_i P(A_i)$$

مجموع احتمال ها

اگه راجع به تکه های  
مختلف مجموعه های حرف  
بزنیم مجموع احتمال اونها  
ا احتمال مجموع اونها یکی  
پشه

# Elements of Probability(Example)

- tossing a six-sided die
- Measure human Color



let's consider the experiment of rolling two fair six-sided dice. The sample space consists of all ordered pairs of dice rolls, which is given by:

$\{(1,1), (1,2), (1,3), (1,4), (1,5), (1,6), (2,1), (2,2), (2,3), (2,4), (2,5), (2,6), (3,1), (3,2), (3,3), (3,4), (3,5), (3,6), (4,1), (4,2), (4,3), (4,4), (4,5), (4,6), (5,1), (5,2), (5,3), (5,4), (5,5), (5,6), (6,1), (6,2), (6,3), (6,4), (6,5), (6,6)\}$

The event space consists of all possible subsets of the sample space. For example:

The event "getting a sum of 7" consists of the following subsets:  $\{(1,6), (2,5), (3,4), (4,3), (5,2), (6,1)\}$ .

The event "getting doubles" (i.e., both dice showing the same number) consists of the following subsets:  $\{(1,1), (2,2), (3,3), (4,4), (5,5), (6,6)\}$ .

The event "getting a total greater than 9" consists of the following subsets:  $\{(4,6), (5,5), (5,6), (6,4), (6,5), (6,6)\}$ .

As you can see, the event space for this experiment can be quite large and complex, since there are many possible combinations of dice rolls and many different types of events that we might be interested in.

# Properties of Probability

---



---

- If  $A \subseteq B \Rightarrow P(A) \leq P(B)$
- $P(A \cap B) \leq \min(P(A), P(B))$
- $P(A \cup B) \leq P(A) + P(B)$  (Union Bound)
- $P(\Omega \setminus A) = 1 - P(A)$ 

مکمل زمان هایی که تاس ۲  
میاد مثلما
- If  $A_1, \dots, A_k$  is a disjoint partition of  $\Omega$ , then

مکمل فضا

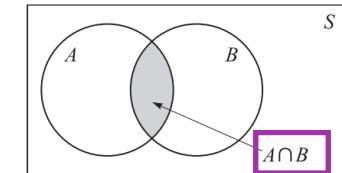
$$\sum_{i=1}^k P(A_i) = 1$$

# Conditional Probability

احتمال شرطی

- A conditional probability  $P(A|B)$  measures the probability of an event A after observing the occurrence of event B

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



- Two events  $A$  and  $B$  are independent iff  $P(A|B) = P(A)$  or equivalently,  $P(A \cap B) = P(A)P(B)$

دو پیشامد نسبت به هم  
مستقل هستند

# Conditional Probability(Examples)

---

- A math teacher gave her class two tests. 25% of the class passed both tests and 42% of the class passed the first test. What percent of those who passed the first test also passed the second test?
- In New England, 84% of the houses have a garage and 65% of the houses have a garage and a back yard. What is the probability that a house has a backyard given that it has a garage?

# Independent Events Examples

---

- What's the probability of getting a sequence of 1,2,3,4,5,6 if we roll a dice six times?
- A school survey found that 9 out of 10 students like pizza. If three students are chosen at random with replacement, what is the probability that all three students like pizza?

# Random Variable

A **random variable  $X$**  is a **function** that maps a sample space  $\Omega$  to **real values**. Formally,

$$X : \Omega \longrightarrow R$$

Examples:

- Rolling **one** dice

$X$  = **number** on the dice at each roll

- Rolling **two dice** at the **same time**

$X$  = **sum** of the two numbers

درباره ی اون مقداری که قابل اندازه  
گیری است صحبت کنیم

اگه یه سکه را ۱۰ بار انداختم چند بارش رو  
مید چند بار پشت؟ اینجا داریم درباره یه  
چیز کمی صحبت میکنیم که مقدار داره  
در این زمان ها سراغ متغیرهای تصادفی  
میریم  
مثال : تعداد دفعاتی که یک سکه رو میاد  
برای ۱۰ بار انداختن  
یا تعداد دفعاتی که تاس زوج میاد در ۶ بار  
انداختن  
تعداد دانش اموزانی که قدشان از یه حدی  
بیشتر است

# Random Variable

A random variable can be continuous. E.g.,

- $X$  = the length of a randomly selected phone call  
(What's the  $\Omega$ ?)
- $X$  = amount of coke left in a can marked 12oz  
(What's the  $\Omega$ ?)

تعریف طول شماره تلفن ها به عنوان متغیر تصادفی ثلا طول شماره تلفن های همراه و ثابت متفاوت است پس چندتا حالت داریم

# Probability Mass Function

If  $X$  is a **discrete random variable**, we can specify a probability for **each** of its **possible values** using the probability mass function (**PMF**). Formally, a **PMF** is a **function**  $p: \Omega \rightarrow R$  such that

احتمال رخدادن اینکه تاس مثلا  
عدد یک بیاد یک ششم است که  
پس پی ام اف میگیم

$$p(x) = P(X = x)$$

میخایم پدیده‌ی تصادفی را  
بیشتر بشناسیم  
اگه متغیر تصادفی همیشه  
مقادیر گستته داشته باشه  
همیشه یه پی ام اف هم  
خواهد داشت

- Rolling a dice:

$$p(X = i) = \frac{1}{6} \quad i = 1, 2, \dots, 6$$

متغیر تصادفی: مقادیری  
که به ازای اندختن تاس  
بدست میاریم باشه

- Rolling two dice at the **same time**:

$X = \text{sum}$  of the two numbers

$$p(X = 2) = \frac{1}{36}$$

اگه متغیر تصادفی مان را جمع مقادیری  
بگیریم که دوتاس در دوبار پرتاب میگیریم

:pmf  
احتمال رخدادن یه حالتی  
از متغیر تصادفی را میگه

A Probability Mass Function (PMF) is a function that describes the probability distribution of a discrete random variable. It assigns a probability to each possible value that the random variable can take on. Here are some examples of PMFs:

Fair Coin Toss: Suppose we toss a fair coin once and let X be the number of heads that come up. The possible values for X are 0 and 1, and the PMF is given by:

$$P(X=0) = 1/2$$

$$P(X=1) = 1/2$$

This PMF tells us that there is a 50% chance of getting zero heads and a 50% chance of getting one head when we toss a fair coin once.

Rolling a Die: Suppose we roll a six-sided die once and let Y be the number that comes up. The possible values for Y are 1, 2, 3, 4, 5, and 6, and the PMF is given by:

$$P(Y=1) = 1/6$$

$$P(Y=2) = 1/6$$

$$P(Y=3) = 1/6$$

$$P(Y=4) = 1/6$$

$$P(Y=5) = 1/6$$

$$P(Y=6) = 1/6$$

This PMF tells us that each of the six possible outcomes has an equal chance of occurring when we roll a fair die once.

Drawing Marbles from a Bag: Suppose we have a bag with three red marbles and two blue marbles. We draw one marble from the bag without replacement and let Z be the color of the marble we draw (where R represents red and B represents blue). The possible values for Z are R and B, and the PMF is given by:

$$P(Z=R) = 3/5$$

$$P(Z=B) = 2/5$$

This PMF tells us that there is a 60% chance of drawing a red marble and a 40% chance of drawing a blue marble when we draw one marble from the bag without replacement.

These are just a few examples of PMFs. In general, the PMF can be used to describe the probability distribution of any discrete random variable.

To identify the random variable in a probability problem, you need to look for the quantity that can take on different values with some probability. A random variable is a mathematical representation of this quantity. Here are some steps to follow to identify the random variable in a probability problem:

Read the problem statement carefully and identify what is being measured or observed.

Determine which of these quantities varies from trial to trial or observation to observation.

Assign a symbol to represent the quantity identified in step 2. This symbol should be capitalized to denote that it represents a random variable (e.g., X, Y, Z).

Define the range of possible values that the random variable can take on. This range should include all possible values that the quantity can take on, even if some values have zero probability.

Determine the probability distribution of the random variable. This involves assigning probabilities to each possible value that the random variable can take on.

For example, consider the following problem: A fair six-sided die is rolled. What is the probability of rolling an even number?

In this problem, the quantity being measured is the number rolled on the die. The quantity varies from trial to trial, so it is a random variable. We can represent this random variable with the symbol X.

The range of possible values for X is {1, 2, 3, 4, 5, 6}. Since the die is fair, each of these values has probability 1/6.

The probability distribution of X is given by:

$$P(X=1) = 1/6$$

$$P(X=2) = 1/6$$

$$P(X=3) = 1/6$$

$$P(X=4) = 1/6$$

$$P(X=5) = 1/6$$

$$P(X=6) = 1/6$$

Therefore, we can say that X is a discrete random variable with a uniform distribution.

In summary, to identify the random variable in a probability problem, you need to look for the quantity that can take on different values with some probability and assign a symbol to represent it.

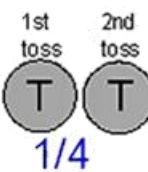
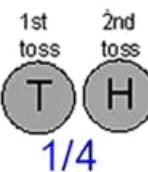
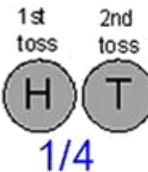
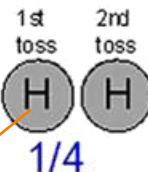


# Probability Mass Function(Examples)

- X: Number of Tail

متغیر تصادفی مان اینجا  
عدد پشت یا تیل هایی که  
سکه میار

سکه ای اول یا سکه ای  
دوم پشت ببین دو تا حالت  
میشه



X=0

X=1

X=2

pmf  
یه تابعی از جنس احتمال است  
احتمال رخداد یه متغیر تصادفی  
را گزارش میکنه

نوشتن حالت های متغیر تصادفی  
و تعداد رخدادهای اون حالت از  
متغیر تصادفی را مینویسیم تووش

List of possible  
values

X

0

1

2

Probability of  
each value

P(X=x)

X	0	1	2
P(X=x)	1/4	1/2	1/4

# Probability Mass Function(Examples)

---

- X be the number of tails in Flipping a Coin Three Times

Outcome	Probability	X
HHH	$1/2 * 1/2 * 1/2 = 1/8$	0
HHT	1/8	1
HTH	1/8	1
THH	1/8	1
HTT	1/8	2
THT	1/8	2
TTH	1/8	2
TTT	1/8	3

# Probability Mass Function(Examples)

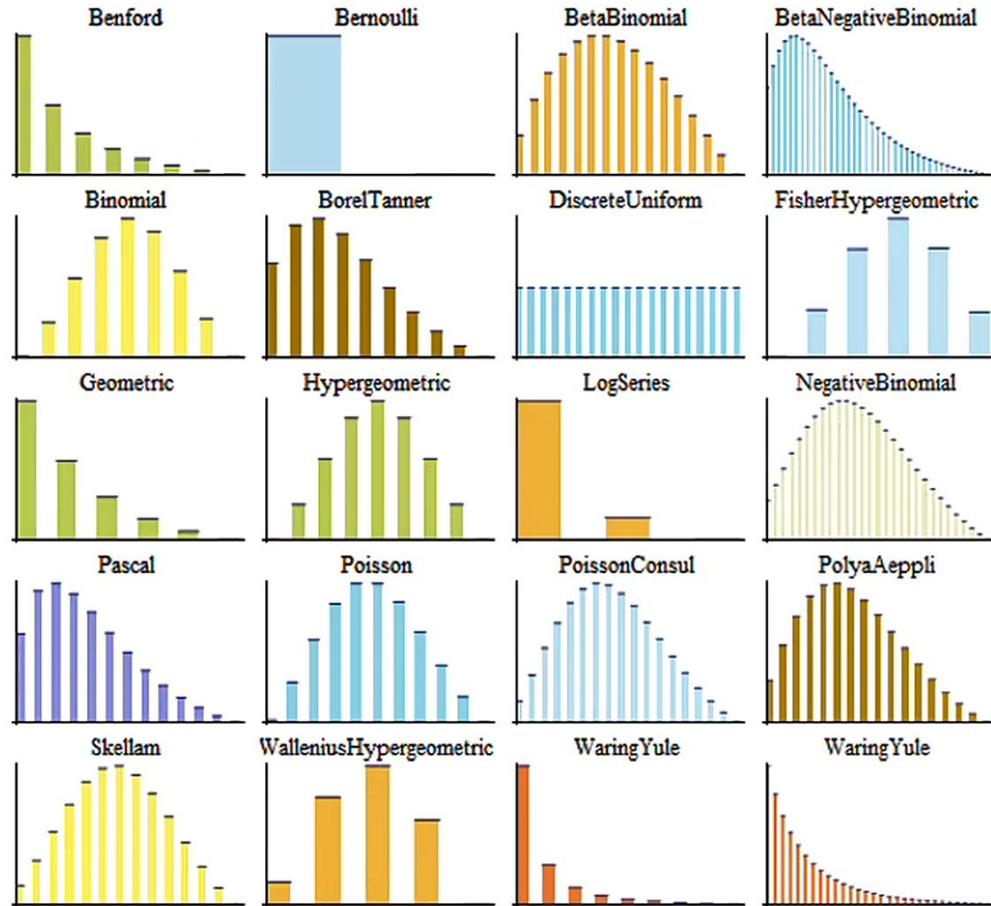
- $X$  be the number of tails in Flipping a Coin Three Times

Outcome	Probability	$X$	
HHH	1/8	0	1/8
HHT	1/8	1	$1/8 + 1/8 = 2/8$
HTH	1/8	1	$1/8 + 1/8 = 2/8$
THH	1/8	1	$1/8 + 1/8 = 2/8$
HTT	1/8	2	$1/8 + 1/8 = 2/8$
THT	1/8	2	$1/8 + 1/8 = 2/8$
TTH	1/8	2	$1/8 + 1/8 = 2/8$
TTT	1/8	3	1/8

# Probability Mass Function

X	$x_1$	$x_2$	$x_3$	...	$x_n$
$P(X=x)$	$p_1$	$p_2$	$p_3$	...	$p_n$

محور ایکس مقادیر متغیر تصادفی است  
و محور وای احتمال رخداد اون متغیر  
است



# Probability Mass Function

---

- $X \sim Bernoulli(p)$ ,  $p \in [0, 1]$

$$p(x) = \begin{cases} p & \text{if } x = 1 \\ 1 - p & \text{if } x = 0 \end{cases}$$

- $X \sim Binomial(n, p)$ ,  $p \in [0, 1]$  and  $n \in Z^+$

$$p(x) = \binom{n}{x} p^x (1 - p)^{n-x}$$

- $X \sim Geometric(p)$ ,  $p > 0$

$$p(x) = p(1 - p)^{x-1}$$

- $X \sim Poisson(\lambda)$ ,  $\lambda > 0$

$$p(x) = e^{-\lambda} \frac{\lambda^x}{x!}$$

# Probability Density Function

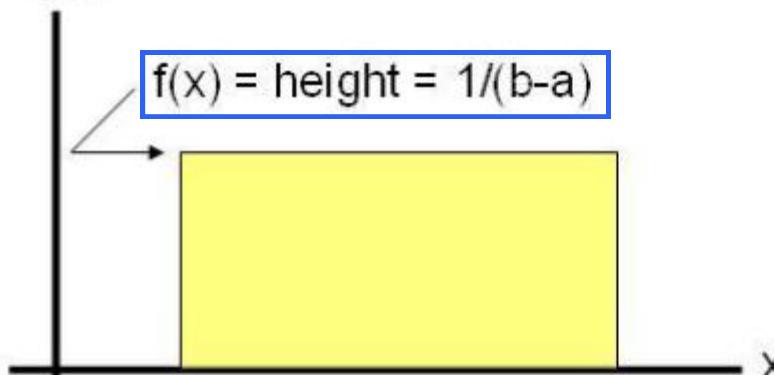
- If  $X$  is a **continuous** random variable, we can NOT specify a probability for each of its possible values (why?)
- We use a **probability density function  $PDF$**  to describe the **relative likelihood** for a random variable to take on a given value
- A ( $PDF$ ) specifies the **probability of  $X$  takes a value within a range.** Formally, a  $PDF$  is a function  $f(x): \Omega \rightarrow R$  such that

$$P(a < X < b) = \int_a^b f(x)dx$$

درفضای پیوسته ما راجع به مقدار دقیق یه عدد حرف نمیزیم بحث های حد داره  
مثلًا اینکه احتمال اینکه قد بین ۱۶۰ تا ۱۸۰ باشه  
چقدر؟ باید یه بازه بش بدیم

# Probability Density Function

- $X \sim \text{uniform on } [a, b]:$

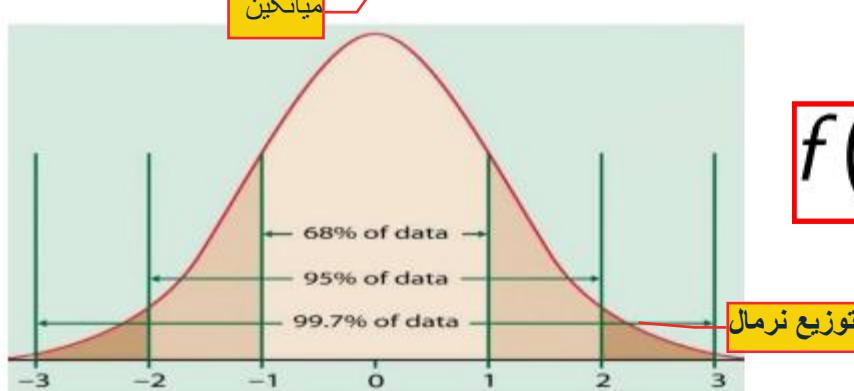


توزیع به صورت یکنواخت است یعنی همهٔ حالت‌های متغیر تصادفی شانس برابر دارند یعنی احتمال تک تک این مقادیر یکسان است

$$f(x) = \frac{1}{b-a}$$

فقط ابتدا و انتهای بازه را بش میدیم خودش ارتفاع را میده

- $X \sim N(\mu, \sigma^2):$



$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

# Expected value of Random Variable

مقدار امیدریاضی یک متغیر یا  
مقدار مورد انتظار ما

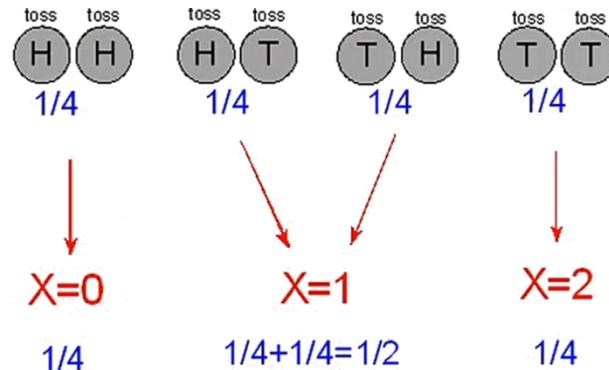
از جنس احتمال نیست!!!  
از جنس همون متغیریه  
که داریم دربارش حرف  
میزندیم

$$E(X) = \sum_x xf(x)$$

ضرب احتمال در مقادیر  
رخداد و جمع اینها باهم

$$E(X) = \int_{-\infty}^{\infty} xf(x)dx$$

مقدار متوسط پشت اومدن دوتا سکه؟



List of possible values	x	0	1	2
Probability of each value	$P(X=x)$	1/4	1/2	1/4

# Variance of Random Variable

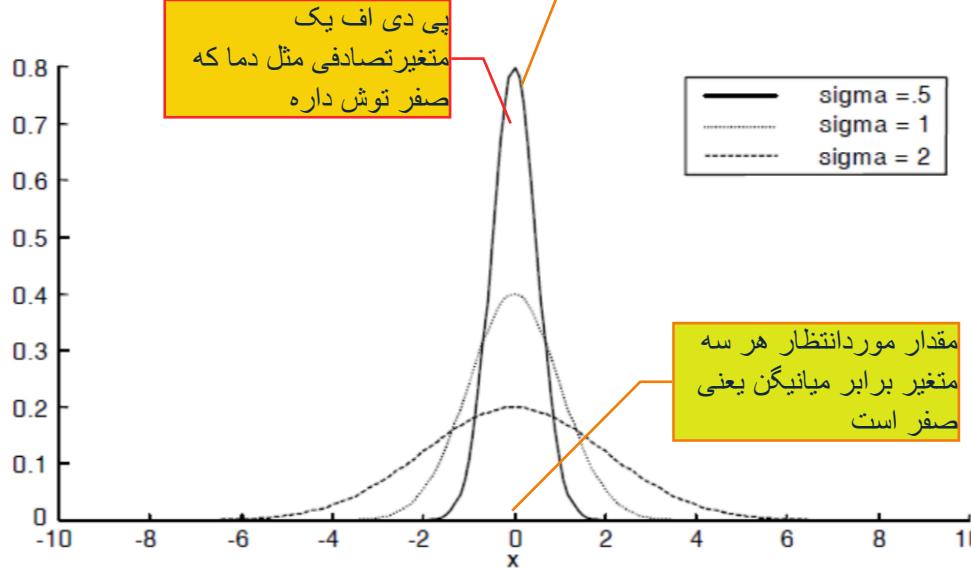
واریانس یه متغیر تصادفی  
میشه پراکندگی اون متغیر

$$Var(X) = \sigma^2 = E[(X - \mu)^2]$$

مقادیر متغیر تصادفی چه  
طیفی داره؟  
قدارهای متغیر تصادفی را  
نهای میانگینشون میکنه و  
بعد امید ریاضیشون را  
حساب میکنه

توزیع نرمال

- $\sigma$  = Deviation Standard



Expected value is a concept in probability theory that represents the long-term average of a random variable over many repeated trials. It is calculated by multiplying each possible outcome of the random variable with its corresponding probability, and then summing up these products.

For example, imagine you are rolling a fair six-sided die. The possible outcomes are the numbers 1 to 6, each with probability 1/6. The expected value of this random variable can be calculated as follows:

$$\begin{aligned} E(X) &= (1/6)*1 + (1/6)*2 + (1/6)*3 + (1/6)*4 + (1/6)*5 + (1/6)*6 \\ &= 3.5 \end{aligned}$$

So the expected value of rolling a fair six-sided die is 3.5. This means that if you were to roll the die many times, the average of all the rolls would tend towards 3.5.

Variance is another important concept in probability theory that measures how much a random variable varies from its expected value. It is calculated by taking the difference between each possible outcome of the random variable and the expected value, squaring this difference, multiplying it by the corresponding probability, and then summing up these products.

In mathematical notation, the variance of a random variable X can be expressed as follows:

$$\text{Var}(X) = E[(X - E(X))^2]$$

where  $E(X)$  is the expected value of X.

For example, let's say we are rolling a fair six-sided die again. We have already calculated that the expected value of this random variable is 3.5. Now let's calculate the variance of this random variable.

The possible outcomes are the numbers 1 to 6, each with probability 1/6. Using the formula above, we can calculate the variance as follows:

$$\begin{aligned} \text{Var}(X) &= (1/6)(1-3.5)^2 + (1/6)(2-3.5)^2 + (1/6)(3-3.5)^2 + (1/6)(4-3.5)^2 + (1/6)(5-3.5)^2 + (1/6)(6-3.5)^2 \\ &= 35/12 \end{aligned}$$

So the variance of rolling a fair six-sided die is approximately 2.92. This means that if you were to roll the die many times, the results would tend to fluctuate around the expected value of 3.5, with some rolls being higher and some lower. The variance gives us a measure of how much this fluctuation is likely to be.

Suppose there are two factories that produce a certain type of product. Factory A produces 60% of the total output and has a defect rate of 5%. Factory B produces the remaining 40% and has a defect rate of 3%.

Now suppose a customer buys a product and discovers that it is defective. What is the probability that the product was produced by Factory A?

We can use Bayes' rule to calculate this probability. Let A be the event that the product was produced by Factory A, and let B be the event that the product is defective. Then we want to find  $P(A|B)$ , the probability that the product was produced by Factory A given that it is defective.

Bayes' rule states:

$$P(A|B) = P(B|A) * P(A) / P(B)$$

where  $P(B)$  is the total probability of the product being defective, which can be calculated by considering the two possible ways a defective product could be produced: either it was produced by Factory A and is defective, or it was produced by Factory B and is defective.

So we have:

$$P(B) = P(B|A) * P(A) + P(B|\text{not } A) * P(\text{not } A)$$

where  $P(\text{not } A) = 1 - P(A)$  is the probability that the product was produced by Factory B.

Using the numbers given, we can fill in the values:

$$\begin{aligned} P(B) &= 0.05 * 0.6 + 0.03 * 0.4 \\ &= 0.036 \end{aligned}$$

Now we can use Bayes' rule to find  $P(A|B)$ :

$$\begin{aligned} P(A|B) &= P(B|A) * P(A) / P(B) \\ &= 0.05 * 0.6 / 0.036 \\ &= 0.833 \text{ or about } 83.3\% \end{aligned}$$

So the probability that the defective product was produced by Factory A is about 83.3%, even though Factory B has a lower defect rate. This shows why it's important to consider not only the defect rates of the factories, but also their production volumes when investigating defects.

Suppose there is a certain medical test for a particular disease. The test is not perfect and gives a false positive 5% of the time (i.e., if a person does not have the disease, the test will incorrectly indicate that they do, with probability 0.05) and a false negative 1% of the time (i.e., if a person does have the disease, the test will incorrectly indicate that they do not, with probability 0.01).

Suppose also that this disease occurs in the population at a rate of 0.2% (i.e., 0.002 of the population has the disease).

Now, suppose we want to know the probability that a person who tests positive actually has the disease.

We can use Bayes' rule to calculate this probability. Let A be the event that a person has the disease, and let B be the event that the person tests positive. Then we want to find  $P(A|B)$ , the probability that the person has the disease given that they tested positive.

Bayes' rule states:

$$P(A|B) = P(B|A) * P(A) / P(B)$$

where  $P(B)$  is the total probability of testing positive, which can be calculated by considering the two possible ways a person could test positive: either they actually have the disease and the test correctly identifies it, or they do not have the disease but the test mistakenly indicates that they do.

So we have:

$$P(B) = P(B|A) * P(A) + P(B|\text{not } A) * P(\text{not } A)$$

where  $P(\text{not } A) = 1 - P(A)$  is the probability that a person does not have the disease.

Using the numbers given, we can fill in the values:

$$\begin{aligned} P(B) &= 0.99 * 0.002 + 0.05 * 0.998 \\ &= 0.05286 \end{aligned}$$

Now we can use Bayes' rule to find  $P(A|B)$ :

$$\begin{aligned} P(A|B) &= P(B|A) * P(A) / P(B) \\ &= 0.99 * 0.002 / 0.05286 \\ &= 0.0746 \text{ or about } 7.5\% \end{aligned}$$

So the probability that a person who tests positive actually has the disease is only about 7.5%, even though the test is 95% accurate (i.e., has a 95% sensitivity and 95% specificity). This shows why it's important to consider the base rate of the disease in the population when interpreting medical test results.

# More Than One Random Variable(Example)

- Flip a coin ten times

صحبت درباره ی قد و وزن  
ادم ها در اینجا دو تا متغیر  
تصادفی داریم

یک پدیده ولی دو تا بعد داره

- $X(\omega)$  = the number of heads that come up as well as
- $Y(\omega)$  = the length of the longest run of consecutive heads

طول طولانی ترین سر های متوالی

# Joint Probability Mass Function

چرا میگیم  
mass function  
چون متغیر هامون گسته هستند

وقتی چندتا متغیر تصادفی داریم یهتابع توزیع تعريف میکنیم که بهش تابع توزیع توام اون متغیرها میگیم ینی دوست داریم راجع به دوناشون اطلاعات کسب کیم

If we have two **discrete** random variables  $X, Y$ , we can define their joint probability mass function (PMF)  $p_{XY} : R^2 \rightarrow [0, 1]$  as:

$$p(x, y) = P(X = x, Y = y)$$

where  $p(x, y) \leq 1$  and  $\sum_{x \in X} \sum_{y \in Y} p(x, y) = 1$

- $X, Y$ : rolling two dice

$$p(x, y) = \frac{1}{36} \quad x, y = 1, 2, \dots, 6$$

- $X$ : rolling one dice     $Y$ : drawing a colored ball

$$p(6, green) = ? \quad p(5, red) = ?$$

تابع توزیع توام متغیر هامون که بهش  
joint probability mass function  
میگیم

اداختن دوتا ناس  
ناس شماره یک میشه متغیر تصادفی اول  
ناس شماره دو میشه متغیر دوم  
وضعیت این دوتا ناس باهمدیگر و احتمال رخدانشون را  
میشه به کمک این تابع بدست اورد

# Joint Probability Density Function

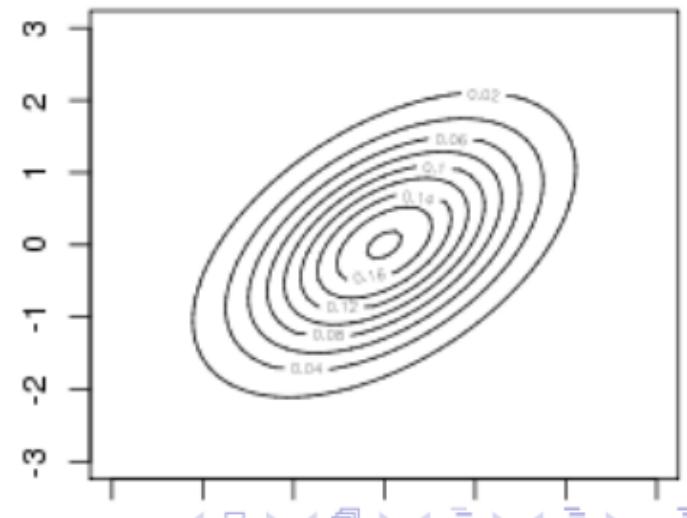
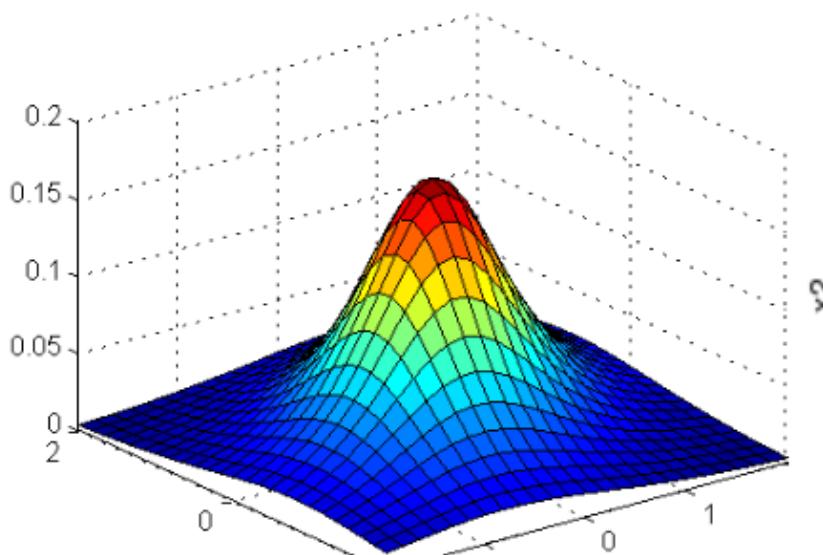
If we have two **continuous** random variables  $X, Y$ , we can define their joint probability density function (PDF)  $f_{XY}$ :  $R^2 \rightarrow [0, 1]$  as:

$$P(a < X < b, c < Y < d) =$$

$$\int_c^d \int_a^b f(x, y) dx dy$$

- 2D Gaussian

فضای پیوسته



# Marginal Probability Mass Function

How does the joint *PMF* over two **discrete** variables relate to the *PMF* for each variable separately? It turns out that

توزیع حاشیه ای پیدا کنیم

$$p(x) = \sum_{y \in Y} p(x, y)$$

پیدا کردن احتمال یک متغیر تصادفی از احتمال توام اون متغیر با بقیه پیدا کنیم باید روی بعدی که نمیخاییم یه جمع انجام بدیم  
مثلًا  $p(x, y)$  را داریم میخاییم راجع به  $(p(x))$  حرف بزنیم

- $X, Y$ : rolling two dice

$$p(x, y) = \frac{1}{36} \quad x, y = 1, 2, \dots, 6$$

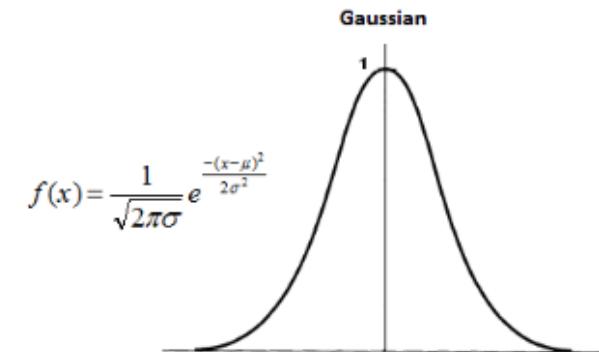
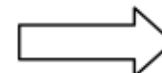
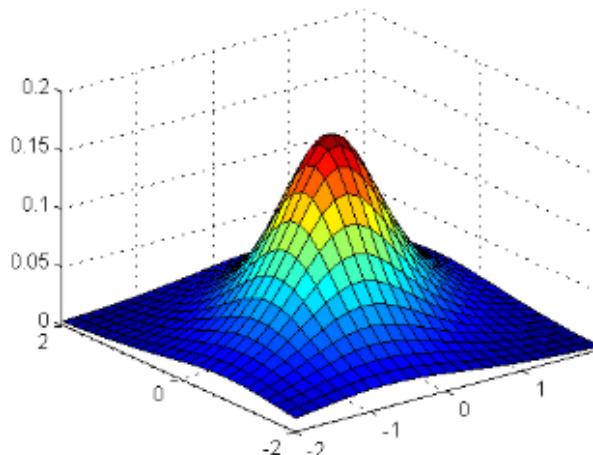
$$p(x) = \sum_{y=1}^6 p(x, y) = \frac{1}{6}$$

# Marginal Probability Density Function

Similarly, we can obtain a marginal *PDF* (also called marginal density) for a **continuous** random variable from a joint *PDF*:

$$f(x) = \int_{-\infty}^{\infty} f(x, y) dy$$

- Integrating out one variable in the 2D Gaussian gives a 1D Gaussian in either dimension



# Conditional Probability Distribution

A conditional probability distribution defines the probability distribution over  $Y$  when we know that  $X$  must take on a certain value  $x$

- Discrete case: conditional PMF

احتمال توان ایکس و وای

$$p(y|x) = \frac{p(x,y)}{p(x)} \iff p(x,y) = p(y|x)p(x)$$

- Continuous case: conditional PDF

$$f(y|x) = \frac{f(x,y)}{f(x)} \iff f(x,y) = f(y|x)f(x)$$

# Marginal vs. Conditional

- Marginal probability:**

احتمال اینکه تاس  
اول ۴ بیاد میشه  
مارجینال

$i \setminus j$	1	2	3	4	5	6	$p_X(i)$
$j$	$1/36$	$1/36$	$1/36$	$1/36$	$1/36$	$1/36$	$1/6$
$p_Y(j)$	$1/6$	$1/6$	$1/6$	$1/6$	$1/6$	$1/6$	

احتمال اینکه هر دو یک بیاد

متغیر های تصادفی مون اینجا  
انداختن دو تا تاس است  
احتمال اینکه هر دو یک بیاد  
یک تقسیم بر ۳۶ است

احتمال اینکه تاس اول  
بیاد میشه  $1/6$

- Conditional probability: probability of rolling a 2**

$i \setminus j$	1	2	3	4	5	6	$p_X(i)$
$j$	$1/36$	$1/36$	$1/36$	$1/36$	$1/36$	$1/36$	$1/6$
$p_Y(j)$	$1/6$	$1/6$	$1/6$	$1/6$	$1/6$	$1/6$	

احتمال اینکه تاس دوم  
بیاد به شرط ۲ بودن تاس  
اول

# Bayes Rule

قوانين بیز  
مشکل اصلی در احتمال ها  
محاسبه‌ی مدل توزیعی است  
 $p(x,y)$

- We can express the joint probability in two ways:

$$p(x, y) = p(y|x)p(x)$$

$$p(x, y) = p(x|y)p(y)$$

ساخت توزیع توأم به  
کمک احتمال شرطی و  
احتمال تکی

- Bayes rule:

حساب کردن توزیع شرطی

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \quad (\text{discrete})$$

جنسش احتمال نیست

$$f(y|x) = \frac{f(x|y)f(y)}{f(x)} \quad (\text{continuous})$$

پیدا کردن توزیع یا احتمال  
یک متغیر

$$P(Y) = \sum_i P(Y|X_i)P(X_i)$$

Proof: law of total probability:

قانون احتمال کل

# Bayes Rule Application

---

A patient underwent a HIV test and got a positive result. Suppose we know that

- Overall risk of having HIV in the population is 0.1%
- The test can accurately identify 98% of HIV infected patients
- The test can accurately identify 99% of healthy patients

What's the probability the person indeed infected HIV?

Bayes' rule is a fundamental theorem in probability theory that describes the probability of an event based on prior knowledge of related events. Here are some examples of Bayes' rule:

**Medical Diagnosis:** Suppose a patient undergoes a medical test for a disease that has a prevalence rate of 5% in the population. The test has a false positive rate of 10% and a false negative rate of 5%. If the patient tests positive, what is the probability that he/she actually has the disease? Using Bayes' rule, we can calculate the probability as follows:

$$P(\text{Disease}|\text{Positive Test}) = P(\text{Positive Test}|\text{Disease}) * P(\text{Disease}) / P(\text{Positive Test})$$

where,

$$P(\text{Positive Test}|\text{Disease}) = 95\% \text{ (True Positive Rate)}$$

$$P(\text{Disease}) = 5\%$$

$$P(\text{Positive Test}) = P(\text{Positive Test}|\text{Disease}) * P(\text{Disease}) + P(\text{Positive Test}|\text{No Disease}) * P(\text{No Disease})$$

$$= 95\% * 5\% + 10\% * 95\%$$

$$= 9.25\%$$

Therefore,

$$P(\text{Disease}|\text{Positive Test}) = 95\% * 5\% / 9.25\% = 51.35\%$$

Hence, the probability that the patient actually has the disease given a positive test result is about 51.35%.

**Email Spam Filtering:** Suppose an email filtering system receives a new email with certain words that occur frequently in spam emails. If 40% of all emails are spam, and the occurrence of these words in spam emails is 70%, while the occurrence of these words in non-spam emails is 20%, what is the probability that this email is spam? Using Bayes' rule, we can calculate the probability as follows:

$$P(\text{Spam}|\text{Words}) = P(\text{Words}|\text{Spam}) * P(\text{Spam}) / P(\text{Words})$$

where,

$$P(\text{Words}|\text{Spam}) = 70\%$$

$$P(\text{Spam}) = 40\%$$

$$P(\text{Words}) = P(\text{Words}|\text{Spam}) * P(\text{Spam}) + P(\text{Words}|\text{No Spam}) * P(\text{No Spam})$$

$$= 70\% * 40\% + 20\% * 60\%$$

$$= 38\%$$

Therefore,

$$P(\text{Spam}|\text{Words}) = 70\% * 40\% / 38\% = 73.68\%$$

Hence, the probability that this email is spam given the occurrence of certain words is about 73.68%.

Sure, here are some numeric examples of Bayes' rule that are slightly more challenging:

**Drug Testing:** A drug test is 99% accurate in detecting a banned substance in an athlete's urine sample. However, the probability of a healthy, non-doping athlete testing positive on the test is 0.1%. If an athlete tests positive, what is the probability that they actually used the banned substance?

$$P(\text{Doping}|\text{Positive Test}) = P(\text{Positive Test}|\text{Doping}) * P(\text{Doping}) / P(\text{Positive Test})$$

where,

$$P(\text{Positive Test}|\text{Doping}) = 99\%$$

$$P(\text{Doping}) = 0.5\%$$

$$\begin{aligned}P(\text{Positive Test}) &= P(\text{Positive Test}|\text{Doping}) * P(\text{Doping}) + P(\text{Positive Test}|\text{No Doping}) * P(\text{No Doping}) \\&= 99\% * 0.5\% + 0.1\% * 99.5\% \\&= 0.598\%\end{aligned}$$

Therefore,

$$P(\text{Doping}|\text{Positive Test}) = 99\% * 0.5\% / 0.598\% = 82.78\%$$

Hence, if an athlete tests positive, there is still an 82.78% chance they actually used the banned substance.

**Disease Testing with Multiple Symptoms:** Suppose a particular disease has a prevalence rate of 1% in a population. The disease causes two symptoms, A and B, with probabilities as follows:  $P(A|\text{Disease}) = 80\%$ ,  $P(B|\text{Disease}) = 70\%$ ,  $P(A|\text{No Disease}) = 10\%$ , and  $P(B|\text{No Disease}) = 20\%$ . If a person has both symptoms A and B, what is the probability that they have the disease?

$$P(\text{Disease}|A,B) = P(A,B|\text{Disease}) * P(\text{Disease}) / P(A,B)$$

where,

$$P(A,B|\text{Disease}) = P(A|\text{Disease}) * P(B|\text{Disease}) = 80\% * 70\% = 56\%$$

$$P(\text{Disease}) = 1\%$$

$$\begin{aligned}P(A,B) &= P(A,B|\text{Disease}) * P(\text{Disease}) + P(A,B|\text{No Disease}) * P(\text{No Disease}) \\&= 56\% * 1\% + 10\% * 20\% * 99\% \\&= 11.84\%\end{aligned}$$

Therefore,

$$P(\text{Disease}|A,B) = 56\% * 1\% / 11.84\% = 4.72\%$$

Hence, if a person has both symptoms A and B, there is only a 4.72% chance that they actually have the disease.

These examples illustrate how Bayes' rule can be used to calculate probabilities when dealing with more complex scenarios like drug testing or diseases with multiple symptoms.



# Bayes Rule - Application

We have two random variables here:

- $X \in \{+, -\}$ : the outcome of the HIV test
- $C \in \{Y, N\}$ : the patient has HIV or not

We want to know:  $P(C=Y|X=+)?$

Apply Bayes rule:

$$p(x=+) = p(x=+ | c=y) * p(c=y) + p(x=+ | c=N) * p(c=N)$$
$$p(x=+) = 0.98 * 0.001 + (1 - 0.99) * (1 - 0.001) =$$

$$P(C=Y|X=+) = \frac{P(X=+|C=Y)P(C=Y)}{P(X=+)}$$

$$P(X=+|C=Y) = 0.98$$

$$P(C=Y) = 0.001$$

$$P(X=+) = 0.98 * 0.001 + (1 - 0.99) * 0.999 = 0.01097$$

$$\text{Answer: } 0.98 * 0.001 / 0.01097 = 8.9\%$$

$$p(x=+ | C=N) = 1 - p(x=- | C=N)$$

$$p(C=N) = 1 - 0.001 = 0.999$$

# Independence

---

Two random variables  $X$  and  $Y$  are independent iff

- For **discrete** random variables

$$p(x, y) = p(x)p(y) \quad \forall x \in X, y \in Y$$

- For **discrete** random variables

$$p(y|x) = p(y) \quad \forall y \in Y \text{ and } p(x) \neq 0$$

- For **continuous** random variables

$$f(x, y) = f(x)f(y) \quad \forall x, y \in R$$

- For **continuous** random variables

$$f(y|x) = f(y) \quad \forall y \in R \text{ and } f(x) \neq 0$$

# Multiple Random Variables

پیدا کردن توزیع توانم تا متغیر تصادفی

Extend to multiple random variables :

- Joint Distribution (discrete):

$$p(x_1, \dots, x_n) = P(X_1 = x_1, \dots, X_n = x_n)$$

- Conditional Distribution (chain rule - discrete)

$$p(x_1, \dots, x_n) = p(x_n | x_1, \dots, x_{n-1}) p(x_1, \dots, x_{n-1})$$

$$= p(x_n | x_1, \dots, x_{n-1}) p(x_{n-1} | x_1, \dots, x_{n-2}) p(x_1, \dots, x_{n-2})$$

$$= p(x_1) \prod_{i=2}^n p(x_i | x_1, \dots, x_{i-1})$$

نوشتن توزیع توانم به کمک  
مجموعه های توزیع شرطی

(continuous case can be defined similarly using PDF)

$$p(x_1) * p(x_2 | x_1) * p(x_3 | x_1, x_2) * p(x_4 | x_1, x_2, x_3) * \dots * p(x_n | x_1, x_2, \dots, x_{n-1})$$

# Multiple Random Variables

---

- Independence:

**Discrete** case:  $X_1, \dots, X_n$  are independent iff

$$p(x_1, \dots, x_n) = \prod_{i=1}^n p(x_i)$$

**Continuous** case:  $X_1, \dots, X_n$  are independent iff

$$f(x_1, \dots, x_n) = \prod_{i=1}^n f(x_i)$$



# **SOME OTHER POINTS**

# Probabilistic View of a Dataset

---

What about a dataset  $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ ?

- We can view  $S$  as  $d + 1$  random variables where  $d$  is the number of attributes in  $\mathbf{x}$ , i.e.

$$X_1, X_2, \dots, X_d, Y$$

- Uncover(model)  $p(x_1, x_2, \dots, x_d, y)$  from the training data

- For ANY  $(x_1, x_2, \dots, x_n)$ , we will compute:

$$P(y = 0 | x_1, x_2, \dots, x_n) ?$$

$$P(y = 1 | x_1, x_2, \dots, x_n) ?$$

That is predicting  $y$  from  $\mathbf{x}$  !

# Bayes Rule Terminology

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

The diagram illustrates the components of Bayes' rule. A red bracket encloses the entire equation  $P(Y|X)$ . Inside this bracket, four yellow boxes with labels are connected by red arrows: 'posterior' points to  $P(Y|X)$ , 'likelihood' points to  $P(X|Y)$ , 'prior' points to  $P(Y)$ , and 'marginal probability' points to  $P(X)$ .

$P(Y)$ : prior probability or, simply, **prior**

$P(X|Y)$ : conditional probability or, **likelihood**

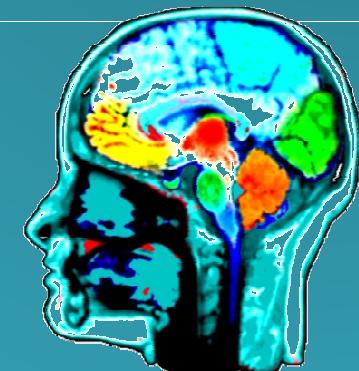
$P(X)$ : marginal probability

$P(Y|X)$ : posterior probability or, simply, **posterior**



# Introduction To Data Mining

Isfahan University of Technology (IUT)  
Bahman 1401



Getting to Know Your Data

Dr. Hamidreza Hakim  
[hamid.hakim.u@gmail.com](mailto:hamid.hakim.u@gmail.com)

# Content

---

---

Attributes and Objects

Types of Data

Basic Statistical Descriptions of Data

Data Visualization

Similarity and Dissimilarity Measures

---

---

# **ATTRIBUTES AND OBJECTS**

# What is Data?

- Collection of *data objects* and their *attributes*

The diagram shows a table representing a dataset. The columns are labeled *Name*, *Team*, *Number*, *Position*, and *Age*. The rows are indexed from 0 to 6. A red box highlights the row for Jonas Jerebko. A blue box highlights the column for Position. An orange arrow labeled "Rows" points to the vertical axis of the table. A blue arrow labeled "Columns" points to the horizontal axis. A pink box labeled "Data" encloses the entire table area.

	Name	Team	Number	Position	Age
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

ما يک شی را با ویژگی هاش میشناسیم  
ویژگی ها را به عنوان ستون ها و اtribut ها توصیف میکنیم

# What is Data?

یه سری مشخصه هستند  
که ابجکت را برامون  
توصیف میکنند

- An **attribute** is a **property** or **characteristic** of an object
  - Examples: **eye color** of a person, **temperature**, etc.
  - Attribute is also known as **variable**, **field**, **characteristic**, **dimension**, or **feature**
- A **collection of attributes** describe an **object**
  - **Object** is also known as **record**, **point**, **case**, **sample**, **entity**, or **instance**

Objects

Attributes

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

# Attribute Values

---

---

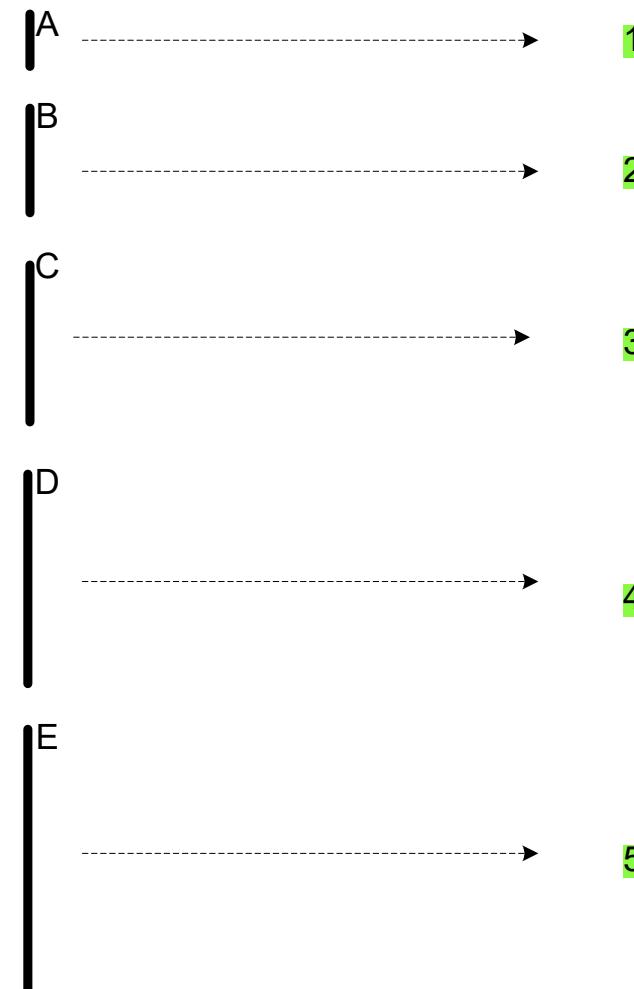
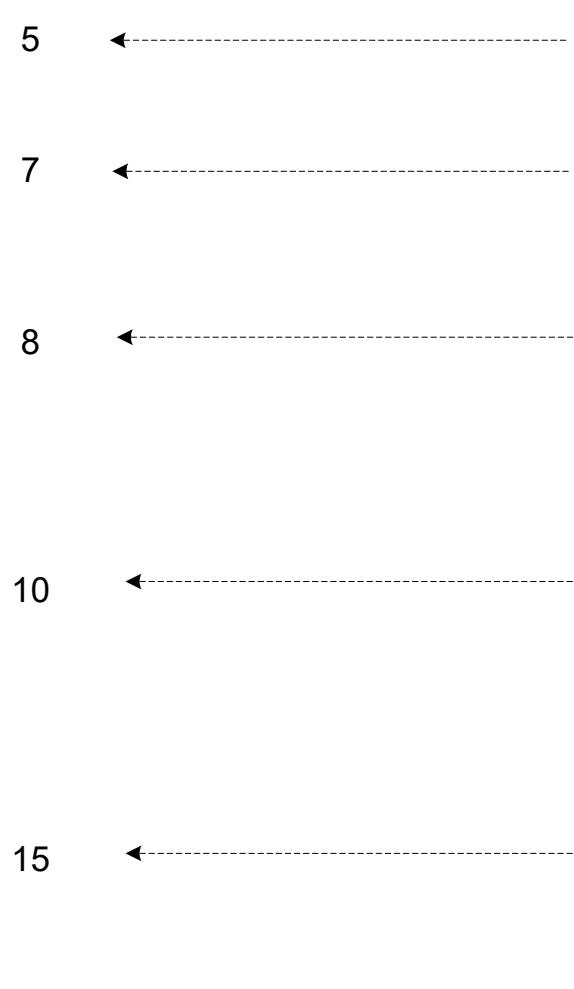
- **Attribute values** are numbers or symbols assigned to an attribute for a particular object
- Distinction between attributes and attribute values
  - Same attribute can be mapped to different attribute values
    - ◆ Example: height can be measured in feet or meters
  - Different attributes can be mapped to the same set of values
    - ◆ Example: Attribute values for ID and age are integers
  - But properties of attribute can be different than the properties of the values used to represent the attribute

# Measurement of Length

ایا فاصله ها و مقیاس هارا داریم حفظ میکنیم؟ ایا اصلا  
خوبه که مقیاس ها حفظ شوند؟  
چطوری با اندازه ها باید برخورد کنیم

- The way you measure an attribute may not match the attributes properties.

This scale preserves only the ordering property of length.



This scale preserves the ordering and additivity properties of length.

# Types of Attributes

- There are different types of attributes

- **Nominal:**

اسمی  
دسته ای

اتribیوت هایی که دسته دسته و  
حالت حالت هستند مثل برچسب ها

- ◆ categories, states, or “names of things”
- ◆ Hair\_color = {auburn, black, blond, brown, grey, red, white}
- ◆ marital status, occupation, ID numbers, zip codes
- ◆ Examples: ID numbers, eye color, zip codes

- **Ordinal:**

یک توالی و سیکونسی بین مقادیر وجود  
داره  
اندازه کتاب : کوچک، متوسط، بزرگ

مقادیر دارای نظم (رتبه بندی) معنیداری هستند اما  
مقدار بین مقادیر متولی مشخص نیست

- ◆ Values have a meaningful order (ranking) but magnitude between successive values is not known
- ◆ Size = {small, medium, large}, grades, army rankings
- ◆ Examples: rankings (e.g., taste of potato chips on a scale from 1-10), grades, height {tall, medium, short}

# Types of Attributes(Example)

- There are different types of attributes

- Nominal

- Ordinal

- Interval متغیرهای فاصله‌ای

- ◆ Measured on a scale of equal-sized units q Values have order

- ◆ Examples: calendar dates, temperatures in Celsius or Fahrenheit.

- Ratio

- ◆ Inherent zero-point

اونایی که یه پایه و بیس  
صفر دارند حتما ratio  
هستند یعنی منفی ندارند  
مثل تعداد افراد در اتاق

منفی هم میتوانند بشوند.

- ◆ Examples: temperature in Kelvin, length, counts, elapsed time (e.g., time to run a race)

منفی برای طول معنا  
نداره پس میشه ریشيو

مثلاً دما که صفر هم توش هست اگه بگیم دما منفی دو درجه است، دو برابر ش چی میشه؟ دو برابر  
بیشتر را چطوری حساب کنیم؟ اگه بگیم دو برابر گرمتر یا دو برابر سردتر مشخص میشه به سمت  
چی باید ببریم عدد را

# Question

---

---

- Q1: Is student ID a nominal, ordinal, or interval-scaled data?
- Q2: What about eye color? Or color in the color spectrum of physics?

# Question

---

---

- Q1: Is student ID a nominal, ordinal, or interval-scaled data?

Nominal

- Q2: What about eye color? Or color in the color spectrum of physics? q

Eye color: Nominal (similar to hair color)

Color spectrum of physics: Interval (RGB space supports +/-)

# Properties of Attribute Values

---

---

- The type of an attribute depends on which of the following properties/operations it possesses:

- Distinctness:

 $= \neq$ 

- Order:

 $< >$ 

- Differences are meaningful

 $+ -$ 

- Ratios are meaningful

 $* /$ 

نسبت

- Nominal attribute: distinctness

ترتيبی

- Ordinal attribute: distinctness & order

- Interval attribute: distinctness, order & meaningful differences

- Ratio attribute: all 4 properties/operations

# Difference Between Ratio and Interval

---

---

- Is it physically meaningful to say that a temperature of  $10^{\circ}$  is **twice** that of  $5^{\circ}$  on
  - the Celsius scale?
  - the Fahrenheit scale?
  - the Kelvin scale?

چون کلوین یه بیس صفر داره پس ریشیو  
است یعنی مقدار منفی نداره پس عملیات  
مای ضرب و تقسیم را ساپورت میکنه پس  
 فقط برای کلوین میشه بگیم  $10^{\circ}$  درجه ی  
 کلوین دو برابر درجه ی کلوین است.  
 برای سلسیوس و فارنهایت نمیشه گفت  
 چون ممکنه منفی باشه درجه اش و  
 نمیتوانیم بگیم  $2^{\circ}$  برایر سردتره یا گرمتر؟

- Consider **measuring the height above average**
  - If Bill's height is three inches above average and Bob's height is six inches above average, then would we say that Bob is twice as tall as Bill?
  - Is this situation analogous to that of temperature?

	<b>Attribute Type</b>	<b>Description</b>	<b>Examples</b>	<b>Operations</b>
Categorical Qualitative	Nominal	Nominal attribute values only distinguish. ( $=, \neq$ )	zip codes, employee ID numbers, eye color, sex: { <i>male</i> , <i>female</i> }	mode, entropy, contingency correlation, $\chi^2$ test
	Ordinal	Ordinal attribute values also order objects. ( $<, >$ )	hardness of minerals, { <i>good</i> , <i>better</i> , <i>best</i> }, grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
Numeric Quantitative	Interval	For interval attributes, differences between values are meaningful. (+, -)	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation, Pearson's correlation, <i>t</i> and <i>F</i> tests
	Ratio	For ratio variables, both differences and ratios are meaningful. (*, /)	temperature in Kelvin, monetary quantities, counts, age, mass, length, current	geometric mean, harmonic mean, percent variation

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

Attribute Type	Transformation	Comments
Nominal	Any permutation of values	If all employee ID numbers were reassigned, would it make any difference?
Ordinal	<p>An order preserving change of values, i.e., <math>new\_value = f(old\_value)</math> where <math>f</math> is a monotonic function</p>	<p>An attribute encompassing the notion of good, better best can be represented equally well by the values {1, 2, 3} or by {0.5, 1, 10}.</p>
Interval	$new\_value = a * old\_value + b$ where $a$ and $b$ are constants	<p>Thus, the Fahrenheit and Celsius temperature scales differ in terms of where their zero value is and the size of a unit (degree).</p>
Ratio	$new\_value = a * old\_value$	<p>Length can be measured in meters or feet.</p>

اگه همه ی شماره دانشجویی ها عوض بشه هیچ تفاوتی ایجاد میشه چون فرقی نداره شماره دانشجویی من باشه یا ۲

تغییر و Transform بدل کردن اطلاعات های ارتباطی با بد ترتیب را حفظ کنه یعنی کوچک و بزرگی مهم است ولی ارتباط بین ولیوها مهم نیست.

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

# Discrete and Continuous Attributes

## ● Discrete Attribute

- Has only a **finite** or **countably infinite set** of values
- Examples: **zip codes**, **counts**, or the set of **words** in a collection of documents
- Often represented as **integer variables**.
- Note: **binary attributes** are a special case of discrete attributes

## ● Continuous Attribute

- Has **real numbers** as attribute **values**
- Examples: **temperature**, **height**, or **weight**.
- Practically, real values can only be measured and represented using a finite number of digits.
- **Continuous attributes** are typically represented as **floating-point variables**.

# Asymmetric Attributes

---

---

- Only presence (a non-zero attribute value) is regarded as important
  - ◆ Words present in documents
  - ◆ Items present in customer transactions
- If we met a friend in the grocery store would we ever say the following?

*“I see our purchases are very similar since we didn’t buy most of the same things.”*

# Critiques of the attribute categorization

---

- Incomplete
  - Asymmetric binary
  - Cyclical
  - Multivariate
  - Partially ordered
  - Partial membership
  - Relationships between the data
- Real data is approximate and noisy
  - This can complicate recognition of the proper attribute type
  - Treating one attribute type as another may be approximately correct

---

---

# **TYPES OF DATA**

# Types of data sets

- Record(Tabular)
  - Data Matrix
  - Document Data
  - Transaction Data

csv files,  
database tables  
each row is  
independant of  
another row  
هر رکورد مثلا برای یک  
نفره

- Graph
  - World Wide Web
  - Molecular Structures

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

داده های دارای ترتیب  
- داده های فضایی  
- داده های زمانی  
- داده های متوالی  
- داده های توالی ژنتیکی

# Record Data

---

---

- Data that consists of a collection of records, each of which consists of a fixed set of attributes

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

# Data Matrix

همهی اtribut ها  
عددی هستند.

- If data objects have the same fixed set of numeric attributes, then the data objects can be thought of as points in a multi-dimensional space, where each dimension represents a distinct attribute
- Such a data set can be represented by an  $m$  by  $n$  matrix, where there are  $m$  rows, one for each object, and  $n$  columns, one for each attribute

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

# Document Data

کاربا داده های متنی  
مثالا میخایم بفهمیم  
موضوع فایل پی دی افی  
که دستمون هست چیه؟  
مثالا اگه کلمات مرتبط با  
ورزش زیاد تکرار شده  
میگیم احتمالا پی دی افه  
درباره ورزشه

- Each document becomes a ‘term’ vector
  - Each term is a component (attribute) of the vector
  - The value of each component is the number of times the corresponding term occurs in the document.

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

این ترم ها میشن اتربیوت های داده داکیومنت ما

هر کلمه ای چندبار توی متن تکرار شده؟

# Transaction Data

- A special type of data, where
  - Each transaction involves a set of items.
  - For example, consider a grocery store. The set of products purchased by a customer during one shopping trip constitute a transaction, while the individual products that were purchased are the items.
  - Can represent transaction data as record data

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

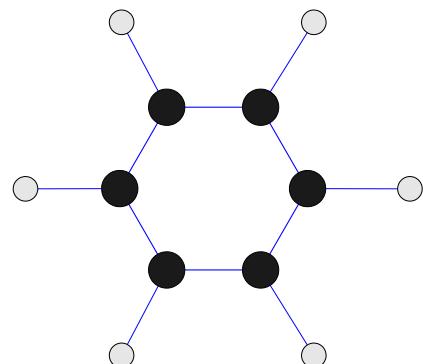
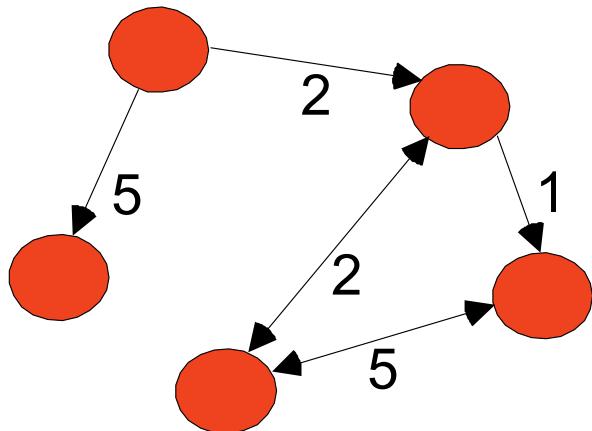
ایدی تراکنش، هر کورد  
یک تراکنش است.

هر کدام از این محصول  
هایی که خریداری شدند  
یک ایتم هستند.

# Graph Data

ارتباط بین چندنفر یا  
دوستی چندنفر  
هر المان مرتبط با یک  
موجودیت  
شبکه های اجتماعی  
صفحات اینترنتی

- Examples: Generic graph, a molecule, and webpages



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

## Useful Links:

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

## Knowledge Discovery and Data Mining Bibliography

(Gets updated frequently, so visit often!)

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

## Book References in Data Mining and Knowledge Discovery

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

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

## General Data Mining

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

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

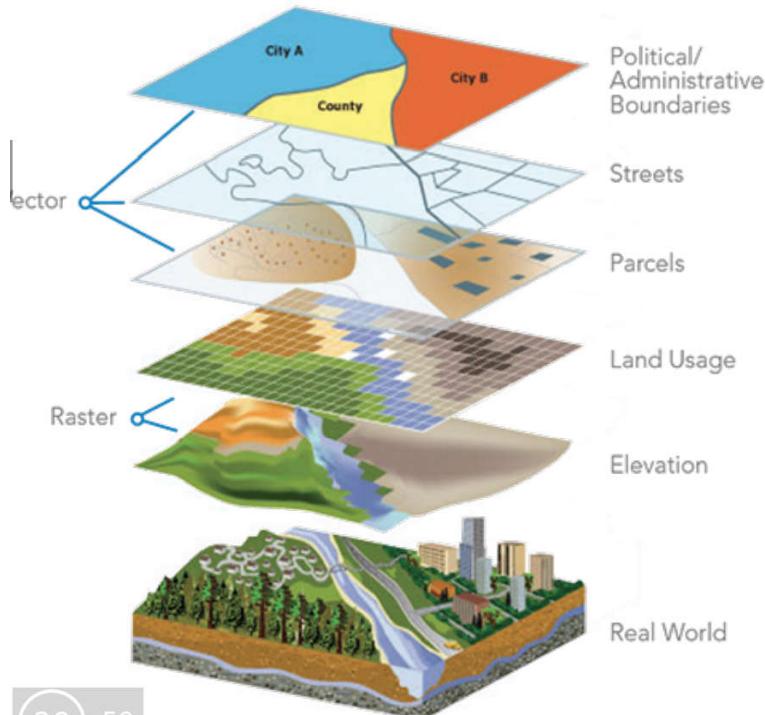
# Spatial/Image Data

دیتاهای فضایی  
تشخیص چهره یا  
شناسایی مکان  
ر تصاویر ماهواره ای  
با چندین لایه از تصویر  
کار داریم

## ● Spatio-Temporal Data

داده های مکانی-زمانی

### Maps



### Images



# Ordered Data

دیتاهايی که ترتیب دارند  
برامون مهمه يه فردی که  
وارد فروشگاه میشه اول  
چی رو میخره بعد چی رو  
تولی کالاهای خریداری  
شده مهمه

- Sequences of transactions

Items/Events



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



An element of  
the sequence

# Ordered Data

---

---

- Genomic sequence data

توالی زن ها که براساس  
ترتیب یه اطلاعاتی به ما  
میده

GGTTCCGCCCTTCAGCCCCGGCGC  
CGCAGGGCCCGCCCCGCGGCCGTC  
GAGAAGGGCCCGCCTGGCGGGCG  
GGGGGAGGCAGGGGCCGCCCAGC  
CCAACCGAGTCCGACCAAGGTGCC  
CCCTCTGCTCGGCCTAGACCTGA  
GCTCATTAGGCGGCAGCGGACAG  
GCCAAGTAGAACACGCGAAGCGC  
TGGGCTGCCTGCTGCGACCAGGG

# Ordered Data

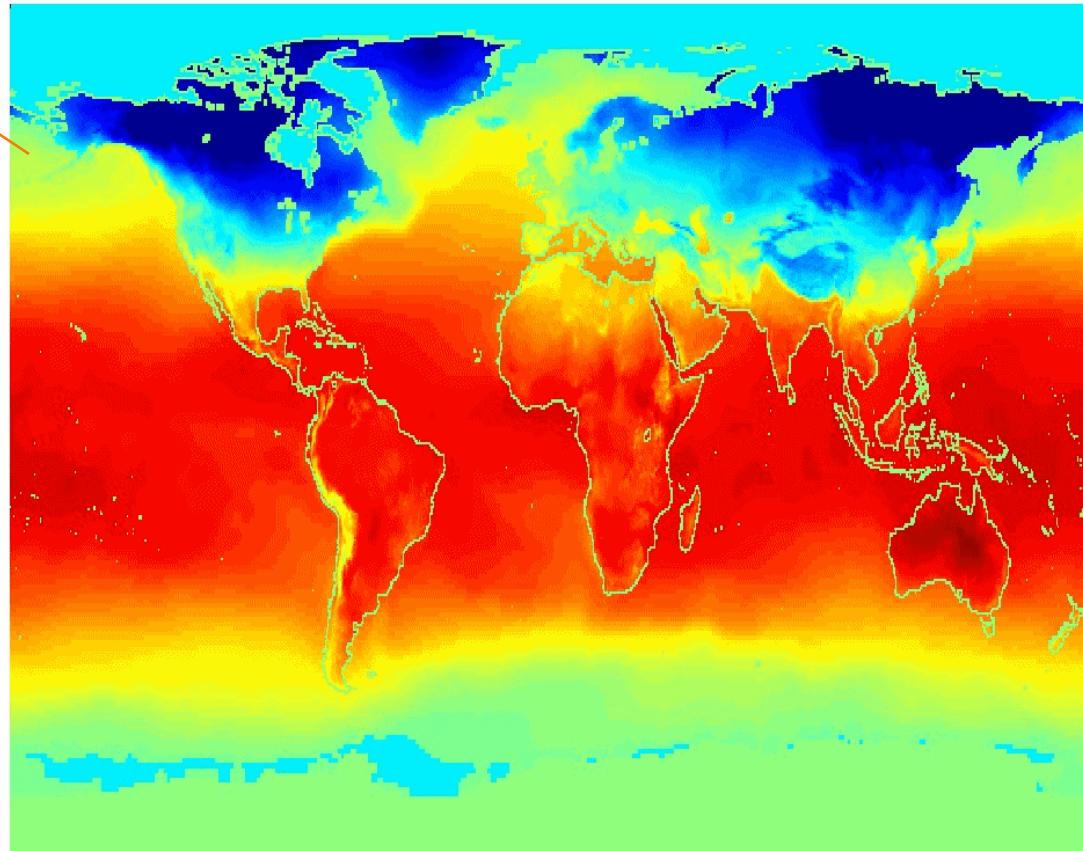
- Spatio-Temporal Data

تغییرات دما  
علاوه بر تغییرات زمانی  
که دارند در فضا پراکنده  
هم هستند  
از اطلاعات همسایه ها  
میتوانیم استفاده کنیم

Jan

ارتباط مکان ها یا  
همسایگیشون اهمیت داره

Average Monthly  
Temperature of  
**land** and ocean



---

---

# **BASIC STATISTICAL DESCRIPTIONS OF DATA**

# Basic Statistical Descriptions of Data

تمایل داده ها به چه سمتی است؟ پراکندگی داده ها  
چطوریه؟ چطوری توزیع شدن؟  
برای اندازه گیری اینها یه سری مقیاس داریم  
چارک و واریانس میانه میانگین ...

- Motivation

- To better understand the data: **central tendency**, **variation** and **spread**

- Data dispersion characteristics

- **median**, **max**, **min**, **quantiles**, **outliers**, **variance**, etc.

مقدیری که خیلی پراکنده  
هستند و از اکثریت  
فاصله دارند

- Numerical dimensions correspond to **sorted intervals**

- **Data dispersion**: analyzed with multiple granularities of precision
- **Boxplot** or **quantile** analysis on **sorted intervals**

- Dispersion analysis on computed measures

- **Folding** measures into numerical dimensions
- **Boxplot** or **quantile** analysis on the transformed cube

# Measuring the Central Tendency

- Mean (algebraic measure) (sample vs. population):

میانگین

میانگین سampل یا نمونه

سایز سampل یا نمونه

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

میانگین جمعیت یا  
پاپولیشن

$$\mu = \frac{\sum x}{N}$$

سایز جمعیت

Note:  $n$  is sample size and  $N$  is population size.

- Weighted arithmetic mean:

میانگین وزن دار

مثل محاسبه ی معدل کل  
که ضریب هر درس  
درش ضرب میشه.

$$\bar{x} = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i}$$

جمع وزن دار ایتم ها  
یعنی وزن هر متغیری را  
هم در نظر میگیریم.

جمع وزن ها یا جمع  
ضرایب هر متغیر

- Trimmed mean: chopping extreme values?(2%)

۲ درصد مقادیر خیلی بالا و خیلی پایین  
را در میانگین در نظر نگیریم

# Measuring the Central Tendency

- Median: میانه

- Middle value if odd number of values, or average of the middle two values otherwise
- Estimated by interpolation (for grouped data):

درون یابی  
میانه توی این رنج است.

age	frequency
1-5	200
6-15	450
16-20	300
21-50	1500
51-80	700
81-110	44

Approximate median  
تخمينی از میانه

$$\text{median} = L_1 + \left( \frac{n/2 - (\sum freq)_l}{freq_{median}} \right) width$$

Low interval limit      Sum before the median interval      Interval width ( $L_2 - L_1$ )

Using Equation (2.3), we have  $L_1 = 20$ ,  $N = 3194$ ,  $(\sum freq)_l = 950$ ,  $freq_{median} = 1500$ ,  $width = 30$ ,  $median = 32.94$  years.

$$200 + 450 + 300 + 1500 + 700 + 44 = 3149$$

$$200 + 450 + 300 = 950$$

$$50 - 20 = 30$$

$$20 + ((3194/2) - 950)/1500 * 30 = 32.94$$

# Measuring the Central Tendency

---

---

- **Mode** بیشترین تکرار
  - Value that occurs **most** frequently in the data
  - Unimodal, bimodal, trimodal
  - Empirical formula:

فرمول تجربی

$$mean - mode = 3 \times (mean - median)$$

# Symmetric vs. Skewed Data

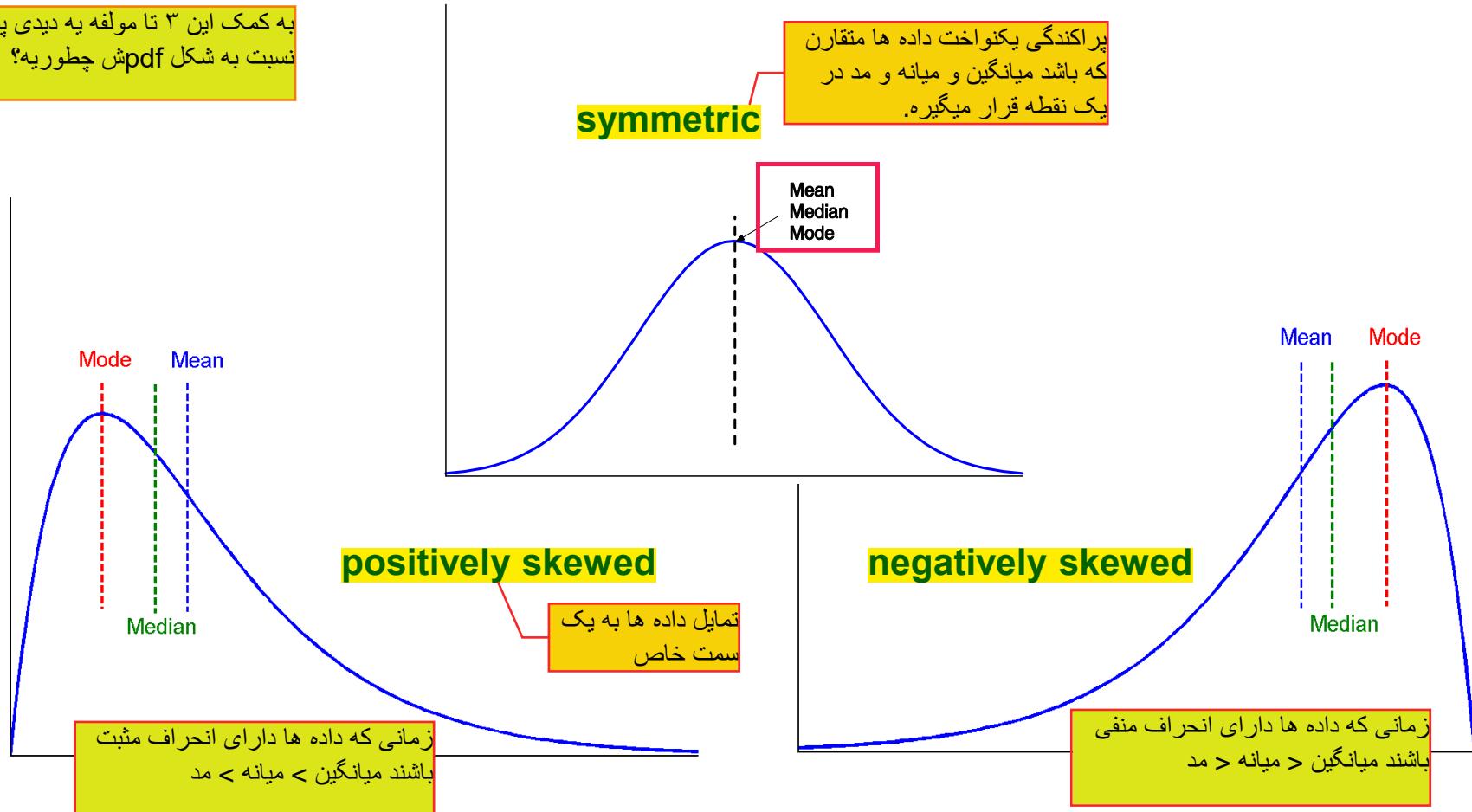
- Median, mean and mode of symmetric, positively and negatively skewed data

به کمک این ۳ تا مولفه په دیدی پیدا کنیم  
نسبت به شکل pdf ش چطوریه؟

پراکندگی یکنواخت داده ها متقارن  
که باشد میانگین و میانه و مد در  
یک نقطه قرار میگیره.

**symmetric**

Mean  
Median  
Mode



# Measuring the Dispersion of Data

پراکندگی

- Variance and standard deviation (sample:  $s$ , population:  $\sigma$ )
  - Variance:** (algebraic, scalable computation)

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^n (x_i - \mu)^2 = \frac{1}{N} \sum_{i=1}^n x_i^2 - \mu^2$$

واریانس جمعیت
میانگین جمعیت (کل داده ها)

شاخص های پراکندگی  
واریانس و انحراف معیار  
ز خود واریانس و قوی استفاده میکنیم که  
میانگین کل جمعیت رو داشته باشیم

Standard deviation  $s$  (or  $\sigma$ ) is the square root of variance  $s^2$  (or  $\sigma^2$ )

sample variance  
یه نمونه داریم میخایم  
راجع به کل جمعیت  
تخمین بزنیم

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 = \frac{1}{n-1} \left[ \sum_{i=1}^n x_i^2 - \frac{1}{n} \left( \sum_{i=1}^n x_i \right)^2 \right]$$

واریانس سempl یا نمونه  
که در مخرجش به جای  
 $n$  باید  $n-1$  بگذاریم.
میانگین تخمینی از داده ها

# Measuring the Dispersion of Data

- Quartiles, outliers and boxplots

چارک اول و دوم و...

**Quartiles:**  $Q_1$  ( $25^{\text{th}}$  percentile),  $Q_3$  ( $75^{\text{th}}$  percentile)

- **Inter-quartile range:**  $IQR = Q_3 - Q_1$

هرچی فاصله این دو تا  
یاد باشه بنی داده هامون  
پراکنده تره

رویکرد ۵ شماره ای

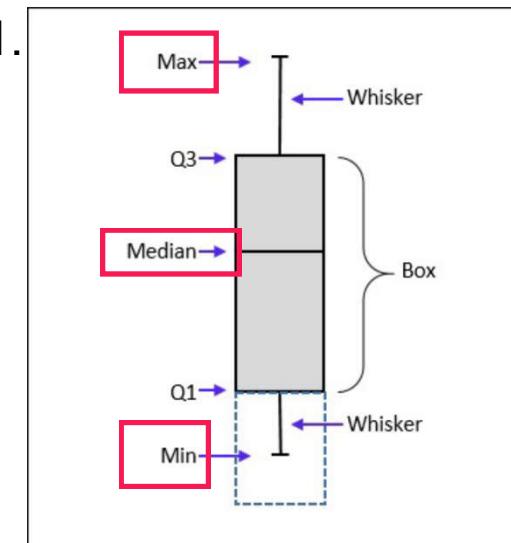
**Five number summary:** min,  $Q_1$ , median,  $Q_3$ , max

- **Boxplot:** ends of the box are the quartiles; median is marked; add whiskers, and plot outliers individually

- **Whiskers:** two lines outside the box extended to Minimum and Maximum

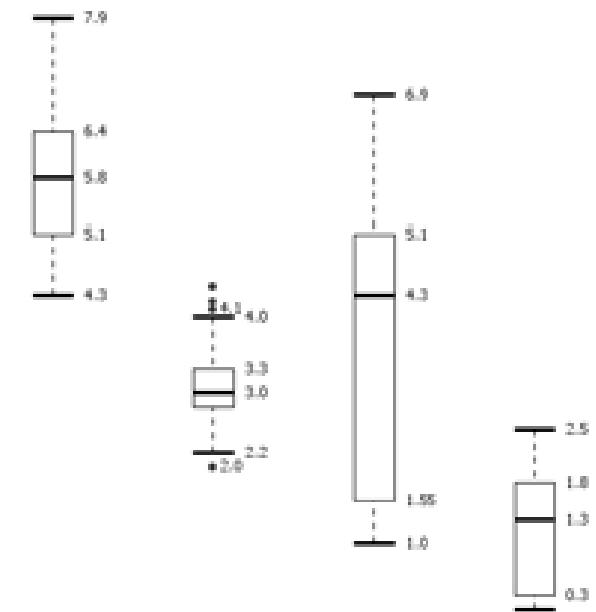
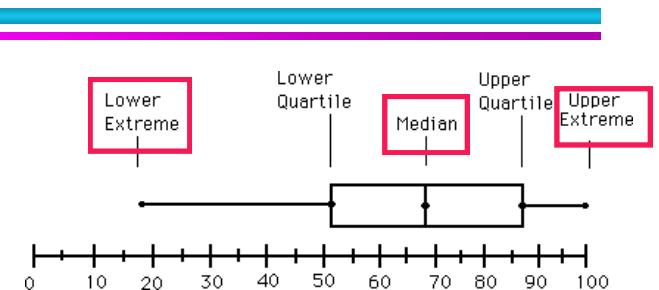
- **Outlier:** usually, a value higher/lower than 1.

مقادیری که یک و نیم  
برابر فاصله ای  
هستند



# Boxplot Analysis

- **Five-number summary** of a distribution
  - Minimum, Q1, Median, Q3, Maximum
- **Boxplot**
  - Data is represented with a **box**
  - The ends of the box are at the first and third quartiles, i.e., **the height of the box is IQR**
  - The **median** is marked by a **line** within the box
  - **Whiskers**: two lines outside the box extended to Minimum and Maximum
  - **Outliers**: points beyond a specified outlier threshold, **plotted individually**



Suppose that the data for analysis includes the attribute *age*. The *age* values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

- (a) What is the *mean* of the data? What is the *median*?
- (b) What is the *mode* of the data? Comment on the data's *modality* (i.e., bimodal, trimodal, etc.).
- (c) What is the *midrange* of the data?
- (d) Can you find (roughly) the first quartile (*Q1*) and the third quartile (*Q3*) of the data?
- (e) Give the *five-number summary* of the data.
- (f) Show a *boxplot* of the data.
- (g) How is a *quantile-quantile plot* different from a *quantile plot*?

**Answer:**

- (a) What is the *mean* of the data? What is the *median*?

The (arithmetic) *mean* of the data is:  $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i = 809/27 = 30$  (Equation 2.1). The *median* (middle value of the ordered set, as the number of values in the set is odd) of the data is: 25.

- (b) What is the *mode* of the data? Comment on the data's modality (i.e., bimodal, trimodal, etc.).
- This data set has *two values* that occur with the *same highest frequency* and is, therefore, *bimodal*. The modes (values occurring with the greatest frequency) of the data are 25 and 35.

- (c) What is the *midrange* of the data?

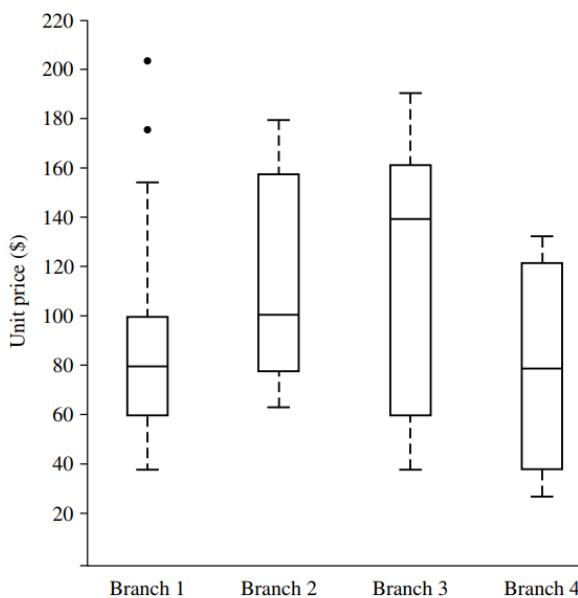
The midrange (average of the largest and smallest values in the data set) of the data is:  $(70 + 13)/2 = 41.5$

- (d) Can you find (roughly) the first quartile (*Q1*) and the third quartile (*Q3*) of the data?

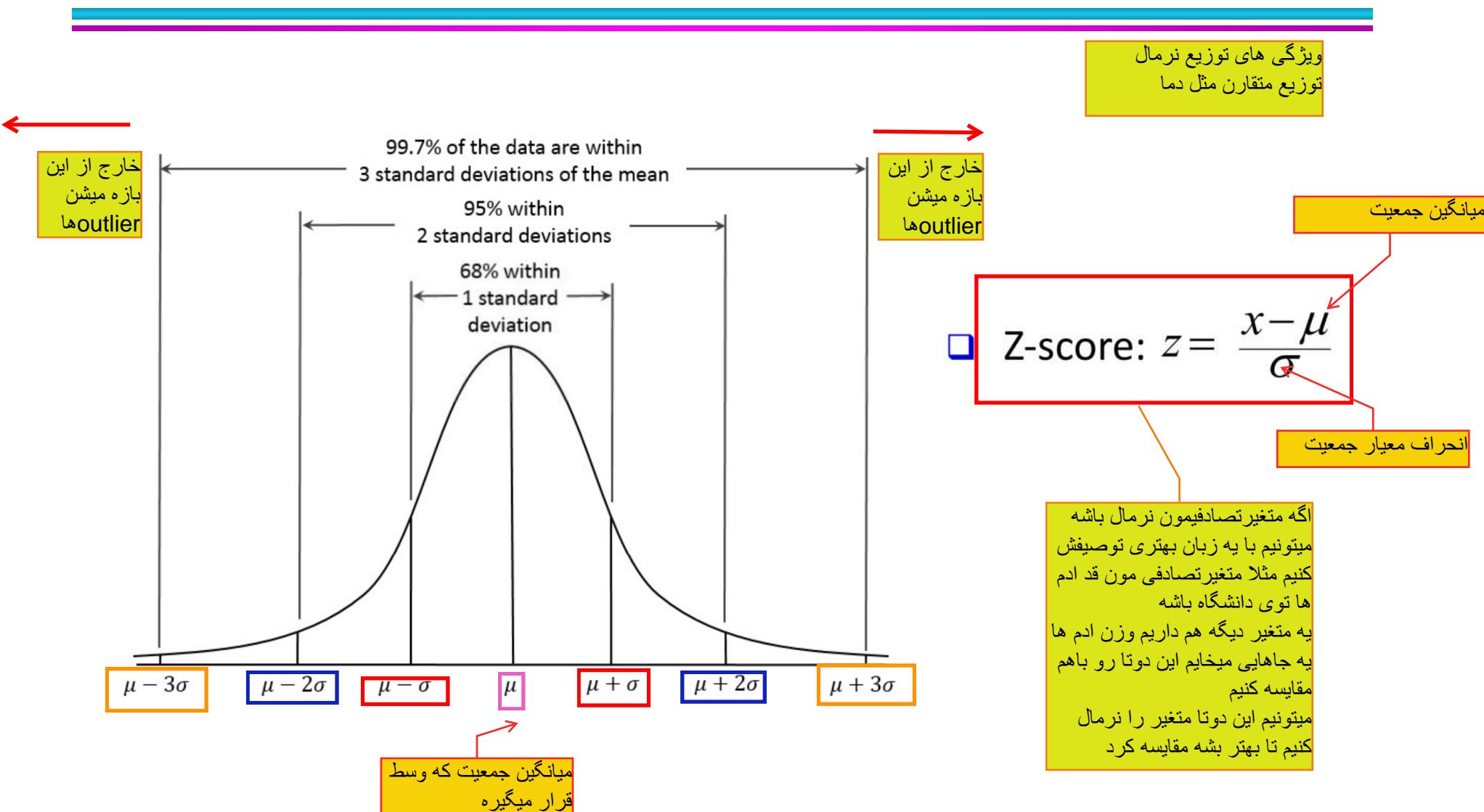
The first quartile (corresponding to the 25th percentile) of the data is: 20. The third quartile (corresponding to the 75th percentile) of the data is: 35.

- (e) Give the *five-number summary* of the data.

The five number summary of a distribution consists of the *minimum* value, *first quartile*, *median* value, *third quartile*, and *maximum* value. It provides a good summary of the shape of the distribution and for this data is: 13, 20, 25, 35, 70.



# Properties of Normal Distribution Curve



z-score normalization

The range is  $[(old\_min - mean) / stdDev, (old\_max - mean) / stdDev]$ . In general the range for all possible data sets is  $(-\infty, +\infty)$ .

### Example 1:

Suppose a student scores 80 out of 100 on a test, and the mean score for the class is 75 with a standard deviation of 5. To calculate the z-score for this student's score:

$$z = (80 - 75) / 5$$

$$z = 1$$

Therefore, the student's score is one standard deviation above the mean.

### Example 2:

A company has a sales team with an average monthly sales of \$10,000 and a standard deviation of \$2,000. If one salesperson had sales of \$14,000 in a month, what is their z-score?

$$z = (14,000 - 10,000) / 2,000$$

$$z = 2$$

This means that the salesperson's monthly sales were two standard deviations above the mean.

### Example 3:

A researcher wants to know if there is a significant difference in height between men and women. She measures the heights of a random sample of men and women and finds that the mean height for men is 70 inches with a standard deviation of 3 inches, while the mean height for women is 65 inches with a standard deviation of 2 inches. If a man is randomly selected from this sample and his height is recorded as 73 inches:

$$z = (73 - 70) / 3$$

$$z = 1$$

This means that this man's height is one standard deviation above the mean height for men.

### Example 4:

A teacher wants to compare her students' scores on two different tests. The first test has an average score of 80 with a standard deviation of 10, while the second test has an average score of 75 with a standard deviation of 8. If one student scored an 85 on both tests:

For Test #1:

$$z = (85 - 80) / 10$$

$$z = 0.5$$

For Test #2:

$$z = (85 - 75) / 8$$

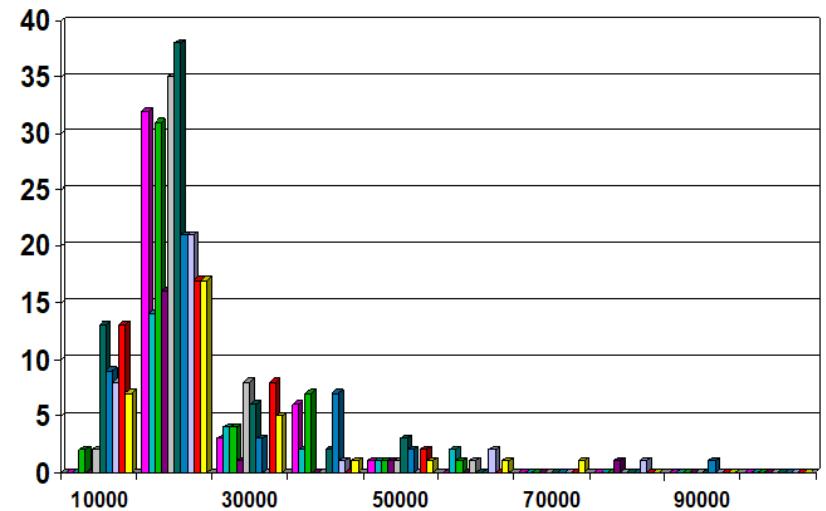
$$z = 1.25$$

This means that the student's score on Test #1 was half a standard deviation above the mean, while their score on Test #2 was 1.25 standard deviations above the mean.

# Histogram Analysis

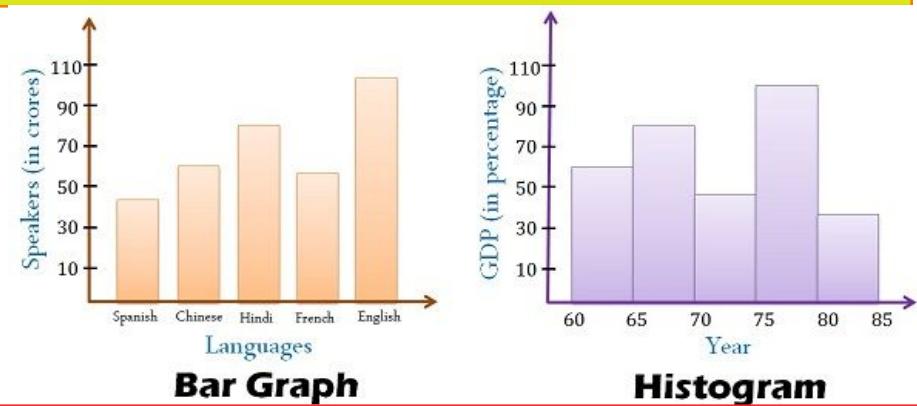
یه نموداری که تعداد تکرار مقادیر مختلف یه متغیر رو نشون میده  
متلا متفیرمون از ۱۰ هزاره تا ۹۰ هزار  
میایم ۱۰۰ تا ۱۰۰ تا جدایشون میکنیم و هر بار  
توى این بازه ها یه متغیری رخداد یکی اضافه  
میشه به تعدادشون  
درباره ی توزیع داده ها یه توضیحی میده

- Histogram: Graph display of tabulated frequencies, shown as bars
- It shows what proportion of cases fall into each of several categories
- Differs from a bar chart in that it is the area of the bar that denotes the value, not the height as in bar charts, a crucial distinction when the categories are not of uniform width
- The categories are usually specified as non-overlapping intervals of some variable. The categories (bars) must be adjacent



1. Data type: A bar chart is used to represent categorical data, while a histogram is used to represent continuous data.
2. X-axis: In a bar chart, the x-axis represents the categories being compared, while in a histogram, the x-axis represents the range of values being measured.
3. Y-axis: In both charts, the y-axis represents the frequency or count of each category or value.
4. Bar width: In a bar chart, there is typically space between each bar, while in a histogram, the bars are usually touching or overlapping to show continuity of data.
5. Interpretation: A bar chart is useful for comparing discrete categories or groups, while a histogram is useful for showing patterns and distributions within continuous data.

Overall, the main difference between a bar chart and a histogram is that a bar chart is used for categorical data with distinct categories, while a histogram is used for continuous data with ranges of values.



here's an example for finding Q1 and Q3 when we have an odd number of data points:

Let's say we have the following dataset with 9 data points:

$$\{3, 7, 8, 10, 11, 12, 14, 15, 18\}$$

Sort the data in ascending order:

$$\{3, 7, 8, 10, 11, 12, 14, 15, 18\}$$

Find the median of the whole dataset:

$$\text{Median}(\{3, 7, 8, 10, 11, 12, 14, 15, 18\}) = 11$$

Note that the median value (11) is not included in either half of the split.

Split the data into two halves:

$$\text{Lower half: } \{3, 7, 8, 10\}$$

$$\text{Upper half: } \{12, 14, 15, 18\}$$

Find the median of the lower half:

$$Q1 = \text{Median}(\{3, 7, 8, 10\}) = 7.5$$

Find the median of the upper half:

$$Q3 = \text{Median}(\{12, 14, 15, 18\}) = 15$$

Therefore,  $Q1 = 7.5$  and  $Q3 = 15$  for this dataset.

When we have an even number of data points, finding Q1 (the first quartile) and Q3 (the third quartile) involves a different method than when we have an odd number of data points.

Here are the steps to find Q1 and Q3 for an even number of data points:

Sort the data in ascending order.

Split the data into two halves: the lower half and the upper half. If there are an even number of data points, each half should have the same number of points.

Find the median of the lower half. This is Q1.

Find the median of the upper half. This is Q3.

For example, let's say we have the following dataset with 10 data points:

{3, 6, 7, 8, 8, 9, 11, 12, 14, 16}

Sort the data in ascending order:

{3, 6, 7, 8, 8, 9, 11, 12, 14, 16}

Split the data into two halves:

Lower half: {3, 6, 7, 8, 8}

Upper half: {9, 11, 12, 14, 16}

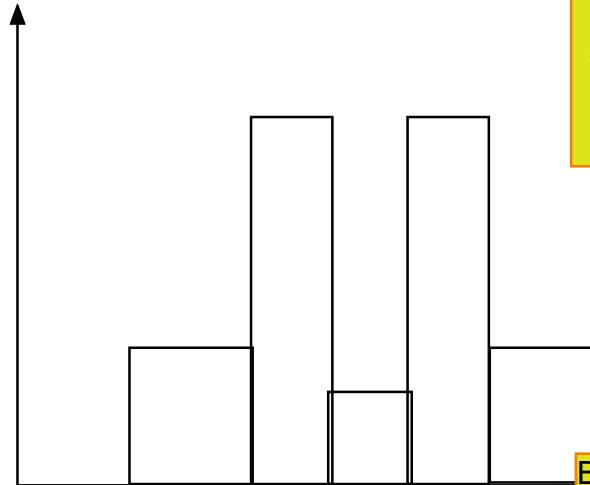
Find the median of the lower half:

$$Q1 = \text{Median}\{3, 6, 7, 8, 8\} = 7$$

Find the median of the upper half:

$$Q3 = \text{Median}\{9, 11, 12, 14, 16\} = 12$$

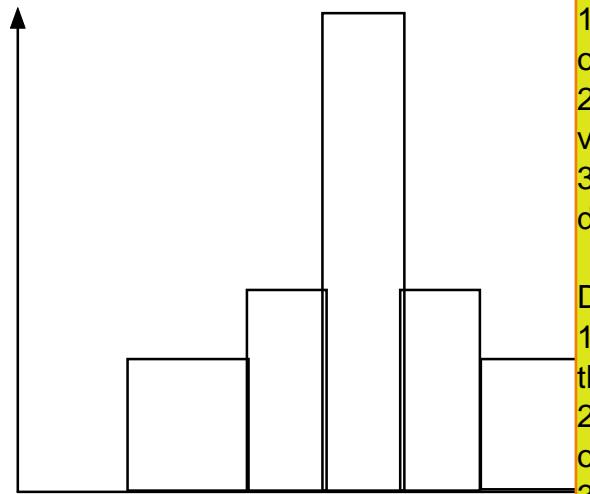
# Histograms Often Tell More than Boxplots



توی هیستوگرام راجع به کل طیف  
مقادیری که متغیر میگیره دید داریم.  
باکس پلات شکل اول و دوم مثل همه  
ولی توزیع داده هاشون خیلی فرق  
داره

The two histograms shown in the left  
may have the same boxplot  
representation

- The same values for: min, Q1, median, Q3, max
- But they have rather different data distributions



Boxplot Chart:

Advantages:

1. It shows the median, quartiles, and outliers in a dataset.
2. It is robust to outliers and extreme values.
3. It can be used to compare multiple datasets side by side.

Disadvantages:

1. It does not show the exact distribution of the data.
2. It may not be suitable for small datasets or datasets with few observations.
3. It may not be as intuitive as a histogram for some people.

Histogram Chart:

Advantages:

1. It shows the frequency distribution of a continuous variable.
2. It is easy to interpret and understand.
3. It can show the shape, center, and spread of the data.
4. It can be used to identify outliers.

Disadvantages:

1. It can be affected by the choice of bin size.
2. It may not be suitable for small datasets.
3. It does not show individual data points.

# Graphic Displays of Basic Statistical Descriptions

---

---

- **Boxplot:** graphic display of five-number summary
- **Histogram:** x-axis are values, y-axis repres. frequencies
- **Quantile plot:** each value  $x_i$  is paired with  $f_i$ , indicating that approximately  $100 f_i \%$  of data are  $\leq x_i$
- **Quantile-quantile (q-q) plot:** graphs the quantiles of one univariant distribution against the corresponding quantiles of another
- **Scatter plot:** each pair of values is a pair of coordinates and plotted as points in the plane

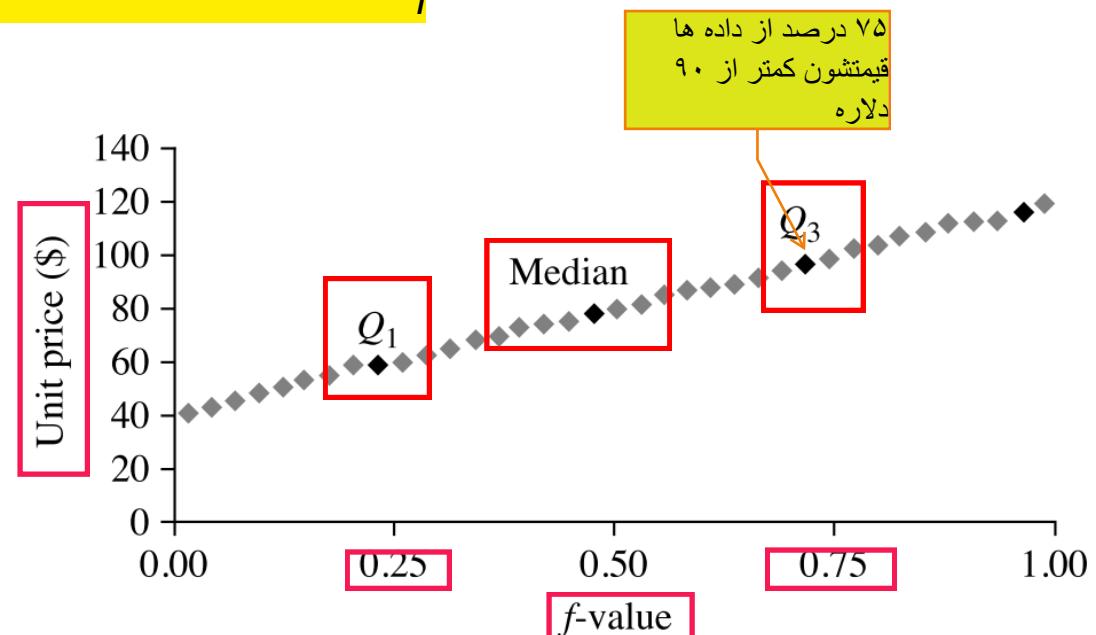
تعداد داده ها هرچی که باشه مبیریم  
به فضای ۱۰۰ درصد و چارک های  
اول و سومش رو حساب میکنیم مثل  
اگه ۱۲۰ تا داده داشته باشیم باز هم  
۱۰۰ مبیریم به

# Quantile Plot

- Displays all of the data (allowing the user to assess both the overall behavior and unusual occurrences)
- Plots quantile information
  - For a data  $x_i$  data sorted in increasing order,  $f_i$  indicates that approximately  $100 f\%$  of the data are below or equal to the value  $x_i$

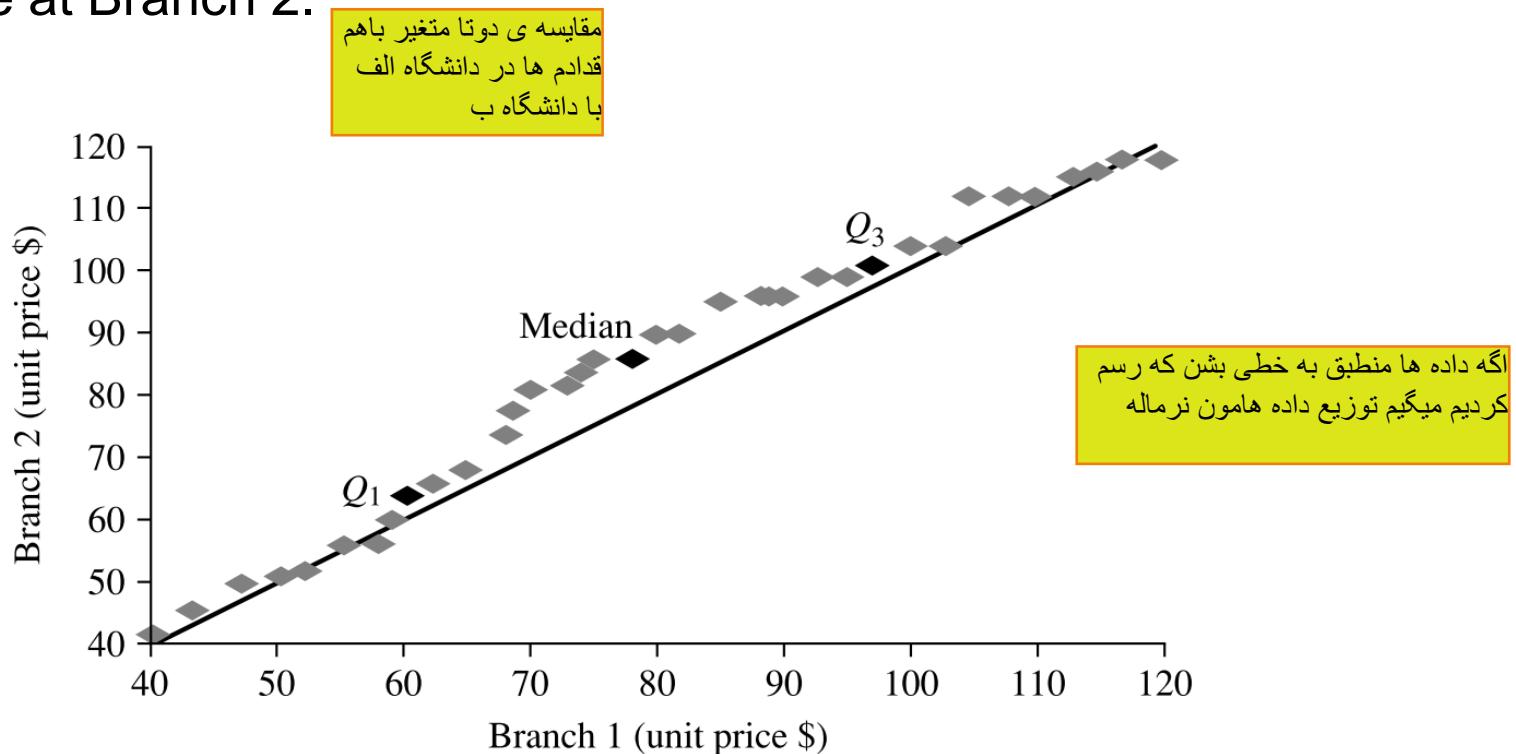
**Table 2.1** A Set of Unit Price Data for Items Sold at a Branch of AllElectronics

Unit price (\$)	Count of items sold
40	275
43	300
47	250
—	—
74	360
75	515
78	540
—	—
115	320
117	270
120	350



# Quantile-Quantile (Q-Q) Plot

- Graphs the quantiles of one univariate distribution against the corresponding quantiles of another
- View: Is there is a shift in going from one distribution to another?
- Example shows unit price of items sold at Branch 1 vs. Branch 2 for each quantile. Unit prices of items sold at Branch 1 tend to be lower than those at Branch 2.



How is a **quantile-quantile plot** different from a **quantile plot**?

A **quantile plot** is a graphical method used to show the **approximate percentage** of values below or equal to the **independent variable** in a univariate distribution. Thus, it displays **quantile information for all the data**, where the **values** measured for the independent variable are **plotted** against their corresponding **quantile**.

A **quantile-quantile plot** however, graphs the quantiles of **one univariate distribution against** the corresponding **quantiles of another univariate distribution**. Both axes display the range of values measured for their corresponding **distribution**, and points are plotted that correspond to the **quantile values** of the **two distributions**. A line ( $y = x$ ) can be added to the graph along with points representing where the first, second and third quantiles lie to increase the graph's informational value. Points that lie **above** such a line indicate a correspondingly **higher value** for the **distribution** plotted on the **y-axis** than for the distribution plotted on the **x-axis** at the same quantile. The opposite effect is true for points lying below this line.

Suppose a **hospital tested** the **age** and **body fat** data for 18 randomly selected adults with the following result

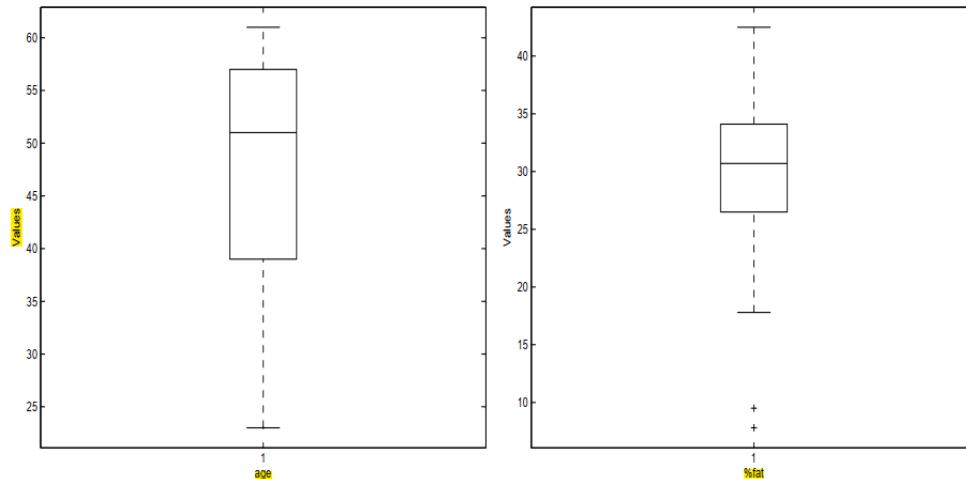
<b>age</b>	23	23	27	27	39	41	47	49	50
<b>%fat</b>	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2
<b>age</b>	52	54	54	56	57	58	58	60	61
<b>%fat</b>	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7

- (a) Calculate the **mean**, **median** and **standard deviation** of **age** and **%fat**.

For the variable **age** the **mean** is 46.44, the **median** is 51, and the **standard deviation** is 12.85. For the variable **%fat** the mean is 28.78, the median is 30.7, and the standard deviation is 8.99.

- (b) Draw the boxplots for **age** and **%fat**.

See Figure 2.1.



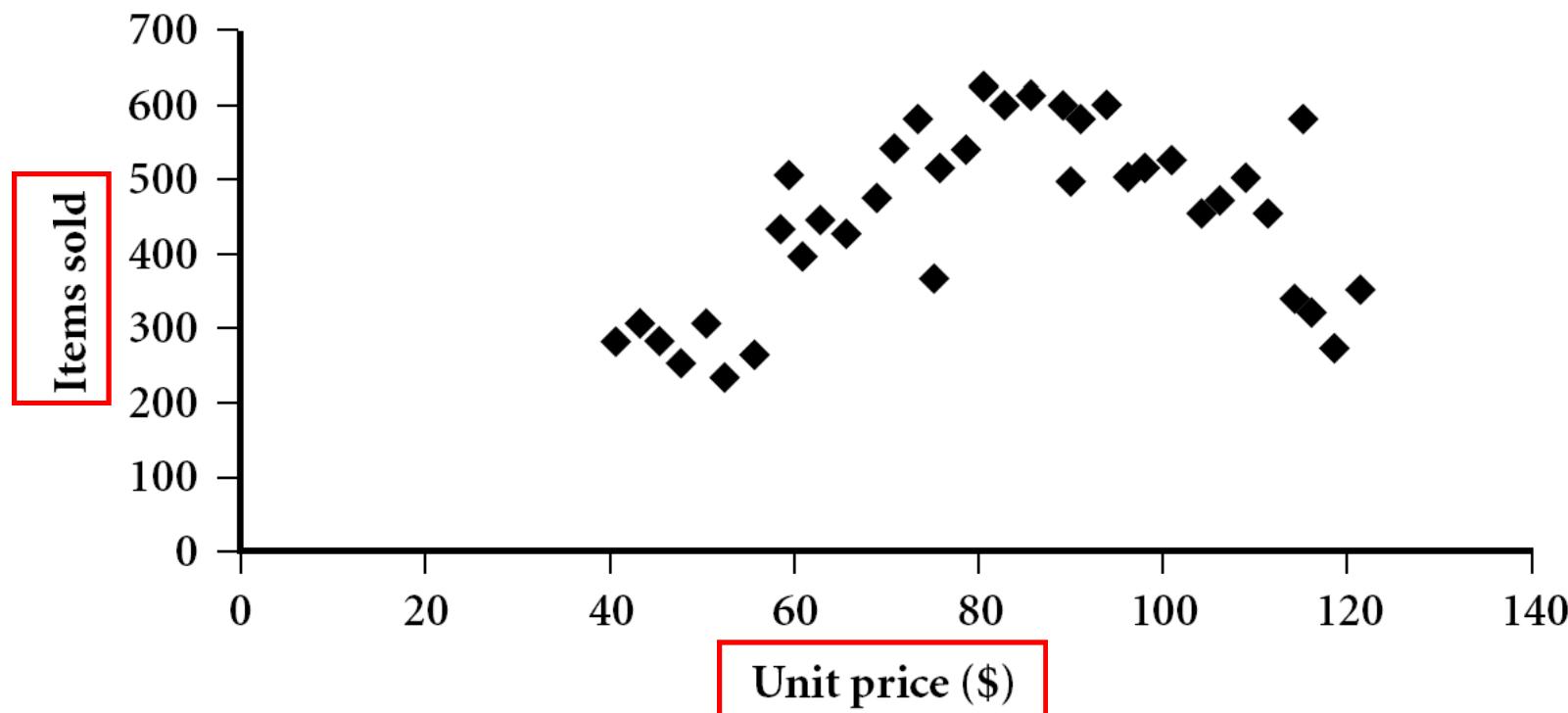
- (d) **Normalize** the two variables based on ***z-score normalization***.

<b>age</b>	23	23	27	27	39	41	47	49	50
<b><i>z-age</i></b>	-1.83	-1.83	-1.51	-1.51	-0.58	-0.42	0.04	0.20	0.28
<b>%fat</b>	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2
<b><i>z-%fat</i></b>	-2.14	-0.25	-2.33	-1.22	0.29	-0.32	-0.15	-0.18	0.27
<b>age</b>	52	54	54	56	57	58	58	60	61
<b><i>z-age</i></b>	0.43	0.59	0.59	0.74	0.82	0.90	0.90	1.06	1.13
<b>%fat</b>	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7
<b><i>z-%fat</i></b>	0.65	1.53	0.0	0.51	0.16	0.59	0.46	1.38	0.77

# Scatter plot

داده های دو متغیره

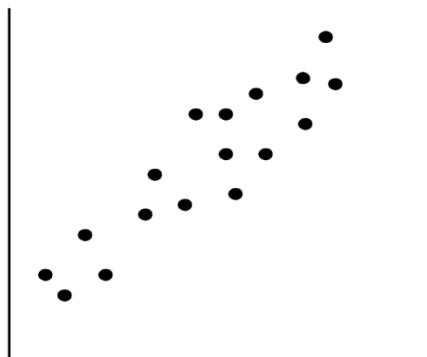
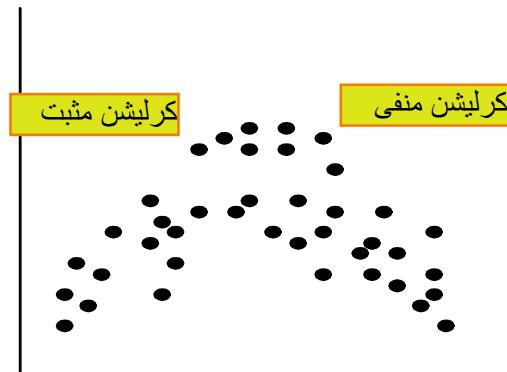
- Provides a first look at bivariate data to see clusters of points, outliers, etc
- Each pair of values is treated as a pair of coordinates and plotted as points in the plane



# Positively and Negatively Correlated Data

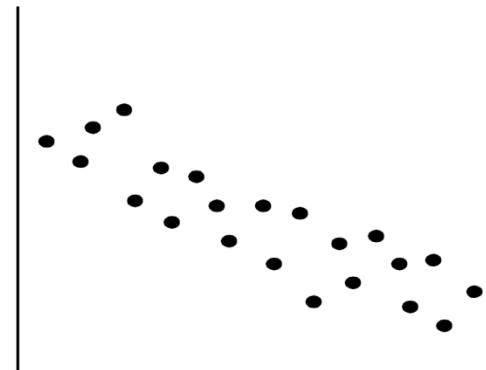
- The left half fragment is positively correlated
- The right half is negative correlated

سن و قد تا بیست سالگی رابطه مثبت داره  
از بیست سالگی به بعد مثلا ارتباطی وجود  
نداره یعنی سن بالا میره ولی قد ثابت میمونه



Positively correlated

شیب خط مثبت است



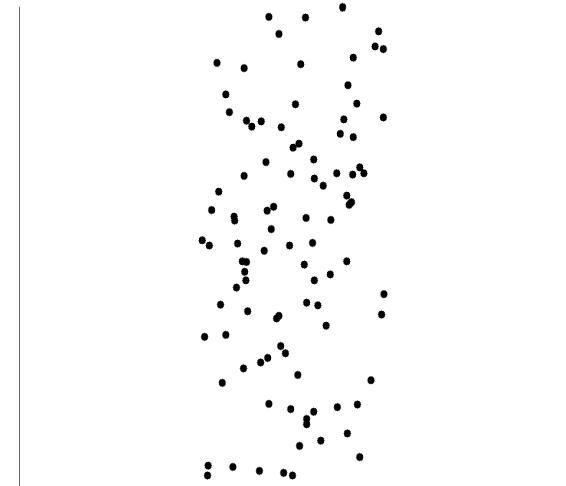
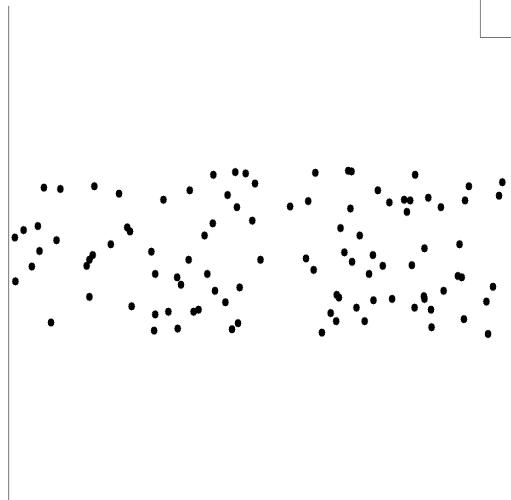
Negatively correlated

شیب خط منفی است

# Uncorrelated Data

یه شکل مستطیلی پیدا  
میکنه

زمانی که اتریبیوت های داده هامون هیچ ربطی به هم نداشته باشن مثلا اگه بگیم مقدار اتریبیوت اول شد ۱۰ لزوما نمیتونیم بگیم مقدار اتریبیوت دوم هم میشه ۱۰ مثلا ممکنه بشه ۲۰





# Introduction To Data Mining

Isfahan University of Technology (IUT)  
Bahman 1401



Getting to Know Your Data

Dr. Hamidreza Hakim  
[hamid.hakim.u@gmail.com](mailto:hamid.hakim.u@gmail.com)

چرا سراغ ویژوالیزیشن میریم؟

یه دیدی از داده ها بمون میده، بمون کمک میکنه بفهمیم کل داده هامون تو ش چه خبره  
پیدا کردن یه دید کیفی از داده ها  
پیدا کردن الگو در داده ها مثلًا بگیم ابتدا یه کرلیشن مثبت داریم بعد یه کرلیشن منفی  
داریم  
برای بهینه کردن پارامترها در انتخاب ابزارها تصمیم بگیریم چی را برداریم؟  
ابزارهای مختلفی برای ویژوالیزیشن هست

# DATA VISUALIZATION

# Data Visualization

- Why data visualization?

- Gain insight into an information space by mapping data onto graphical primitives
- Provide qualitative overview of large data sets
- Search for patterns, trends, structure, irregularities, relationships among data
- Help find interesting regions and suitable parameters for further quantitative analysis
- Provide a visual proof of computer representations derived

- Categorization of visualization methods:

- Pixel-oriented visualization techniques
- Geometric projection visualization techniques
- Icon-based visualization techniques
- Hierarchical visualization techniques
- Visualizing complex data and relations

با نگاشت داده ها بر روی نمونه های اولیه گرافیکی، بینش نسبت به فضای اطلاعاتی به دست می اوریم.

ارائه نمای کلی کیفی از مجموعه داده های بزرگ.  
جستجو برای الگوهای روندها، ساختار، بی نظمی ها، روابط بین داده ها.

به یافتن مناطق جالب و پارامترهای مناسب برای تجزیه و تحلیل کمی بیشتر کمک میکند.

ارائه یک مدرک بصری از نمایش های کامپیوتری به دست آمده

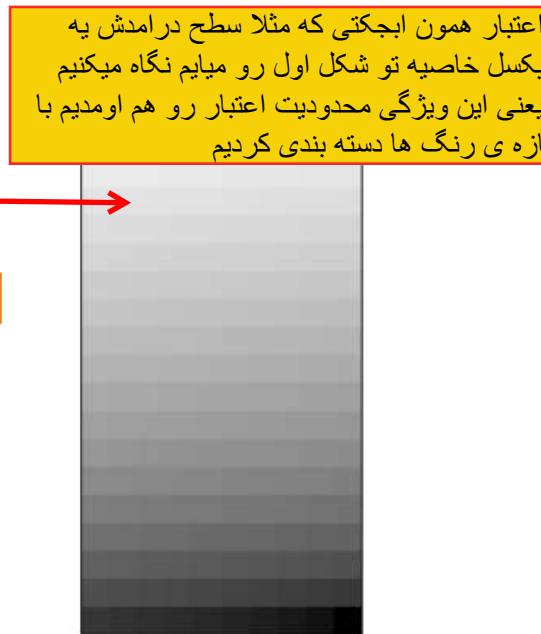
# Pixel-Oriented Visualization Techniques

- For a data set of  $m$  dimensions, create  $m$  windows on the screen, one for each dimension
- The  $m$  dimension values of a record are mapped to  $m$  pixels at the corresponding positions in the windows
- The colors of the pixels reflect the corresponding values

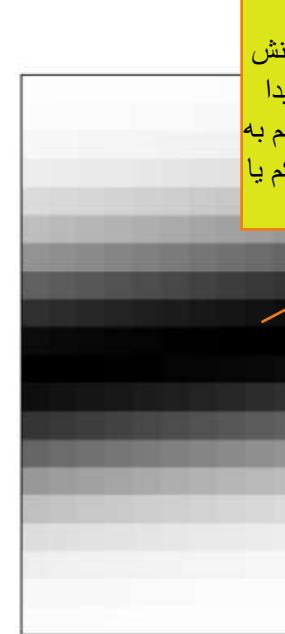
مثلای اینجا که ۴ تا اتریبیوت داریم، ۴ تا نجره کشیده شده که مقدار اتریبیوت های هر رکورد یا ابجکت منتظر رسم شده



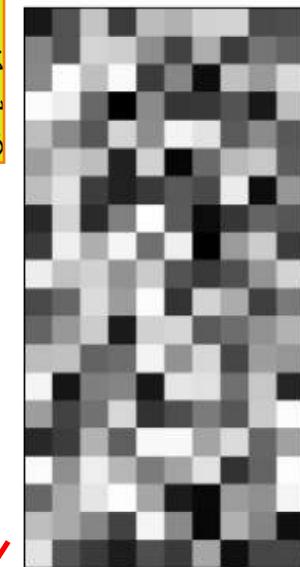
(a) Income



(b) Credit  
Limit



(c) transaction  
volume



(d) age

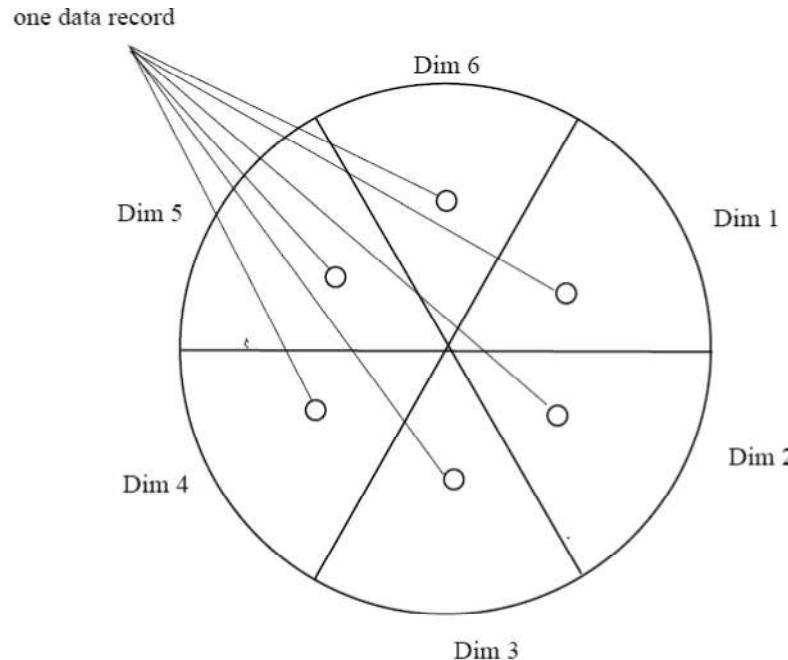
به ازای هر مقداری که سطح درامدشون داره، یه پیکسلی میسازیم که میزان سطح درامدشون با رنگ پیکسله مرتبط و قدری سطح درامد زیاد شد، رنگ مشکی تر میشه اگه سطح درامد کم شد رنگ سفید

هر پیکسل نظیر یک ابجکت است

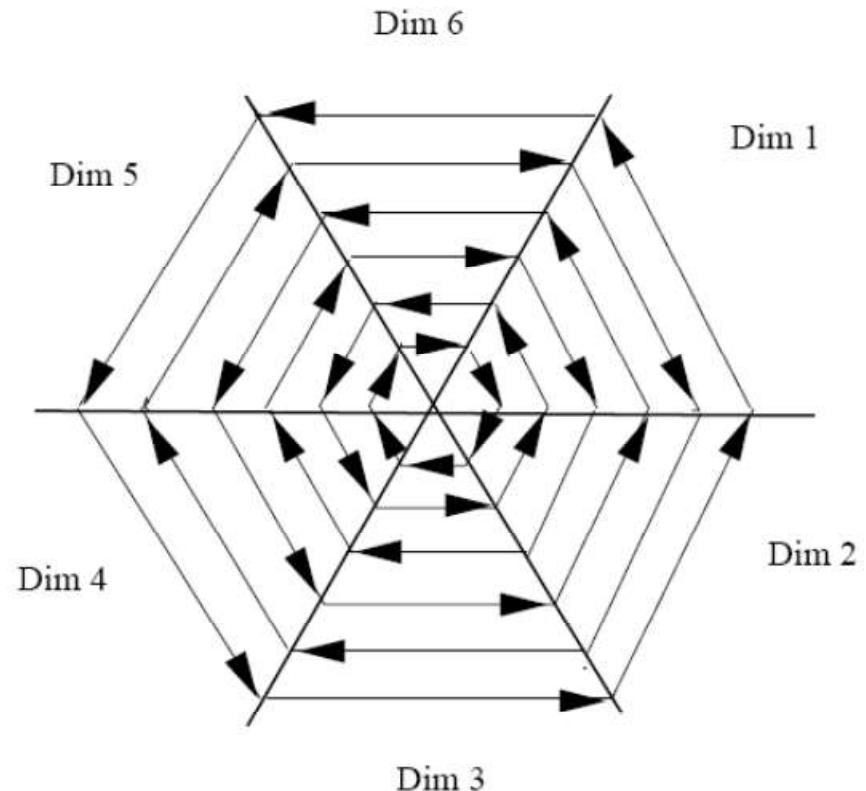
رابطه ای خاصی بین سن و سطح درامد وجود نداره

# Laying Out Pixels in Circle Segments

- To save space and show the connections among multiple dimensions, space filling is often done in a circle segment



**(a)** Representing a data record in circle segment



**(b)** Laying out pixels in circle segment

# Geometric Projection Visualization Techniques

---

---

- Visualization of geometric transformations and projections of the data تجسم تبدیل های هندسی و پیش بینی داده ها
- Methods
  - Scatterplot and scatterplot matrices
  - Landscapes
  - Parallel coordinates

طرح ها

# Scatterplot Matrices

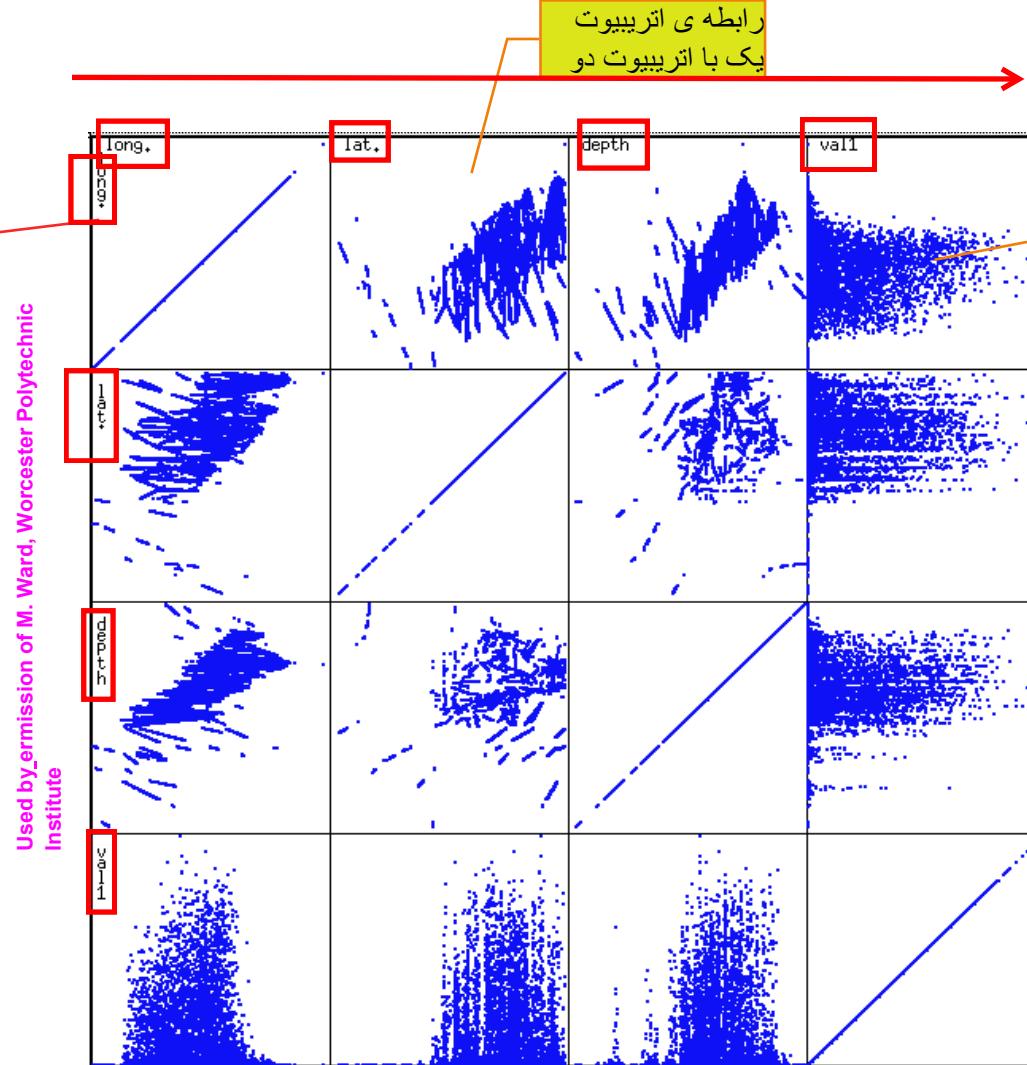
اتریبیوت یک با خودش  
چه رابطه ای داشته؟  
به ازای هر ایکسی از ش  
همون ایکس را میگیریم  
پس میشه  $y=x$   
داشتن رابطه ی یک  
تریبیوت با خودش به چه  
دردی میخوره؟

طرح پراکنده  
تکنیکی که یک تصویر و  
دیدکلی از داده ها میده

رابطه ی اتریبیوت  
یک با اتریبیوت دو

چهارتا اتریبیوت را در صفحه  
ی ایکس و وای رسم میکنیم  
میشه یه جدول مانند ۴ در ۴

هربجکتمون یک نقطه  
ای میشه

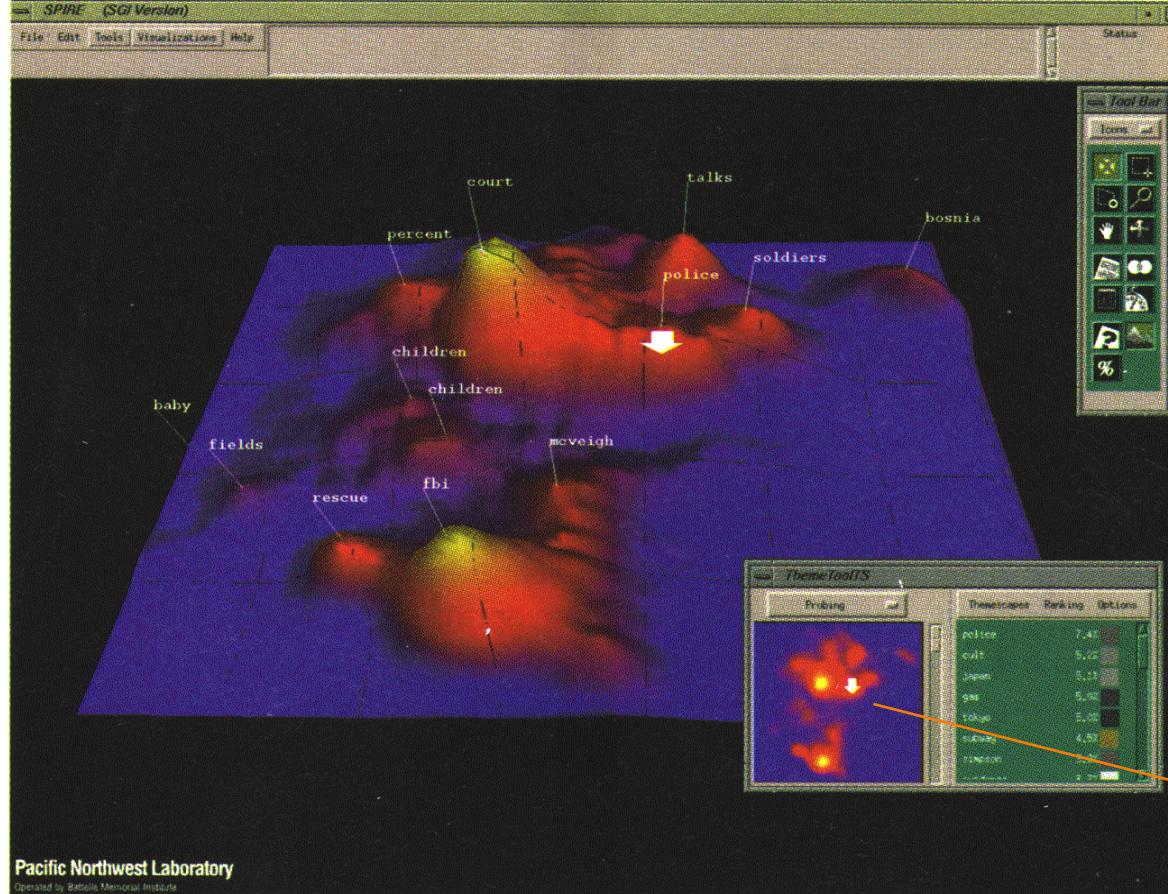


Matrix of scatterplots (x-y-diagrams) of the k-dim. data [total of  $(k^2/2-k)$  scatterplots]

# Landscapes

علاوه برایکس و وای بعد زد هم اضافه میشه برای نمایش اطلاعات جاهایی که با ۳تا اتریبیوت کار داریم از این نمودارها استفاده میشه مثلًا میخایم بینیم با مقدار ایکس از اتریبیوت یک و مقدار وای از اتریبیوت ۲ ، اتریبیوت ۳ چه مقداری پیدا کرده؟ با سطح ارتفاع یا رنگ مقادیر اتریبیوت ها را نشان میده

Used by permission of B. Wright, Visible Decisions Inc.



news articles  
visualized as  
a landscape

از بالا به تصویر سه  
بعدی نگاه کنیم این شکل  
را میبینیم

- Visualization of the data as perspective landscape
- The data needs to be transformed into a (possibly artificial) **2D spatial** representation which preserves the characteristics of the data

# Parallel Coordinates

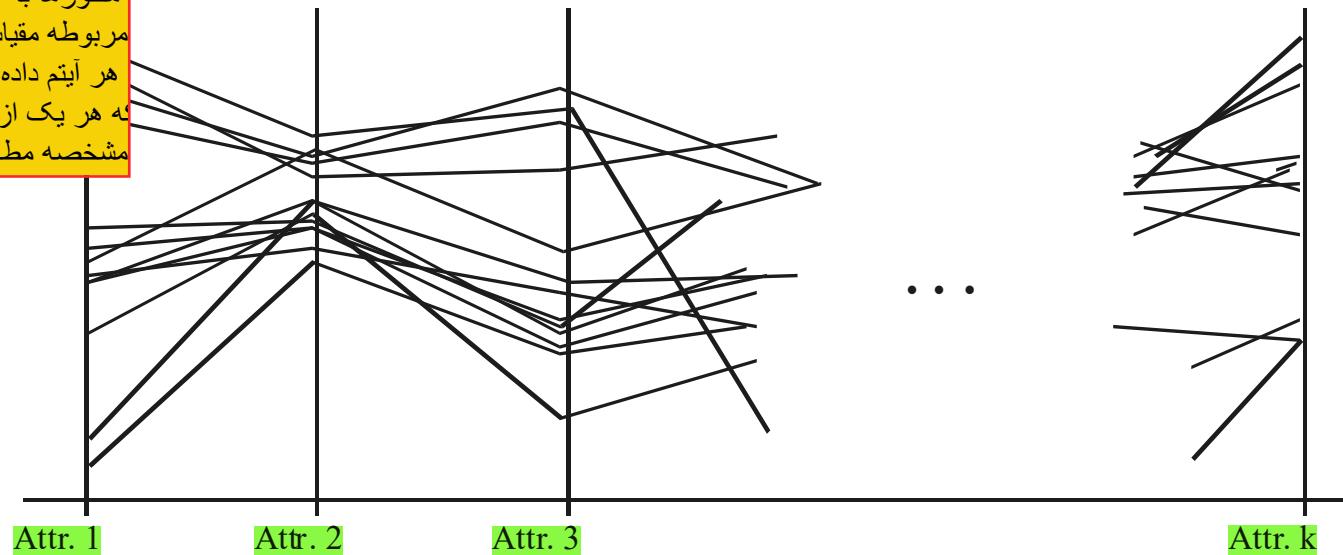
همنواهی بین رفتارهای  
اتریبیوت ها را نشان میده

کنار هم گذاشتن اتریبیوت های مختلف  
ما چندین اتریبیوت داریم که هر کدام یه مقدار میں و  
یه مقدار مaks داره  
هر ایجکتی هم از هر اتریبیوتی یک مقدار داره

مشاهده ی توده ی رفتارهای کلی مثل مفهومیم اونایی که  
قدشون کمeh و وزنشون کمeh مدلشون بالاست!

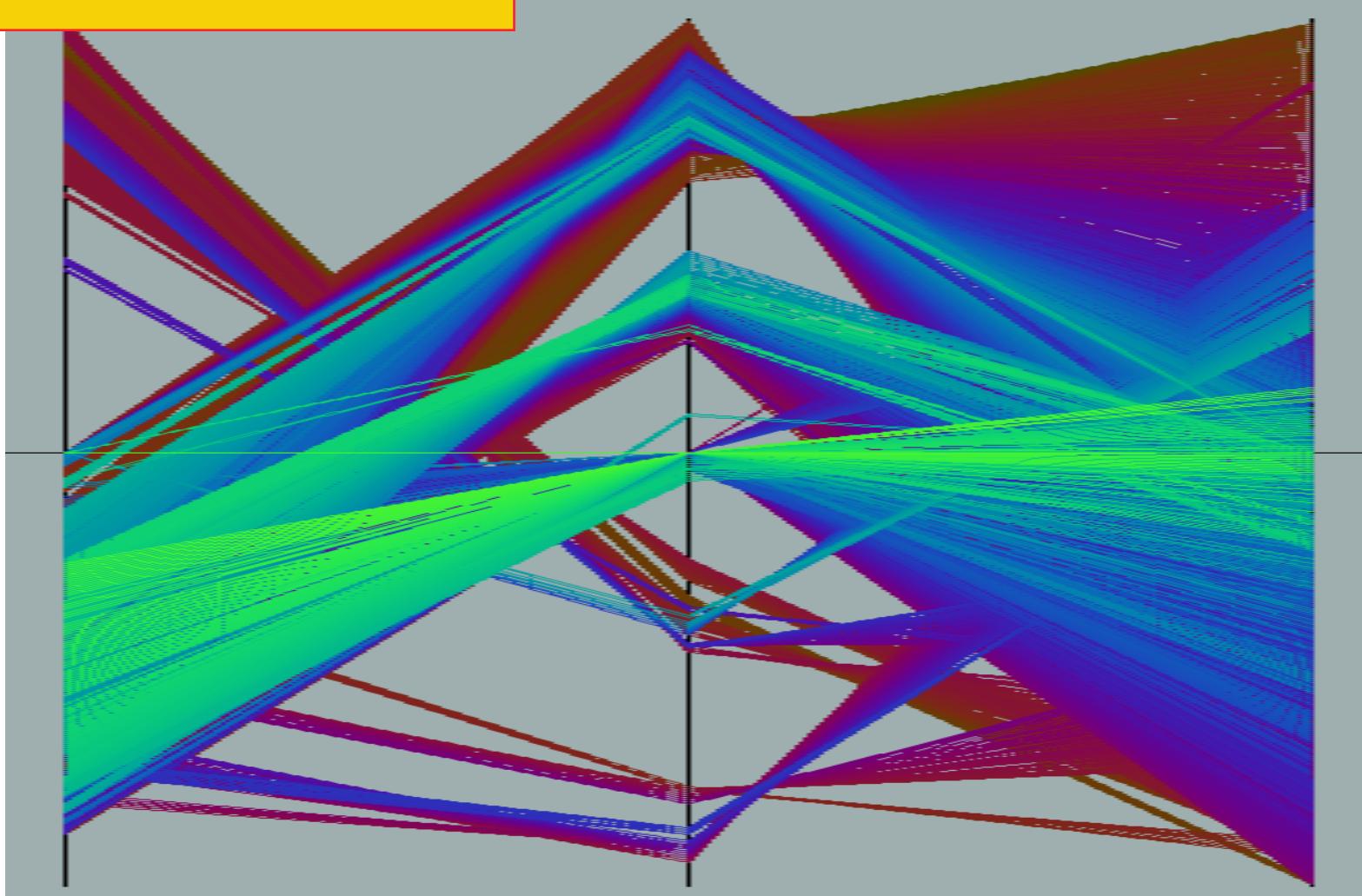
- **n equidistant axes** which are parallel to one of the screen axes and correspond to the attributes
- The **axes are scaled** to the **[minimum, maximum]**: range of the corresponding attribute
- Every data item corresponds to a **polygonal line** which intersects each of the axes at the point which corresponds to the value for the attribute

n محور مساوی که با یکی از محورهای صفحه موازی هستند و با ویژگی ها مطابقت دارند.  
محورها به [حداقل، حداکثر]: محدوده ویژگی مربوطه مقیاس می شوند.  
هر آیتم داده مربوط به یک خط چند ضلعی است که هر یک از محورها را در نقطه ای که با مقدار مشخصه مطابقت دارد قطع می کند.



# Parallel Coordinates of a Data Set

کاربرد: مثلا برای تقسیم کردن داده ها به دسته های مختلف مثل  
کلاسترینگ نمونه ها  
برای پیدا کردن یک سری پترن و الگو از نمونه ها



# Icon-Based Visualization Techniques

---

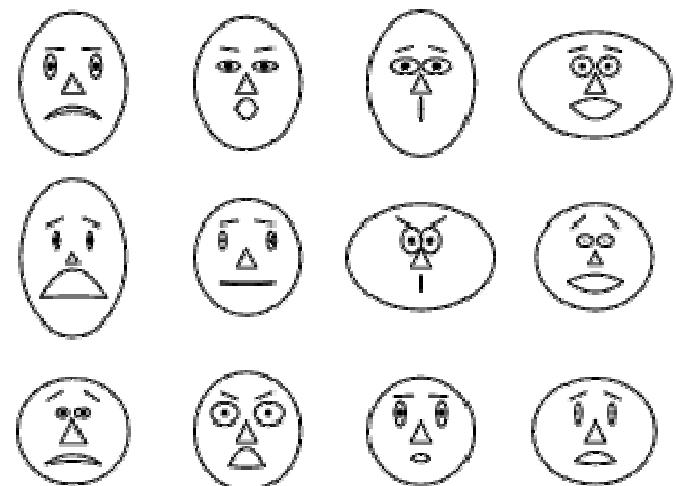
- Visualization of the data values as features of icons
- Typical visualization methods
  - Chernoff Faces
  - Stick Figures
- General techniques
  - Shape coding: Use shape to represent certain information encoding
  - Color icons: Use color icons to encode more information
  - Tile bars: Use small icons to represent the relevant feature vectors in document retrieval

# Chernoff Faces

---

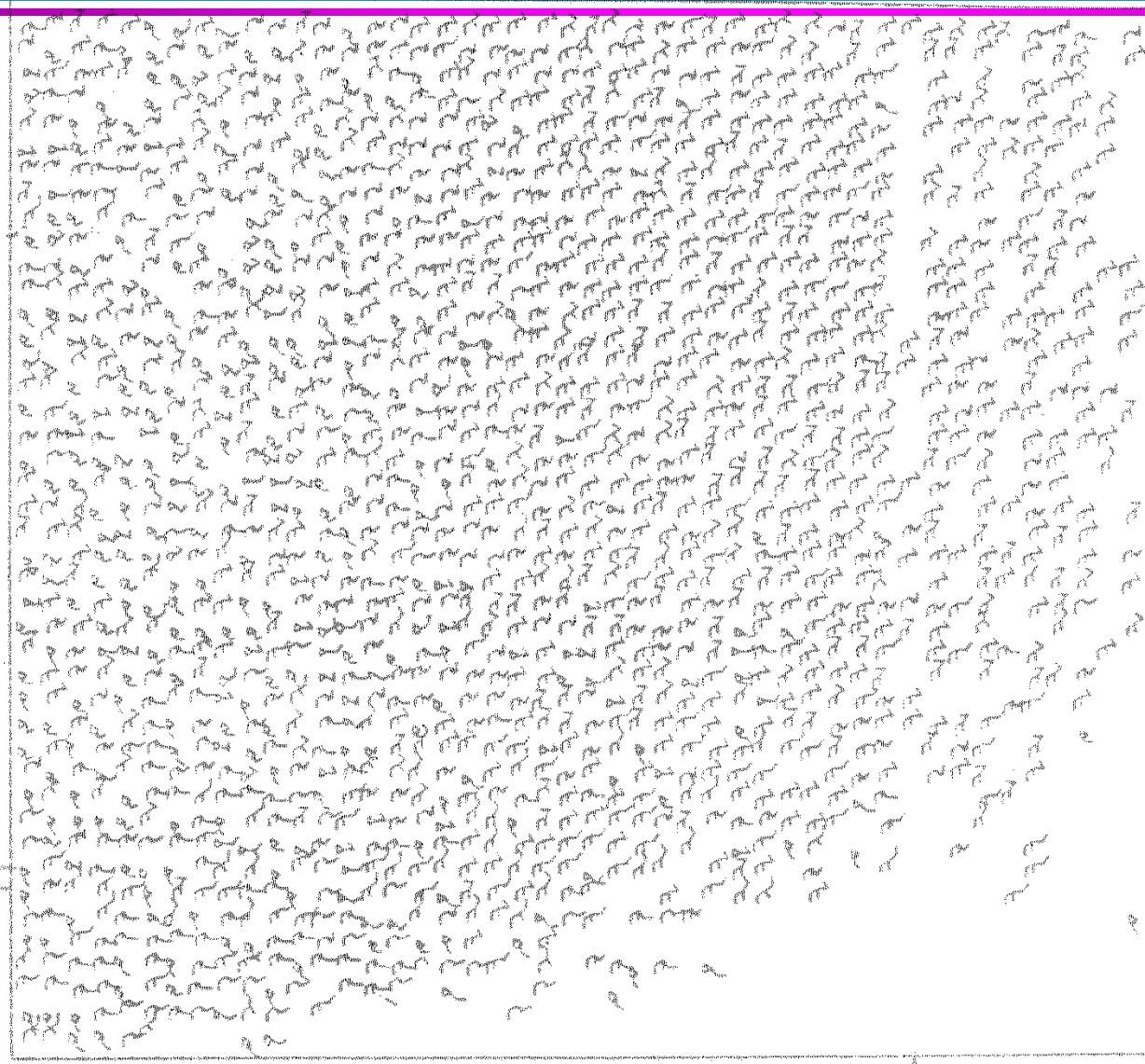
---

- A way to display variables on a two-dimensional surface, e.g., let  $x$  be eyebrow slant,  $y$  be eye size,  $z$  be nose length, etc.
- The figure shows faces produced using 10 characteristics--head eccentricity, eye size, eye spacing, eye eccentricity, pupil size, eyebrow slant, nose size, mouth shape, mouth size, and mouth opening): Each assigned one of 10 possible values, generated using *Mathematica* (S. Dickson)
- REFERENCE: Gonick, L. and Smith, W. *The Cartoon Guide to Statistics*. New York: Harper Perennial, p. 212, 1993
- Weisstein, Eric W. "Chernoff Face." From *MathWorld--A Wolfram Web Resource*. [mathworld.wolfram.com/ChernoffFace.html](http://mathworld.wolfram.com/ChernoffFace.html)



# Stick Figure

used by permission of G. Grinstein, University of Massachusetts at Lowell



INCOME

60

Two attributes mapped to axes, remaining attributes mapped to angle or length of limbs". Look at texture pattern

A census data figure  
showing age, income,  
gender, education, etc.

A 5-piece stick figure (1  
body and 4 limbs w.  
different angle/length)

---

---

# **SIMILARITY AND DISSIMILARITY MEASURES**

ما انسان‌ها خودمون هم وقتی میخاییم قضاوت کنیم دنبال شباخته‌ها میگردیم توی تاریخچه مغز‌مون میگیم این ابجکت شبیه به کدوم ابجکتی است که قبلاً باش برخورد کردم و چه طوری برخورد کردم باش که الان با این جدیده هم همون طوری رفتار کنم

معیارهای پیداکردن  
شباخته و تفاوت

# Similarity and Dissimilarity Measures

## ● Similarity measure

- Numerical measure of how alike two data objects are.
- Is higher when objects are more alike.
- Often falls in the range [0, 1]

## ● Dissimilarity measure

- Numerical measure of how different two data objects are
- Lower when objects are more alike
- Minimum dissimilarity is often 0
- Upper limit varies

## ● Proximity refers to a similarity or dissimilarity

نزدیکی یا مجاورت

مثلاً اگه میخاییم ببینیم یه فردی یه درسی رو پاس میکنه یا نه میبینیم نسبت به ادم قبلی ها چقدر فاصله داره اگه شبیه درس خوان کلاس باشه به احتمال زیاد هم پاس میکنه

دوتا ابجکت که شبیه هم هستند مقدار بیشتری بده  
اگه هم شبیه هم نیستند صفر بده

رداده کاوی دنبال شباخته یابی هستیم یعنی دوست داریم که شباخته‌ها را در یک مساله کشف کنیم این طوری نسبت به داده‌ها بی‌پایان می‌شیم یعنی وقتی یه داده‌ی جدیدی اومد برحسب شباخته‌ش به داده‌های قبلی دربارش تصمیم میگیریم

# Similarity/Dissimilarity for Simple Attributes

The following table shows the similarity and dissimilarity between two objects,  $x$  and  $y$ , with respect to a single, simple attribute.

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

رنگ چشم مثلا سبز و قرمز وابی  
تیم های لیگ برتر مثلا  
شباهت میشه مساوی هستند یا نه؟

کوچک متوسط بزرگ  
مثلا اندازه ای موبایل یا توب فوتبال  
کیفیت یک کالا : خوب عالی بد

يشترین ميزان فاصله بين  
دوتا ايجكت

چون ميخايم رنج را  
نرمال كنيم و بيريم به  
باشه ي صفر و يك

براي مقاييسه ى دوتا کارخانه که ماشين توليد ميکنند ميخايم  
کيفيتون را مقاييسه کنيم  
باید به عددی به مقدار کیفیت ها بدمیم مثل صفر یک دو  
با نسبت دادن عدد، داده ها را برای مقاييسه اماده ميکنیم  
شاید اندازه ای رنج و طيف خوب و بد بودن و عالي بودن در  
یک مساله متفاوت باشه  
مثل رنج عددی در خوب بودن خيلي بيشتر از رنج در حالت  
بد باشه

# Euclidean Distance

- Euclidean Distance

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2}$$

مقدار اtribیوت کم از  
بجکت ایکس را از مقدار  
tribیوت کم از بجکت y  
کم کن و به توان ۲  
برسان.

تعداد اtribیوت ها

پیداکردن فاصله ی بین دو تا ابجکت

where  $n$  is the number of dimensions (atributes) and  $x_k$  and  $y_k$  are, respectively, the  $k^{th}$  attributes (components) or data objects  $\mathbf{x}$  and  $\mathbf{y}$ .

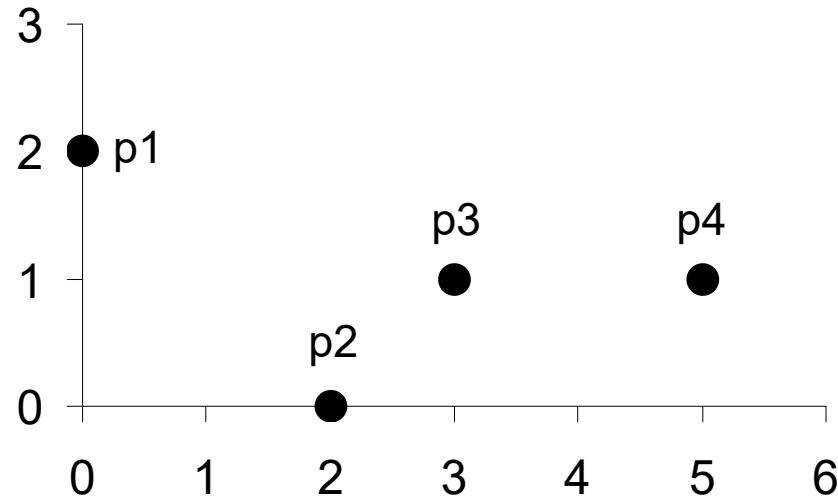
- Standardization is necessary, if scales differ.

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

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

# Euclidean Distance

وتر مثلث قائم الزاويه



مقادير اtribوبوت های ایکس و واي

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

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

Distance Matrix

بين يه اجككت با خودش هیچ  
فاصله ای نیست دیگه

# Minkowski Distance

- Minkowski Distance is a generalization of Euclidean Distance

$$d(\mathbf{x}, \mathbf{y}) = \left( \sum_{k=1}^n |x_k - y_k|^r \right)^{1/r}$$

Where  $r$  is a parameter,  $n$  is the number of dimensions (attributes) and  $x_k$  and  $y_k$  are, respectively, the  $k^{\text{th}}$  attributes (components) or data objects  $x$  and  $y$ .

**Minkowski distance** is a generalization of the Euclidean and Manhattan distances. It is defined as

$$d(i, j) = \sqrt[p]{|x_{i1} - x_{j1}|^p + |x_{i2} - x_{j2}|^p + \cdots + |x_{ip} - x_{jp}|^p}, \quad (2.18)$$

where  $p$  is a real number such that  $p \geq 1$ . (Such a distance is also called  $L_p$  norm in some literature, where the symbol  $p$  refers to our notation of  $h$ . We have kept  $p$  as the number of attributes to be consistent with the rest of this chapter.) It represents the Manhattan distance when  $p = 1$  (i.e.,  $L_1$  norm) and Euclidean distance when  $p = 2$  (i.e.,  $L_2$  norm).

# Minkowski Distance: Examples

---

- $r = 1$ . City block (Manhattan, taxicab,  $L_1$  norm) distance.
  - A common example of this for binary vectors is the Hamming distance, which is just the number of bits that are different between two binary vectors
- $r = 2$ . Euclidean distance
- $r \rightarrow \infty$ . “supremum” ( $L_{\max}$  norm,  $L_\infty$  norm) distance.
  - This is the maximum difference between any component of the vectors

Euclidean distance and Manhattan distance. Let  $x_1 = (1, 2)$  and  $x_2 = (3, 5)$  represent two objects as shown in Figure 2.23. The Euclidean distance between the two is  $\sqrt{2^2 + 3^2} = 3.61$ . The Manhattan distance between the two is  $2 + 3 = 5$ . ■
- Do not confuse  $r$  with  $n$ , i.e., all these distances are defined for all numbers of dimensions.

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

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

1) Calculate the Minkowski distance of order 2 between points (1, 2, 3) and (4, 5, 6):

$$\text{Minkowski distance} = ((4-1)^2 + (5-2)^2 + (6-3)^2)^{(1/2)}$$

Minkowski distance = 5.196

2) Calculate the Minkowski distance of order 3 between points (0, 0, 0) and (1, 2, 3):

$$\text{Minkowski distance} = ((1-0)^3 + (2-0)^3 + (3-0)^3)^{(1/3)}$$

Minkowski distance = 3.301

3) Calculate the Minkowski distance of order 4 between points (-1, -2) and (3, 4):

$$\text{Minkowski distance} = ((3-(-1))^4 + (4-(-2))^4)^{(1/4)}$$

Minkowski distance = 6.211

4) Calculate the Minkowski distance of order 5 between points (2, -5, 1) and (-3, 7, -2):

$$\text{Minkowski distance} = ((-3-2)^5 + (7-(-5))^5 + (-2-1)^5)^{(1/5)}$$

Minkowski distance = 10.342

5) Calculate the Minkowski distance of order 6 between points (0, 0) and (5, -12):

$$\text{Minkowski distance} = ((5-0)^6 + (-12-0)^6)^{(1/6)}$$

Minkowski distance = 12.069

1) Calculate the Euclidean distance between points (1, 2) and (4, 6):

$$\text{Euclidean distance} = ((4-1)^2 + (6-2)^2)^{(1/2)}$$

Euclidean distance = 5

2) Calculate the Euclidean distance between points (-3, 0) and (0, 4):

$$\text{Euclidean distance} = ((0-(-3))^2 + (4-0)^2)^{(1/2)}$$

Euclidean distance = 5

3) Calculate the Euclidean distance between points (2, -5, 1) and (-3, 7, -2):

$$\text{Euclidean distance} = ((-3-2)^2 + (7-(-5))^2 + (-2-1)^2)^{(1/2)}$$

Euclidean distance = 13

4) Calculate the Euclidean distance between points (0, 0, 0) and (1, 2, 3):

$$\text{Euclidean distance} = ((1-0)^2 + (2-0)^2 + (3-0)^2)^{(1/2)}$$

Euclidean distance = 3.742

5) Calculate the Euclidean distance between points (-1, -2, -3) and (4, 5, 6):

$$\text{Euclidean distance} = ((4-(-1))^2 + (5-(-2))^2 + (6-(-3))^2)^{(1/2)}$$

Euclidean distance = 11.225

# Minkowski Distance

جمع تفاوت ایکس ها و وای ها باهمدیگه  
 $2+2 = 4$

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

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

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

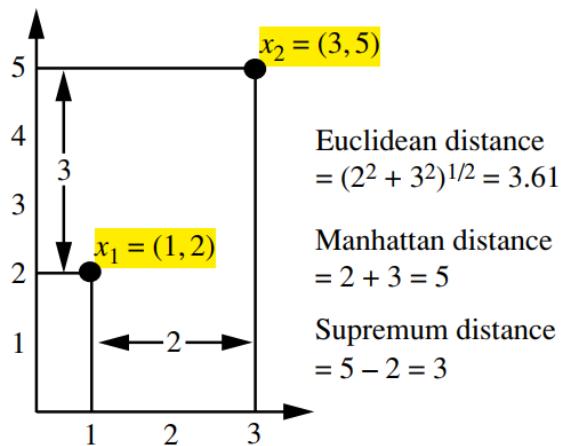
L $\infty$	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Distance Matrix

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*.



**Supremum distance.** Let's use the same two objects,  $x_1 = (1, 2)$  and  $x_2 = (3, 5)$ , as in Figure 2.23. The second attribute gives the greatest difference between values for the objects, which is  $5 - 2 = 3$ . This is the supremum distance between both objects. ■

میخایم به بعدهای ایکس و وای به شکل وزن دار نگاه کنیم

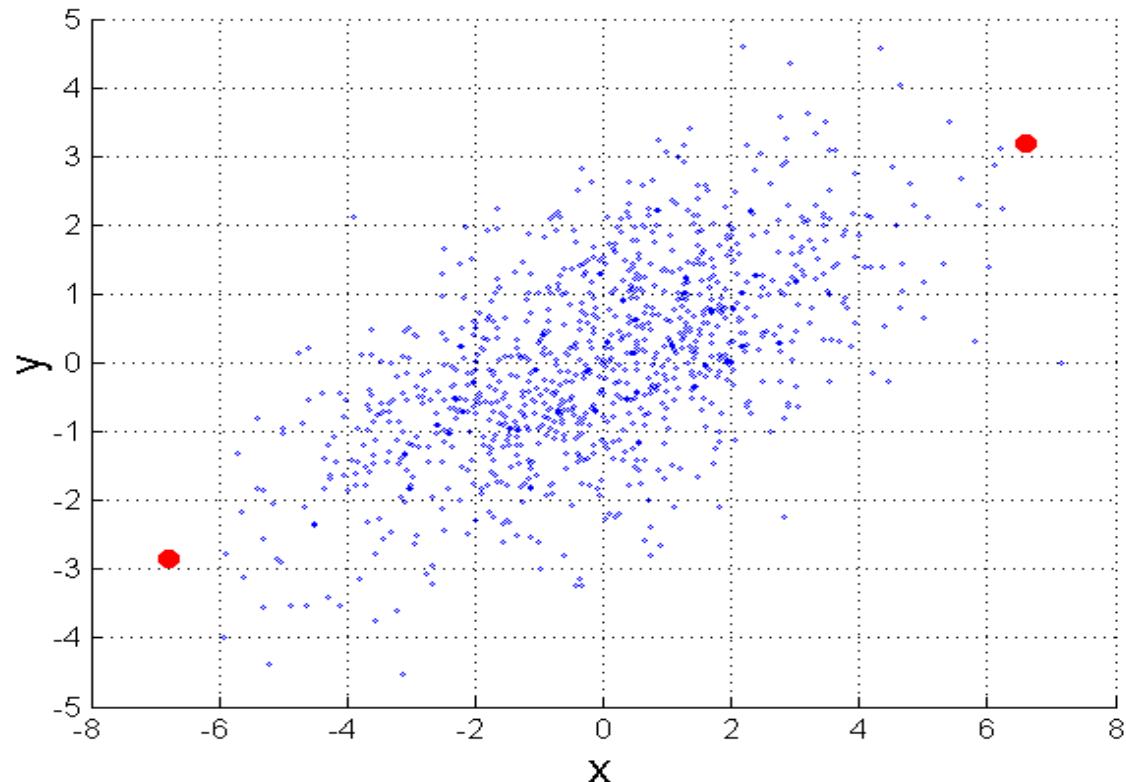
# Mahalanobis Distance

The Mahalanobis distance is calculated for each point, and those that are far away from the center of the distribution (i.e., have a high Mahalanobis distance) are considered outliers.

Mahalanobis distance is a measure of the distance between a point and a distribution. It takes into account the covariance between variables, which makes it useful for multivariate data analysis. The formula for Mahalanobis distance is:

$$D^2 = (x - \mu)' \Sigma^{-1} (x - \mu)$$

where D is the Mahalanobis distance, x is the vector of observations,  $\mu$  is the vector of means, and  $\Sigma^{-1}$  is the inverse covariance matrix.



نسبت به کواریانس میابیم  
فاصله سنجی میکنیم

$\Sigma$  is the covariance matrix

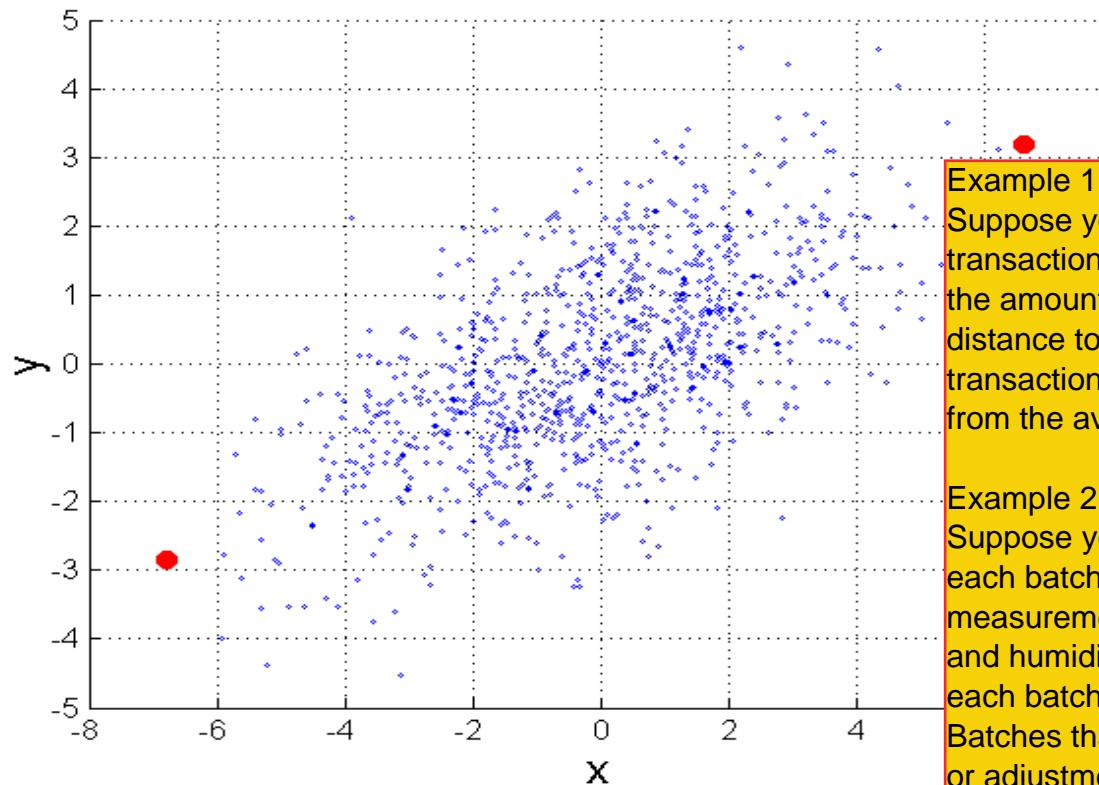
$$\begin{bmatrix} \text{Var}(x_1) & \dots & \text{Cov}(x_n, x_1) \\ \vdots & \ddots & \vdots \\ \text{Cov}(x_n, x_1) & \dots & \text{Var}(x_n) \end{bmatrix}$$

$$\text{var}(x) = \frac{\sum_{i=1}^n (x_i - \mu)^2}{n}$$

$$\text{cov}(x, y) = \frac{\sum_{i=1}^n (x_i - \mu_x)(y_i - \mu_y)}{n}$$

# Mahalanobis Distance

$$\text{mahalanobis}(\mathbf{x}, \mathbf{y}) = ((\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y}))^{-0.5}$$



$\Sigma$  is the covariance matrix

Example 1: Fraud detection

Suppose you are working for a bank and you want to detect fraudulent transactions. You have data on customers' transaction history, such as the amount spent, location, and time of day. You can use Mahalanobis distance to calculate how far each transaction is from the average transaction in terms of these variables. Transactions that are far away from the average may be flagged as potentially fraudulent.

Example 2: Quality control

Suppose you are manufacturing a product and you want to ensure that each batch meets certain quality standards. You have data on various measurements taken during production, such as temperature, pressure, and humidity. You can use Mahalanobis distance to calculate how far each batch is from the average batch in terms of these variables.

Batches that are far away from the average may need further inspection or adjustment.

For red points, the Euclidean distance is 14.7, Mahalanobis distance is 6.

اگه پراکندگی داده ها یکسان باشه بنی یه  
دایره تشکیل بده فاصله ای اقلیدسی و  
مهلانوبیس یکی میشه

The formula for covariance between two variables X and Y is:

$$\text{cov}(X, Y) = (\sum[(X_i - \mu_X)(Y_i - \mu_Y)]) / (n - 1)$$

where:

$\Sigma$  represents the sum of

$X_i$  represents the  $i$ th observation of variable X

$\mu_X$  represents the mean of variable X

$Y_i$  represents the  $i$ th observation of variable Y

$\mu_Y$  represents the mean of variable Y

n represents the number of observations

In other words, to calculate the covariance between X and Y, we take the product of the deviations of each observation from their respective means, sum these products across all observations, and then divide by the number of observations minus one.

This formula gives us a measure of how much two variables vary together. If the covariance is positive, it means that when one variable is above its mean, the other tends to be above its mean as well. If the covariance is negative, it means that when one variable is above its mean, the other tends to be below its mean. If the covariance is zero, it means that there is no linear relationship between the two variables. Let's use the formula for covariance to calculate the covariance between two variables X and Y.

Suppose we have the following data for X and Y:

X: 10, 20, 30, 40, 50

Y: 15, 25, 35, 45, 55

To calculate the covariance between X and Y, we first need to find the means of both variables. The mean of X is  $(10+20+30+40+50)/5 = 30$ , and the mean of Y is  $(15+25+35+45+55)/5 = 35$ .

Next, we calculate the deviations of each variable from their respective means. For X, the deviations are (-20, -10, 0, 10, 20) and for Y, the deviations are (-20, -10, 0, 10, 20).

We then multiply the deviations of each pair of observations together, and sum these products across all observations. Using the formula, we get:

$$\begin{aligned}\text{cov}(X, Y) &= (\sum[(X_i - \mu_X)(Y_i - \mu_Y)]) / (n - 1) \\ &= [(-20)(15-35) + (-10)(25-35) + (0)(35-35) + (10)(45-35) + (20)(55-35)] / (5-1) \\ &= (-400 - 100 + 0 + 100 + 400) / 4 \\ &= 250\end{aligned}$$

So the covariance between X and Y is equal to 250. This tells us that there is a positive relationship between X and Y; in other words, when X is above its mean, Y tends to be above its mean as well.

Let's say we have two variables, X and Y, with the following values:

X: 1, 2, 3, 4, 5

Y: 2, 4, 6, 8, 10

To calculate covariance, we first need to find the means of both X and Y. The mean of X is  $(1+2+3+4+5)/5 = 3$ , and the mean of Y is  $(2+4+6+8+10)/5 = 6$ .

Next, we calculate the deviations of each variable from their respective means. For X, the deviations are (-2, -1, 0, 1, 2) and for Y, the deviations are (-4, -2, 0, 2, 4).

We then multiply the deviations of each pair of observations together, and take the average of these products. In this case, we get:

$$(-2 * -4) + (-1 * -2) + (0 * 0) + (1 * 2) + (2 * 4) = 20$$

Dividing by the number of observations minus one (which is  $5-1=4$ ), we get the covariance:

$$\text{cov}(X,Y) = 20 / 4 = 5$$

So the covariance between X and Y is 5.

Suppose we have a dataset with two variables, x and y, and three observations:

Observation 1: x = 2, y = 4

Observation 2: x = 3, y = 5

Observation 3: x = 4, y = 6

We want to calculate the Mahalanobis distance between observation 1 and observation 2. To do this, we first need to calculate the covariance matrix of the dataset:

Covariance matrix:

	x	y
---	---	---
x	1/2	1/2
y	1/2	1/2

Next, we need to calculate the inverse of the covariance matrix:

Inverse covariance matrix:

	x	y
---	---	---
x'	2	-2
y'	-2	2

Now we can calculate the Mahalanobis distance using the formula:

$$D^2 = (x_1 - x_2)' * S^{-1} * (x_1 - x_2)$$

where D is the Mahalanobis distance, x1 is the vector of variables for observation 1 ( $x=2, y=4$ ), x2 is the vector of variables for observation 2 ( $x=3, y=5$ ), and  $S^{-1}$  is the inverse covariance matrix.

Plugging in our values:

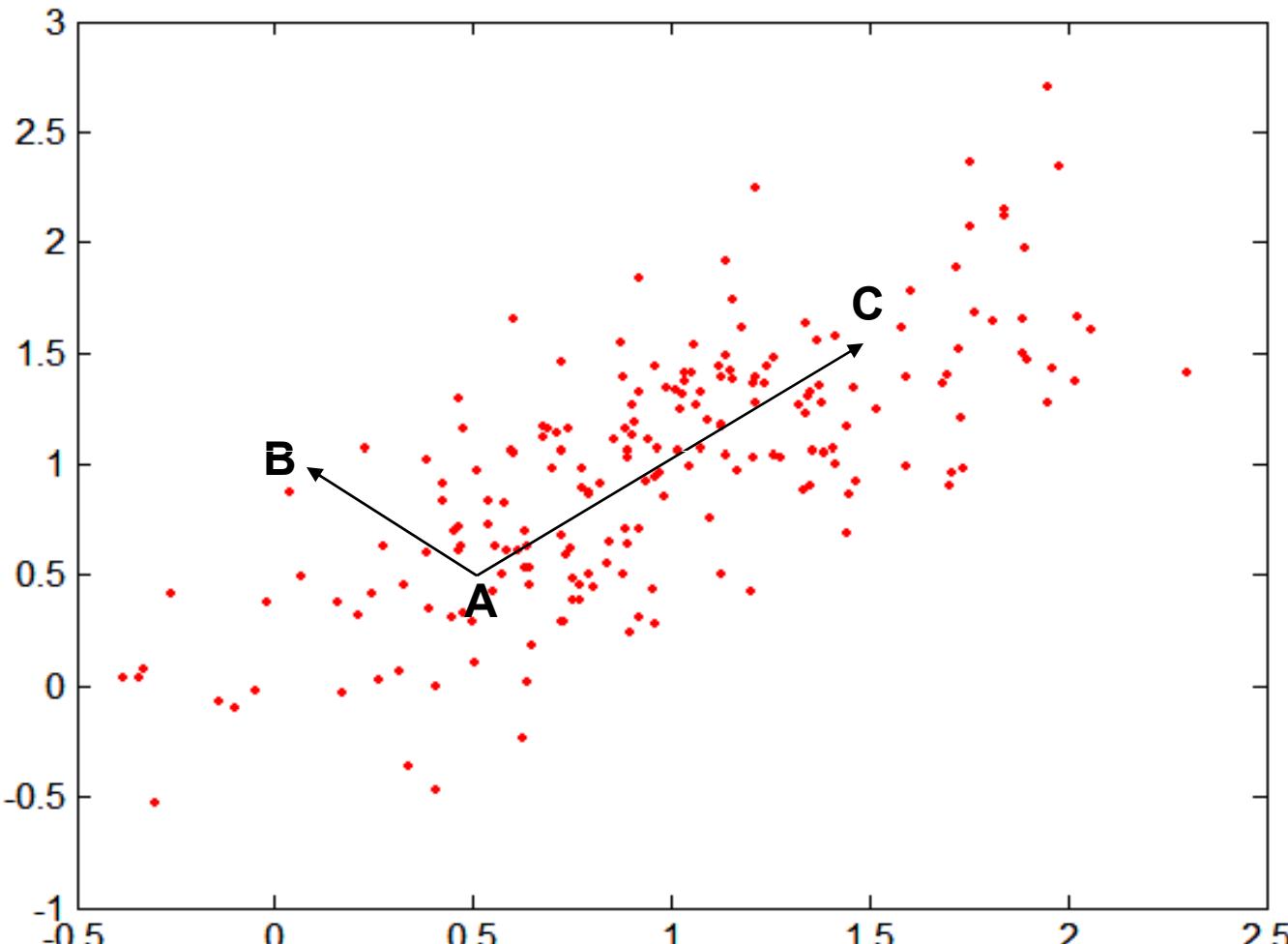
$$\begin{aligned} D^2 &= (([2,4] - [3,5])' * [[2,-2],[-2,2]] * ([2,4] - [3,5])) \\ &= (([-1,-1])' * [[-4],[-4]] * ([-1,-1])) \\ &= (-8) \end{aligned}$$

Taking the square root of this value gives us our final result:

$$D = \sqrt{-8} = 2.83$$

Therefore, the Mahalanobis distance between observation 1 and observation 2 is 2.83.

# Mahalanobis Distance



Covariance Matrix:

$$\Sigma = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}$$

A: (0.5, 0.5)

B: (0, 1)

C: (1.5, 1.5)

$\text{Mahal}(A, B) = 5$

$\text{Mahal}(A, C) = 4$

Mahalanobis Distance is a measure that takes into account the covariance between variables when calculating distance. It is often used in machine learning and pattern recognition applications where the variables are correlated with each other.

Let's say we have a dataset of 5 observations, each with 3 variables:

$x_1 = [1, 2, 3]$

$x_2 = [4, 5, 6]$

$x_3 = [7, 8, 9]$

$x_4 = [10, 11, 12]$

$x_5 = [13, 14, 15]$

To calculate Mahalanobis Distance between any two observations, we need to calculate the inverse of the covariance matrix of the data. The covariance matrix is a square matrix that shows the variances of each variable along the diagonal and the covariances between each pair of variables off-diagonal.

To calculate the covariance matrix for this dataset, we first need to find the means of each variable:

$\text{mean}_x_1 = (1 + 4 + 7 + 10 + 13) / 5 = 7$

$\text{mean}_x_2 = (2 + 5 + 8 + 11 + 14) / 5 = 8$

$\text{mean}_x_3 = (3 + 6 + 9 + 12 + 15) / 5 = 9$

Next, we calculate the deviations of each observation from their respective means:

$\text{dev}_x_1 = [1-7, 2-8, 3-9] = [-6, -6, -6]$

$\text{dev}_x_2 = [4-7, 5-8, 6-9] = [-3, -3, -3]$

$\text{dev}_x_3 = [7-7, 8-8, 9-9] = [0, 0, 0]$

$\text{dev}_x_4 = [10-7, 11-8, 12-9] = [3, 3, 3]$

$\text{dev}_x_5 = [13-7, 14-8, 15-9] = [6, 6, 6]$

We then calculate the covariance matrix by taking the dot product of the deviation matrix with its transpose, and dividing by n-1:

covariance\_matrix =

```
array([[13.33333333, 13.33333333, 13.33333333],
       [13.33333333, 13.33333333, 13.33333333],
       [13.33333333, 13.33333333, 13.33333333]])
```

Next, we calculate the inverse of the covariance matrix using numpy:

```
import numpy as np
```

```
cov_inv = np.linalg.inv(covariance_matrix)
```

Now, let's say we want to calculate the Mahalanobis Distance between observations  $x_1$  and  $x_2$ . We first calculate the difference vector between the two observations:

```
diff_vector = np.array(x1) - np.array(x2)
```

Then, we calculate the Mahalanobis Distance using the formula:

```
mahalanobis_distance = np.sqrt(np.dot(np.dot(diff_vector, cov_inv), diff_vector.T))
```

Plugging in the values, we get:

```
mahalanobis_distance = np.sqrt(np.dot(np.dot([-3, -3, -3], cov_inv), [-3, -3, -3].T))
                           = np.sqrt(np.dot([-1.5, -1.5, -1.5], [-3, -3, -3]))
                           = np.sqrt(13.5)
                           = 3.674
```

So the Mahalanobis Distance between observations  $x_1$  and  $x_2$  is approximately equal to 3.674.



# Common Properties of a Distance

---

- Distances, such as the Euclidean distance, have some well known properties.
  1.  $d(x, y) \geq 0$  for all  $x$  and  $y$  and  $d(x, y) = 0$  if and only if  $x = y$ .
  2.  $d(x, y) = d(y, x)$  for all  $x$  and  $y$ . (Symmetry)
  3.  $d(x, z) \leq d(x, y) + d(y, z)$  for all points  $x, y$ , and  $z$ .  
(Triangle Inequality)

where  $d(x, y)$  is the distance (dissimilarity) between points (data objects),  $x$  and  $y$ .

- A distance that satisfies these properties is a metric

# Common Properties of a Similarity

---

- Similarities, also have some well known properties.
  1.  $s(x, y) = 1$  (or maximum similarity) only if  $x = y$ .  
(does not always hold, e.g., cosine)
  2.  $s(x, y) = s(y, x)$  for all  $x$  and  $y$ . (Symmetry)

where  $s(x, y)$  is the similarity between points (data objects),  $x$  and  $y$ .

# Similarity Between Binary Vectors

---

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

$f_{01}$  = the number of attributes where  $x$  was 0 and  $y$  was 1

$f_{10}$  = the number of attributes where  $x$  was 1 and  $y$  was 0

$f_{00}$  = the number of attributes where  $x$  was 0 and  $y$  was 0

$f_{11}$  = the number of attributes where  $x$  was 1 and  $y$  was 1

- Simple Matching and Jaccard Coefficients

SMC = number of matches / number of attributes

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

J = number of 11 matches / number of non-zero attributes

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

## SMC versus Jaccard: Example

---

**x** = 1 0 0 0 0 0 0 0 0 0

**y** = 0 0 0 0 0 0 1 0 0 1

$f_{01} = 2$  (the number of attributes where **x** was 0 and **y** was 1)

$f_{10} = 1$  (the number of attributes where **x** was 1 and **y** was 0)

$f_{00} = 7$  (the number of attributes where **x** was 0 and **y** was 0)

$f_{11} = 0$  (the number of attributes where **x** was 1 and **y** was 1)

$$\begin{aligned}\text{SMC} &= (f_{11} + f_{00}) / (f_{01} + f_{10} + f_{11} + f_{00}) \\ &= (0+7) / (2+1+0+7) = 0.7\end{aligned}$$

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

# Cosine Similarity

دوتا وکتور یا بردار یا  
اجکت داریم

ضرب داخلی

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

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

ضرب اندازه‌ی وکتورها  
برای نرمال کردن  
صورت که بدست اومد

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

- Example:

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

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

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

محاسبه‌ی نرم  
این وکتورها

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

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

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

چه زمانی بیشترین شباهت بدست می‌آید?  
وقتی که وکتورها دقیقاً مثل هم باشند  
که جواب  $\cos(\mathbf{d}_1, \mathbf{d}_2) = 1$  می‌شود

دوتا اجکت دی ۱ و دی  
۲ چقدر دارن شبیه هم  
رفتار می‌کنند؟

1) Calculate the cosine similarity between vectors [2, 3, 4] and [5, 6, 7]:

$$\text{cosine similarity} = (2*5 + 3*6 + 4*7) / (\sqrt{2^2 + 3^2 + 4^2} * \sqrt{5^2 + 6^2 + 7^2})$$

cosine similarity = 0.994

2) Calculate the cosine similarity between vectors [1, 0, -1] and [-1, 0, 1]:

$$\text{cosine similarity} = (1*(-1) + 0*0 + (-1)*1) / (\sqrt{1^2 + 0^2 + (-1)^2} * \sqrt{(-1)^2 + 0^2 + 1^2})$$

cosine similarity = -1

3) Calculate the cosine similarity between vectors [4, 5, 6] and [7, 8, 9]:

$$\text{cosine similarity} = (4*7 + 5*8 + 6*9) / (\sqrt{4^2 + 5^2 + 6^2} * \sqrt{7^2 + 8^2 + 9^2})$$

cosine similarity = 0.997

4) Calculate the cosine similarity between vectors [0, 1, 0] and [0, -1, 0]:

$$\text{cosine similarity} = (0*0 + 1*(-1) + 0*0) / (\sqrt{0^2 + 1^2 + 0^2} * \sqrt{0^2 + (-1)^2 + 0^2})$$

cosine similarity = -1

5) Calculate the cosine similarity between vectors [3, -4] and [-6, 8]:

$$\text{cosine similarity} = (3*(-6) + (-4)*8) / (\sqrt{3^2 + (-4)^2} * \sqrt{(-6)^2 + 8^2})$$

cosine similarity = -1

Suppose we have the following two-dimensional data set:

	$A_1$	$A_2$
$x_1$	1.5	1.7
$x_2$	2	1.9
$x_3$	1.6	1.8
$x_4$	1.2	1.5
$x_5$	1.5	1.0

- (a) Consider the data as two-dimensional data points. Given a new data point,  $\mathbf{x} = (1.4, 1.6)$  as a query, rank the database points based on similarity with the query using (1) Euclidean distance (Equation 7.5), and (2) cosine similarity (Equation 7.16).

The Euclidean distance of two  $n$ -dimensional vectors,  $\mathbf{x}$  and  $\mathbf{y}$ , is defined as:  $\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$ . The cosine similarity of  $\mathbf{x}$  and  $\mathbf{y}$  is defined as:  $\frac{\mathbf{x}^t \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$ , where  $\mathbf{x}^t$  is a transposition of vector  $\mathbf{x}$ ,  $\|\mathbf{x}\|$  is the Euclidean norm of vector  $\mathbf{x}$ ,<sup>1</sup> and  $\|\mathbf{y}\|$  is the Euclidean norm of vector  $\mathbf{y}$ . Using these definitions we obtain the distance from each point to the query point.

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
Euclidean distance	0.14	0.67	0.28	0.22	0.61
Cosine similarity	0.9999	0.9957	0.9999	0.9990	0.9653

Based on the Euclidean distance, the ranked order is  $x_1, x_4, x_3, x_5, x_2$ . Based on the cosine similarity, the order is  $x_1, x_3, x_4, x_2, x_5$ .

---

<sup>1</sup>The Euclidean normal of vector  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  is defined as  $\sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$ .

- | **Cosine similarity between two term-frequency vectors.** Suppose that  $\mathbf{x}$  and  $\mathbf{y}$  are the first two term-frequency vectors in Table 2.5. That is,  $\mathbf{x} = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)$  and  $\mathbf{y} = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)$ . How similar are  $\mathbf{x}$  and  $\mathbf{y}$ ? Using Eq. (2.23) to compute the cosine similarity between the two vectors, we get:

$$\begin{aligned} \mathbf{x}^t \cdot \mathbf{y} &= 5 \times 3 + 0 \times 0 + 3 \times 2 + 0 \times 0 + 2 \times 1 + 0 \times 1 + 0 \times 0 + 2 \times 1 \\ &\quad + 0 \times 0 + 0 \times 1 = 25 \end{aligned}$$

$$\|\mathbf{x}\| = \sqrt{5^2 + 0^2 + 3^2 + 0^2 + 2^2 + 0^2 + 0^2 + 2^2 + 0^2} = 6.48$$

$$\|\mathbf{y}\| = \sqrt{3^2 + 0^2 + 2^2 + 0^2 + 1^2 + 1^2 + 0^2 + 1^2 + 0^2} = 4.12$$

$$sim(\mathbf{x}, \mathbf{y}) = 0.94$$

Therefore, if we were using the cosine similarity measure to compare these documents, they would be considered quite similar. ■

# Correlation measures the linear relationship between objects

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

where we are using the following standard statistical notation and definitions

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

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

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

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

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

The correlation coefficient is a measure of the linear relationship between two variables. It ranges from -1 to 1, where -1 indicates a perfect negative linear relationship, 0 indicates no linear relationship, and 1 indicates a perfect positive linear relationship.

Let's consider an example where we have two variables, X and Y, with the following values:

$$X = [1, 2, 3, 4, 5]$$

$$Y = [3, 5, 7, 9, 11]$$

To calculate the correlation coefficient between X and Y, we first need to calculate the mean and standard deviation of each variable.

The mean of X is:

$$\text{mean}(X) = (1 + 2 + 3 + 4 + 5) / 5 = 3$$

The mean of Y is:

$$\text{mean}(Y) = (3 + 5 + 7 + 9 + 11) / 5 = 7$$

The standard deviation of X is:

$$\text{std}(X) = \sqrt{((1-3)^2 + (2-3)^2 + (3-3)^2 + (4-3)^2 + (5-3)^2) / (5-1)} = 1.5811$$

The standard deviation of Y is:

$$\text{std}(Y) = \sqrt{((3-7)^2 + (5-7)^2 + (7-7)^2 + (9-7)^2 + (11-7)^2) / (5-1)} = 2.8284$$

Next, we can calculate the covariance between X and Y:

$$\text{cov}(X, Y) = ((1-3)*(3-7) + (2-3)*(5-7) + (3-3)*(7-7) + (4-3)*(9-7) + (5-3)*(11-7)) / (5-1) = 5$$

Finally, we can calculate the correlation coefficient between X and Y:

$$\text{corr}(X, Y) = \text{cov}(X, Y) / (\text{std}(X) * \text{std}(Y)) = 5 / (1.5811 * 2.8284) = 0.8839$$

Therefore, the correlation coefficient between X and Y is 0.8839, indicating a strong positive linear relationship between the two variables.

Suppose we have two variables, X and Y, with the following values:

$$X = [10, 20, 30, 40, 50]$$

$$Y = [5, 15, 25, 35, 45]$$

The steps to calculate the correlation coefficient are similar to the previous example. First, we calculate the mean and standard deviation of each variable:

$$\text{mean}(X) = (10 + 20 + 30 + 40 + 50) / 5 = 30$$

$$\text{std}(X) = \sqrt{((10-30)^2 + (20-30)^2 + (30-30)^2 + (40-30)^2 + (50-30)^2) / (5-1)} = 15.8114$$

$$\text{mean}(Y) = (5 + 15 + 25 + 35 + 45) / 5 = 25$$

$$\text{std}(Y) = \sqrt{((5-25)^2 + (15-25)^2 + (25-25)^2 + (35-25)^2 + (45-25)^2) / (5-1)} = 15.8114$$

Next, we calculate the covariance between X and Y:

$$\text{cov}(X,Y) = ((10-30)*(5-25) + (20-30)*(15-25) + (30-30)*(25-25) + (40-30)*(35-25) + (50-30)*(45-25)) / (5-1) = 500$$

Finally, we calculate the correlation coefficient between X and Y:

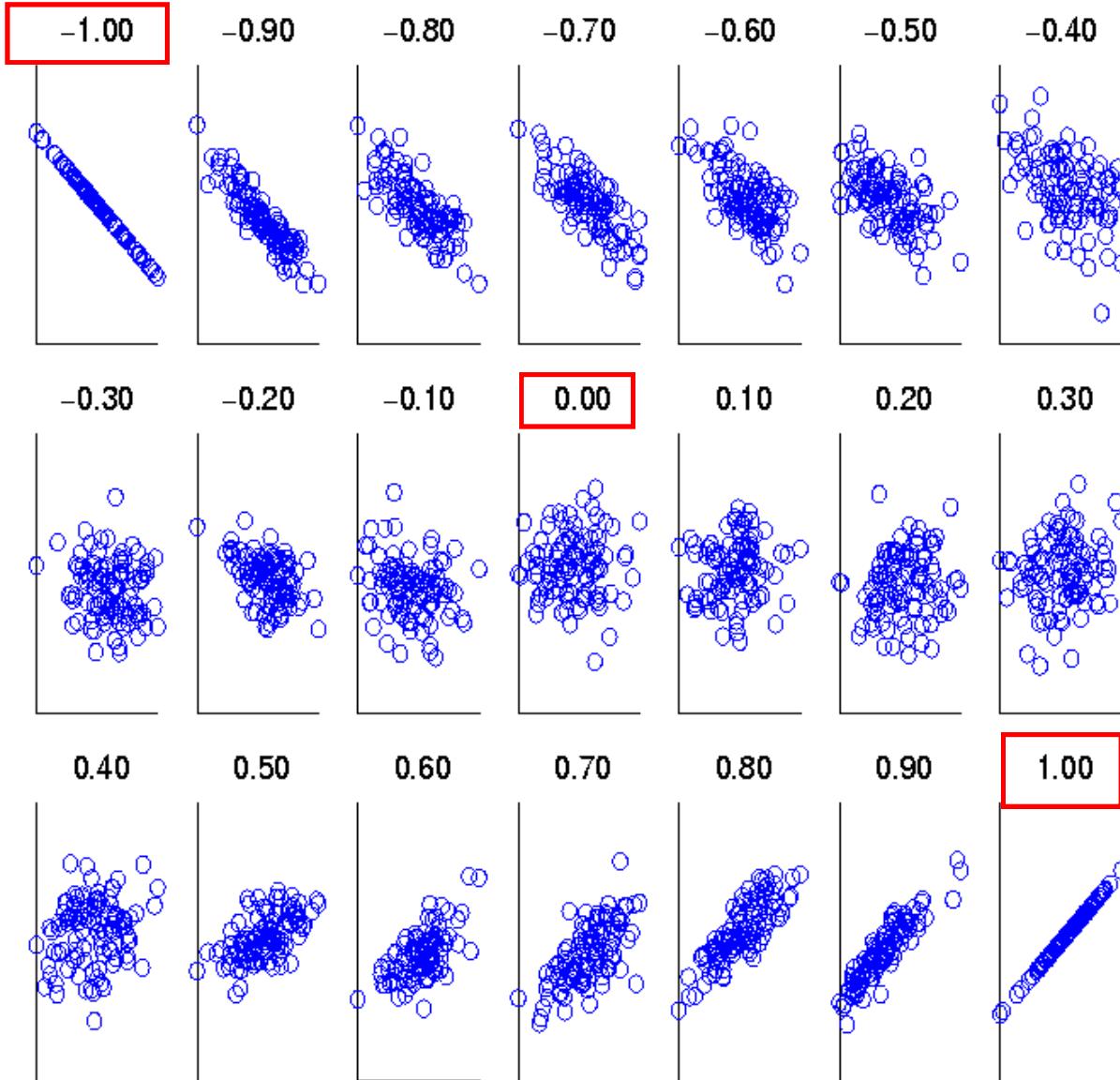
$$\text{corr}(X,Y) = \text{cov}(X,Y) / (\text{std}(X) * \text{std}(Y)) = 500 / (15.8114 * 15.8114) = 0.9439$$

Therefore, the correlation coefficient between X and Y is 0.9439, indicating a strong positive linear relationship between the two variables.

Drawbacks of correlation:

1. Correlation measures only linear relationships: Correlation measures only the linear relationship between two variables and does not capture nonlinear relationships. In machine learning, nonlinear relationships are often present in the data, and using correlation as a measure of distance can lead to inaccurate results.
2. Correlation is sensitive to outliers: Correlation is sensitive to outliers, which can have a significant impact on the correlation coefficient. In machine learning, outliers are common in real-world datasets, and using correlation as a measure of distance can lead to inaccurate results.
3. Correlation does not account for differences in scale: Correlation does not account for differences in the scale of the variables being compared. In machine learning, variables often have different scales, and using correlation as a measure of distance can lead to inaccurate results.
4. Correlation does not capture complex relationships: Correlation measures only the linear relationship between two variables and does not capture complex relationships, such as interactions between variables. In machine learning, complex relationships between variables are often present in the data, and using correlation as a measure of distance can lead to inaccurate results.

# Visually Evaluating Correlation



Scatter plots  
showing the  
similarity from  
–1 to 1.

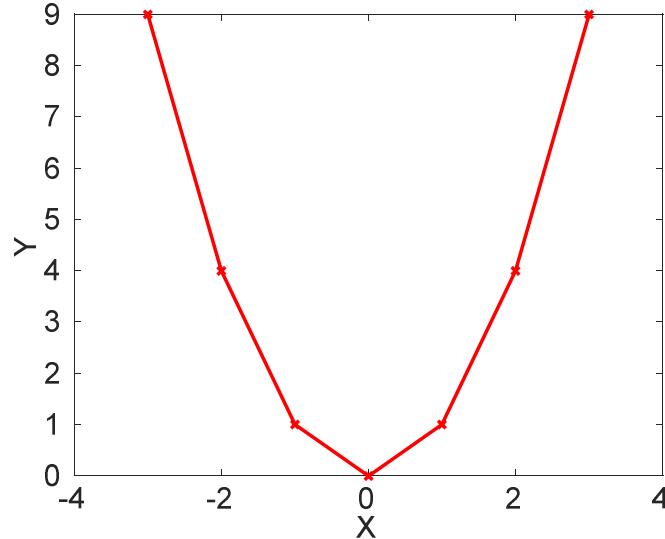
مقایسه‌ی خود اتربیوت  
با خودش

نمیتواند روابط غیر خطی را تشخیص  
بده  
رابطه ای که بین اtribut ها هست  
غیر خطی است  
خطی یعنی اگه ایکس را دو برابر  
کردیم وای متناظرش هم دو برابر پشه

# Drawback of Correlation

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

$$y_i = x_i^2$$



- $\text{mean}(\mathbf{x}) = 0, \text{mean}(\mathbf{y}) = 4$
- $\text{std}(\mathbf{x}) = 2.16, \text{std}(\mathbf{y}) = 3.74$

$$\bullet \text{corr} = \frac{(-3)(5)+(-2)(0)+(-1)(-3)+(0)(-4)+(1)(-3)+(2)(0)+3(5)}{(6 * 2.16 * 3.74)}$$

ایکس و یای باهم ارتباط  
دارند ولی داره میگه  
ارتباطشون صفر است  
بنی تنوانت ارتباط  
اینها را تشخیص بد

$$= 0$$

$$y_i - \text{mean}(y) \\ 9 - 5 = 4$$

$$x_i - \text{mean}(x) \\ -3 - 0 = -3$$

$$n=7 \\ n-1 = 6$$

# Correlation vs Cosine vs Euclidean Distance

- Compare the three proximity measures according to their behavior under variable transformation

- scaling: multiplication by a value
- translation: adding a constant

آیا رفتارشون با تغییر  
اتribut ها تغییر میکنه یا  
نه؟

ثابت بودن جواب در اثر  
اسکلیل کردن داده ها یعنی  
در به عددی ضرب کنیم  
مثل

Property	Cosine	Correlation	Euclidean Distance
Invariant to scaling (multiplication)	Yes	Yes	No
Invariant to translation (addition)	No	Yes	No

- Consider the example

- $x = (1, 2, 4, 3, 0, 0, 0)$ ,  $y = (1, 2, 3, 4, 0, 0, 0)$
- $y_s = y * 2$  (scaled version of y),  $y_t = y + 5$  (translated version)

Measure	$(x, y)$	$(x, y_s)$	$(x, y_t)$
Cosine	0.9667	0.9667	0.7940
Correlation	0.9429	0.9429	0.9429
Euclidean Distance	1.4142	5.8310	14.2127

در هر شرایطی ثابت است  
و تغییر نمیکند

# Correlation vs cosine vs Euclidean distance

---

- Choice of the right proximity measure depends on the domain
- What is the correct choice of proximity measure for the following situations?
  - Comparing documents using the frequencies of words
    - ◆ Documents are considered similar if the word frequencies are similar
  - Comparing the temperature in Celsius of two locations
    - ◆ Two locations are considered similar if the temperatures are similar in magnitude
  - Comparing two time series of temperature measured in Celsius
    - ◆ Two time series are considered similar if their “shape” is similar, i.e., they vary in the same way over time, achieving minimums and maximums at similar times, etc.

اندازه بزرگی

وقتی میخاییم یه مدل بسازیم برآمون مهم میشه از کدام اتریبیوت ها باید استفاده کنیم کدام هارا انتخاب کنیم و کدام هارا کنار بگذاریم؟ مثلا ممکنه هزارتا اتریبیوت از یه اجکت داشته باشیم ولی نمیدونیم کدامش را انتخاب کنیم؟ همش را نمیشه استفاده کرد اگه همش را بخاییم استفاده کنیم اصلا مدل سازی نخواهیم داشت

- Correlation: Correlation measures the linear relationship between two variables. It ranges from -1 to 1, where -1 indicates a perfect negative linear relationship, 0 indicates no linear relationship, and 1 indicates a perfect positive linear relationship. Correlation is commonly used in statistical analysis to measure the strength and direction of the relationship between two variables.
- Cosine similarity: Cosine similarity measures the cosine of the angle between two vectors. It ranges from -1 to 1, where -1 indicates that the two vectors are diametrically opposed, 0 indicates that the two vectors are orthogonal, and 1 indicates that the two vectors are identical. Cosine similarity is commonly used in information retrieval and text processing to measure the similarity between two documents or two vectors of word frequencies.
- Euclidean distance: Euclidean distance measures the distance between two points in n-dimensional space. It is calculated as the square root of the sum of the squared differences between the corresponding coordinates of the two points. Euclidean distance is commonly used in machine learning and data mining to measure the similarity between two data points or to cluster data points based on their similarity.

# Comparison of Proximity Measures

---

- Domain of application
  - Similarity measures tend to be specific to the type of attribute and data
  - Record data, images, graphs, sequences, 3D-protein structure, etc. tend to have different measures
- However, one can talk about various properties that you would like a proximity measure to have
  - Symmetry is a common one
  - Tolerance to noise and outliers is another
  - Ability to find more types of patterns?
  - Many others possible
- The measure must be applicable to the data and produce results that agree with domain knowledge

ابزارهای اندازه‌گیری براساس تئوری اطلاعات  
باید بینیم چقدر عدم قطعیت توى اطلاعاتش هست؟ چقدر اطلاعاتش  
پراکنده است؟ احتمال رخدادن داده ها از نظر قد دانشجویان چقدر؟

# Information Based Measures

- Information theory is a well-developed and fundamental discipline with broad applications
- Some similarity measures are based on information theory
  - Mutual information in various versions
  - Maximal Information Coefficient (MIC) and related measures
  - General and can handle non-linear relationships
  - Can be complicated and time intensive to compute

پس یه جایی باید اتریبیوت هامون را باهم مقایسه کنیم  
اگه بخایم دوتا اتریبیوتی که هیچ ربطی به هم ندارن را مقایسه کنیم چطوری باید اینکارو کنیم؟

اگه یه سکه بندازیم که  
همش رو بیاد هیچ  
اطلاعاتی به ما نمیده  
چون هیچ عدم قطعیتی  
توش نیست

# Information and Probability

- Information relates to possible outcomes of an event
  - transmission of a message, flip of a coin, or measurement of a piece of data
- The more certain an outcome, the less information that it contains and vice-versa
  - For example, if a coin has two heads, then an outcome of heads provides no information
  - More quantitatively, the information is related to the probability of an outcome
    - ◆ The smaller the probability of an outcome, the more information it provides and vice-versa
  - Entropy is the commonly used measure

پدیده هایی که احتمال رخدادشون  
کمتر باشه اطلاعاتی که میدن بیشتر  
است

اتribut ها را در یک فضای مشترک مثل فضای اطلاعات میبریم و بعد مقایشون میکنیم چون  
مستقیم نمیتونیم مقایشون کنیم  
فضای اطلاعاتی: information scale  
اطلاعات یک رابطه ی نزدیک با احتمال و عدم قطعیت داره پدیده ای که همیشه قطعی است و  
عدم قطعیت نداره هیچ اطلاعاتی به ما نمیده اگه سکه طوری باشه که ما نتوانیم پیشینی کنیم که  
رو بیاد یا پشت در دفعه ی بعدی میگیریم یه اطلاعاتی توش هست ولی اگه قرار باشه سکه همش  
رو بیاد اطلاعاتی نمیده نتیجه حاصل از انداختن سکه



# Entropy

ارتباط و شباهت بین  
اتribut ها را میسنجیم  
مثل ارتباط بین وزن و قد

ثلا تاس شش تا مقدار داره و هر کدام  
به احتمالی داره که ببیاد

## For

- a variable (event),  $X$ ,
- with  $n$  possible values (outcomes),  $x_1, x_2 \dots, x_n$
- each outcome having probability,  $p_1, p_2 \dots, p_n$
- the entropy of  $X$ ,  $H(X)$ , is given by

$$H(X) = - \sum_{i=1}^n p_i \log_2 p_i$$

چقدر اطلاعات نوی  
متغیر تصادفی مون  
هست؟

باید احتمال متغیر تصادفی ها را در بیاریم  
بنی به ازای هر مقداری که  
متغیر تصادفی مون پیدا میکنه باید احتمالش  
را حساب کنیم  
اگه متغیر تصادفی مون انداختن یک تاس  
باشه پس ۶ تا حالت داره متغیر مون پس  
شش تا احتمال باید بدست بیاریم

اگه احتمال همه مقادیر یکسان باشه ینی هر کدام  
 $1/n$   
باشد  
انتروپی حاصل میشه یک  
انتروپی: چندتا بیت لازم داریم تا اطلاعات را  
ذخیره کنیم؟

## Entropy is between 0 and $\log_2 n$ and is measured in bits

- Thus, entropy is a measure of how many bits it takes to represent an observation of  $X$  on average

اگه انتروپی صفر بشه  
بنی احتمال یک مقدار از  
متغیر تصادفی یک است  
و بقیه صفر است  
بنی اطلاعاتی بمون نمیده

ماکس مقدار یه متغیر را  
اگه ازش لگاریتم در  
مبناي دو بگيريم تعداد  
بيت لازم برای ذخیره  
كردنش در میاد

# Entropy Examples

---

---

- For a coin with probability  $p$  of heads and probability  $q = 1 - p$  of tails
- What is the entropy of a fair four-sided die?

# Entropy Examples

- For a coin with probability  $p$  of heads and probability  $q = 1 - p$  of tails

$$H = -p \log_2 p - q \log_2 q$$

- For  $p=0.5, q=0.5$  (fair coin)  $H=1$
- For  $p=1$  or  $q=1$ ,  $H=0$

احتمال رخدادن مقادیر مختلف سکه یکسان بود و هر کدام یک دوم بود پس انتروپی میشه یک

- What is the entropy of a fair four-sided die?

# Entropy for Sample Data: Example

Hair Color	Count	$p$	$-p \log_2 p$
Black	75	0.75	0.3113
Brown	15	0.15	0.4105
Blond	5	0.05	0.2161
Red	0	0.00	0
Other	5	0.05	0.2161
Total	100	1.0	1.1540

سوال:  
انتروپی رنگ  
مو چیست؟

Maximum entropy is  $\log_2 5 = 2.3219$

انتروپی حد پایین داره ولی حد  
بالاش معلوم نیست  
ولی راجع به پرائندگی اون  
متغیر تصادفی یه سری اطلاعات  
بمون میده

# Entropy for Sample Data

---

---

- Suppose we have

- a number of observations ( $m$ ) of some attribute,  $X$ ,  
e.g., the hair color of students in the class,
- where there are  $n$  different possible values
- And the number of observation in the  $i^{\text{th}}$  category is  $m_i$
- Then, for this sample

$$H(X) = - \sum_{i=1}^n \frac{m_i}{m} \log_2 \frac{m_i}{m}$$

- For continuous data, the calculation is harder

Suppose we have a dataset with 10 samples, where each sample belongs to one of two classes, A or B. The number of samples in each class is as follows:

Class A: 6 samples

Class B: 4 samples

To calculate the entropy of this dataset, we first need to calculate the proportion of samples in each class:

Proportion of class A =  $6 / 10 = 0.6$

Proportion of class B =  $4 / 10 = 0.4$

Next, we can use the entropy formula to calculate the entropy of the dataset:

Entropy = - (Proportion of class A \* log2(Proportion of class A) + Proportion of class B \* log2(Proportion of class B))

Entropy = - (0.6 \* log2(0.6) + 0.4 \* log2(0.4))

Entropy = 0.971

Therefore, the entropy of the dataset is 0.971.

Suppose we have a dataset with 20 samples, where each sample belongs to one of three classes, A, B, or C. The number of samples in each class is as follows:

Class A: 8 samples

Class B: 6 samples

Class C: 6 samples

To calculate the entropy of this dataset, we first need to calculate the proportion of samples in each class:

Proportion of class A =  $8 / 20 = 0.4$

Proportion of class B =  $6 / 20 = 0.3$

Proportion of class C =  $6 / 20 = 0.3$

Next, we can use the entropy formula to calculate the entropy of the dataset:

Entropy = - (Proportion of class A \* log2(Proportion of class A) + Proportion of class B \* log2(Proportion of class B) + Proportion of class C \* log2(Proportion of class C))

Entropy = - (0.4 \* log2(0.4) + 0.3 \* log2(0.3) + 0.3 \* log2(0.3))

Entropy = 1.576

# Mutual Information

راجع به ارتباط دو تا  
متغیر اطلاعات کسب  
کنیم با رویدهای  
تئوری اطلاعاتی

- Information one variable provides about another

Formally,  $I(X, Y) = H(X) + H(Y) - H(X, Y)$ , where

$H(X, Y)$  is the joint entropy of  $X$  and  $Y$ ,

عددی که بدست میاد  
میگه چقدر این دو تا  
متغیر به هم ربط دارند

مثلًا قد و وزن را داریم

$$H(X, Y) = - \sum_i \sum_j p_{ij} \log_2 p_{ij}$$

Where  $p_{ij}$  is the probability that the  $i^{\text{th}}$  value of  $X$  and the  $j^{\text{th}}$  value of  $Y$  occur together

- For discrete variables, this is easy to compute
- Maximum mutual information for discrete variables is  $\log_2(\min(n_X, n_Y))$ , where  $n_X$  ( $n_Y$ ) is the number of values of  $X$  ( $Y$ )

# Mutual Information Example

<b>Student Status</b>	<b>Count</b>	<b><math>p</math></b>	<b><math>-p \log_2 p</math></b>
Undergrad	45	0.45	0.5184
Grad	55	0.55	0.4744
Total	100	1.00	0.9928

<b>Student Status</b>	<b>Grade</b>	<b>Count</b>	<b><math>p</math></b>	<b><math>-p \log_2 p</math></b>
Undergrad	A	5	0.05	0.2161
Undergrad	B	30	0.30	0.5211
Undergrad	C	10	0.10	0.3322
Grad	A	30	0.30	0.5211
Grad	B	20	0.20	0.4644
Grad	C	5	0.05	0.2161
<b>Total</b>		<b>100</b>	<b>1.00</b>	<b>2.2710</b>

<b>Grade</b>	<b>Count</b>	<b><math>p</math></b>	<b><math>-p \log_2 p</math></b>
A	35	0.35	0.5301
B	50	0.50	0.5000
C	15	0.15	0.4105
Total	100	1.00	1.4406

Mutual information of Student Status and Grade =  $0.9928 + 1.4406 - 2.2710 = 0.1624$

$$I(X; Y) = \sum_{x,y} P_{XY}(x, y) \log \frac{P_{XY}(x, y)}{P_X(x)P_Y(y)} = E_{P_{XY}} \log \frac{P_{XY}}{P_X P_Y} .$$

Mutual information is a measure of the amount of information that two variables share. It is often used in machine learning and data analysis to determine the relationship between two variables.

Here are a few numeric examples:

1. Suppose we have two binary variables, X and Y, where X can take on the values 0 or 1 and Y can also take on the values 0 or 1. If we observe that X and Y are perfectly correlated (i.e., whenever X=1, Y=1 and whenever X=0, Y=0), then the mutual information between X and Y would be 1 bit.
2. Let's say we have two continuous variables, A and B, where A represents a person's age (in years) and B represents their income (in thousands of dollars). If we find that there is a strong positive correlation between A and B (i.e., as age increases, so does income), then the mutual information between A and B would be relatively high.
3. Consider two categorical variables C and D, where C represents a person's gender (either male or female) and D represents their favorite color (red, blue, or green). If we observe that males tend to prefer blue while females tend to prefer red, then the mutual information between C and D would be non-zero but relatively low.

Suppose we have two binary variables X and Y with the following data:

	Y=0	Y=1
---	-----	-----
X=0	10	30
X=1	20	40

To calculate the mutual information between X and Y, we can use the same formula as in Example 1:

$$I(X;Y) = \sum \sum p(x,y) \log_2(p(x,y) / (p(x) * p(y)))$$

First, we need to calculate the marginal probabilities:

$$\begin{aligned} p(X=0) &= 10 + 30 = 40 / 100 = 0.4 \\ p(X=1) &= 20 + 40 = 60 / 100 = 0.6 \\ p(Y=0) &= 10 + 20 = 30 / 100 = 0.3 \\ p(Y=1) &= 30 + 40 = 70 / 100 = 0.7 \end{aligned}$$

Next, we can calculate the joint probabilities:

$$\begin{aligned} p(X=0, Y=0) &= 10 / 100 = 0.1 \\ p(X=0, Y=1) &= 30 / 100 = 0.3 \\ p(X=1, Y=0) &= 20 / 100 = 0.2 \\ p(X=1, Y=1) &= 40 / 100 = 0.4 \end{aligned}$$

Now we can calculate the mutual information:

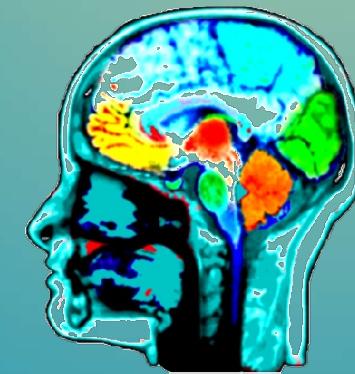
$$\begin{aligned} I(X;Y) &= 0.1 \log_2(0.1 / (0.4 * 0.3)) + \\ &\quad 0.3 \log_2(0.3 / (0.4 * 0.7)) + \\ &\quad 0.2 \log_2(0.2 / (0.6 * 0.3)) + \\ &\quad 0.4 \log_2(0.4 / (0.6 * 0.7)) = 0.029 \end{aligned}$$

Therefore, the mutual information between X and Y is 0.029.



# Introduction To Data Mining

Isfahan University of Technology (IUT)  
Bahman 1401



## Classification

---

Dr. Hamidreza Hakim  
[hamid.hakim.u@gmail.com](mailto:hamid.hakim.u@gmail.com)

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ

# Content

---

---

Classification: Basic Concepts

یکی از روش های  
معروف دسته بندی داده  
ها به نام درخت تصمیم

Decision Tree Induction

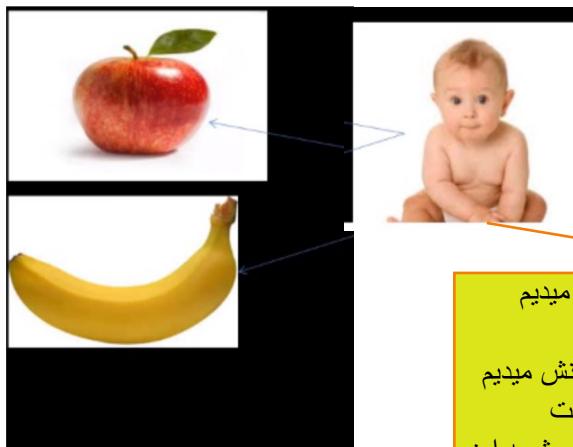
# Supervised vs. Unsupervised Learning

به ماشین یه سری داده میگیم و بهش میگیم که هرکدام از این ابجکت ها چین ینی ابجکت هارا معرفی میکنیم بش ینی لیبل و تارگت را بهش میگیم

- Supervised learning (classification)
- Unsupervised learning (clustering)

در کلسفیکیشن: شی را با یه لیبل به مدل نشون  
بیدیم تا یاد بگیره بعد یه سری داده ی تست میدیم که  
لیبل نداره  
اگه توانست تشخیص بده که چه لیبلی برای درست  
است مدل خوبی داریم

دیدی نداریم نسبت به داده ها ینی یه سری  
ابجکت داریم به مدل میگیم یه طوری  
برامون دسته بندیشون کن



به بچه نشون میدیم  
هرچیزی چیه  
سیب رو نشونش میدیم  
میگیم سیب است  
پس اگه یه شی شبیه این  
نشونش بدیم میفهمه که

<https://gowthamy.medium.com/machine-learning-vs-unsupervised-learning-f1658e12a780>



<https://www.mehrnews.com/04>

کلاسترینگ: مثل اینکه وارد تالار جشن  
بیشیم و بگیم ادم هایی که توی سالان نشستند  
را دسته بندی کن  
متلا مینونه بگه ادم های عروس و فامیل  
های داماد و دوست های عروس و ...  
در این دسته دیگه ابجکت هارا به مدل  
معرفی نمیکنیم و به خودش میسپاریم تا  
بفهمه  
از خودش میخاییم تا دسته بندی کنه داده ها  
را

# Supervised vs. Unsupervised Learning

---

---

- Supervised learning (classification)
  - Supervision: The training data (observations, measurements, etc.) are accompanied by **labels** indicating the **class of the observations**
  - New data is classified based on the training set
- Unsupervised learning (clustering)
  - The **class labels** of training data is unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of **classes** or **clusters** in the data

په سري اندازه گيري و  
مشاهدات داريم و  
براساس اونها مي خايم  
تصميم گيري کنيم

# Problems: Classification vs. Prediction

---

---

## ● Classification

- predicts **categorical class labels** (discrete or nominal)
- **classifies data** (constructs a model) based on the **training set** and the **values** (**class labels**) in a classifying attribute and uses it in classifying new data

## ● Prediction

- models **continuous-valued functions**, i.e., predicts **unknown** or **missing values**

در supervised learning دو تا مساله اصلی وجود داره:  
بعضی وقت ها بر چسبی که میخایم بزنیم به داده ها مشخص و **discrete** است مثل وقتی  
که میگیم هوا دو حالت داره : سرد است یا گرم  
یه جاهایی از مدل میخایم یه طیف را برامون پیشینی کنه مثل مثلا میخایم دما را برامون  
پیشینی کنه  
ما اطلاعات دمای روزهای قبل را داریم مثل گرما و میزان رطوبت و تابش خورشید و  
مکان جغرافیایی و ... و میگیم دما توی این نقطه با توجه به این اطلاعاتی که بت دادیم  
اینقدر حالا بیا دما را توی نقطه‌ی جدید که ازت میپرسیم پیشینی کن  
ینی میخایم چیزی را پیشینی کنی که پیوسته است یه تعداد مشخصی از حالت را نداره  
اگه تعداد حالت داشته باشه میگیم میخایم دسته بنده کنیم که میشه **classification**  
اگه تعداد دسته نداشته باشیم و یه طیف داشته باشیم میشه پیشینی یا **prediction**

# Prediction Problems: Classification vs. Prediction

- Typical applications

- Credit/loan approval:
- Medical diagnosis: if a tumor is **cancerous** or **benign**
- Fraud detection: if a transaction is **fraudulent**
- Web page categorization: **which category it is**

بے یکی وام بدم یا نه؟  
دوحالته پس تعداد گسته  
و مشخص دسته داریم  
پس classification

فردی سرطان داره یا نه؟

Fraud detection is typically considered a classification task, as the goal is to determine whether or not a given transaction or activity is fraudulent. In a classification task, the model is trained to assign input data to one or more categories (in this case, "fraudulent" or "not fraudulent") based on patterns and relationships in the training data.

Prediction, on the other hand, generally refers to a task where the aim is to predict a numerical value, such as a stock price or a customer's lifetime value. While fraud detection might involve making predictions about the likelihood of fraud occurring, the ultimate output is still a binary classification: fraudulent or not fraudulent.

Web page categorization can also be considered a classification task, as the goal is to assign web pages to one or more categories based on their content and characteristics. In this case, the categories might include topics such as "sports," "entertainment," "business," etc.

The process of web page categorization typically involves training a machine learning model using labeled data - that is, a set of web pages that have already been manually assigned to their respective categories. The model then uses patterns and relationships in the training data to predict the category of new, unlabeled web pages.

Like other classification tasks, web page categorization can be approached using various machine learning algorithms, including decision trees, support vector machines, and neural networks.

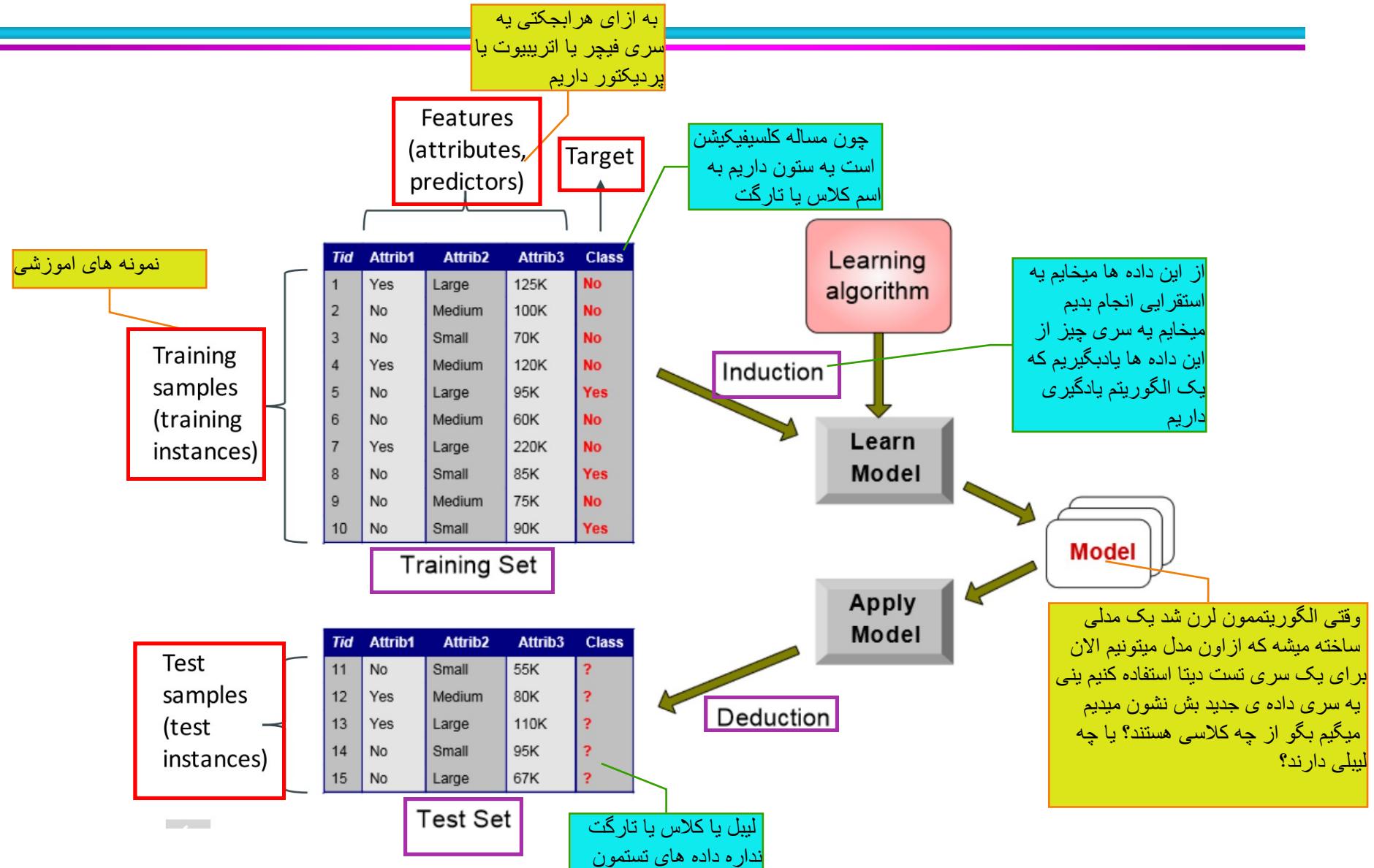
# Examples of Classification Task

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

فهرست نویسی کوهکشان ها

کوهکشان های بیضوی،  
مارپیچی یا نامنظم

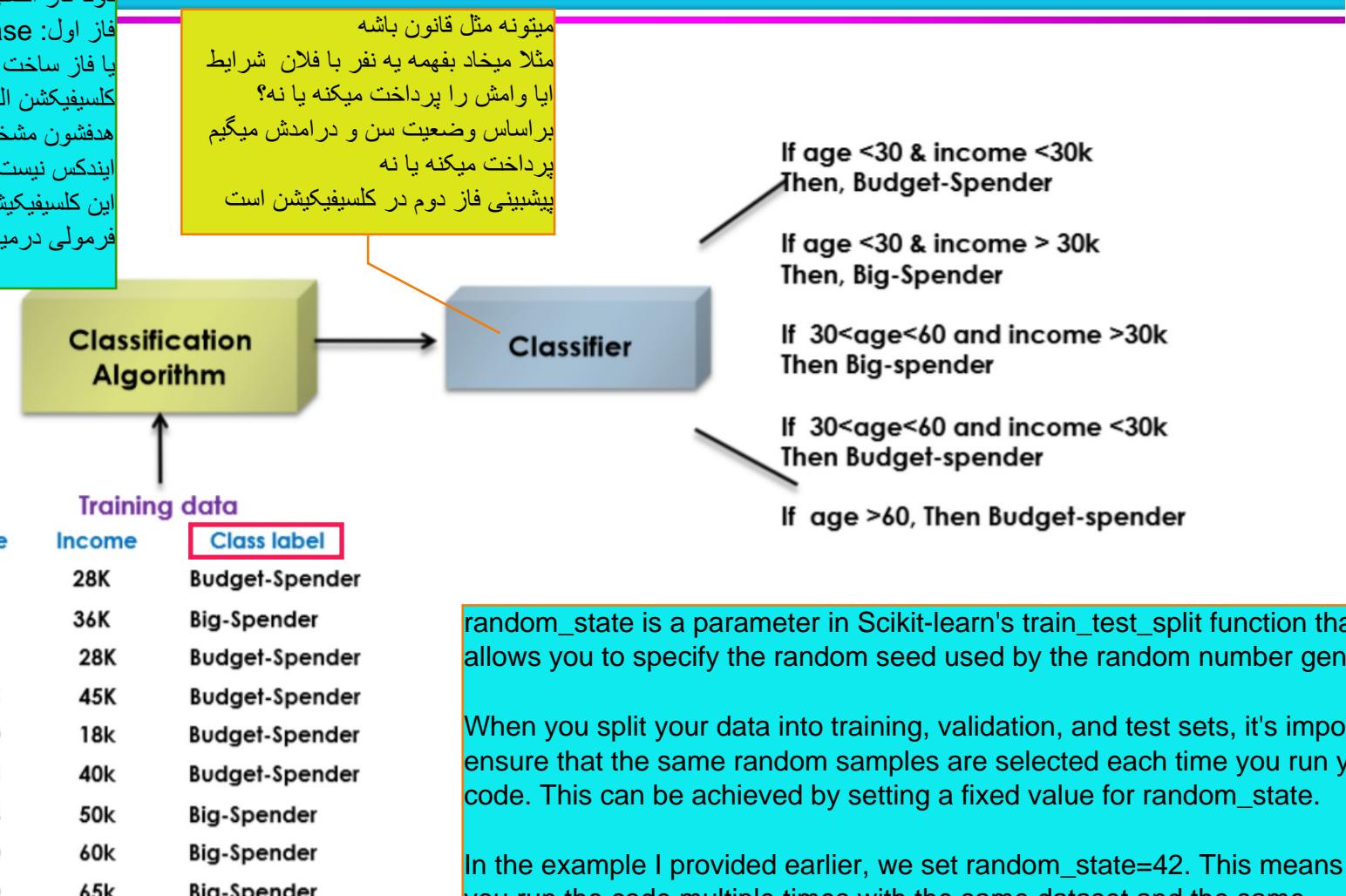
# Classification Process



# Training phase , Model construction

دروت افزار اصلی برای کلسیفیکیشن داریم

فاز اول: training phase  
یا فاز ساخت مدل یعنی داده هارا دست یک کلسیفیکیشن الگوریتم میدیم داده هایی که لیبل و هدفشوون مشخص است نیازی به ستون ایدی یا ایندکس نیست اینجا این کلسیفیکیشن الگوریتم میاد بین داده ها یه فرمولی درمیاره



random\_state is a parameter in Scikit-learn's train\_test\_split function that allows you to specify the random seed used by the random number generator.

When you split your data into training, validation, and test sets, it's important to ensure that the same random samples are selected each time you run your code. This can be achieved by setting a fixed value for random\_state.

In the example I provided earlier, we set random\_state=42. This means that if you run the code multiple times with the same dataset and the same random\_state value, you will get the same splits each time. This is useful for reproducibility and debugging purposes.

Note that the specific value of random\_state doesn't matter as long as it's consistent across runs. You can choose any integer value for random\_state to achieve this consistency.

Sure! Here's an example of how you can split a dataset into training, validation, and test sets in Python:

```
import pandas as pd
from sklearn.model_selection import train_test_split

# Load the dataset into a Pandas dataframe
df = pd.read_csv('customer_transactions.csv')

# Split the data into training and test sets (70/30 split)
X_train, X_test, y_train, y_test = train_test_split(df.drop('returned', axis=1), df['returned'], test_size=0.3, random_state=42)

# Split the training set further into training and validation sets (50/50 split)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.5, random_state=42)

# Print the shapes of each set
print(f'Training set shape: {X_train.shape}')
print(f'Validation set shape: {X_val.shape}')
print(f'Test set shape: {X_test.shape}'')
```

In this example, we first load the customer transactions dataset into a Pandas dataframe. We then use the `train_test_split` function from Scikit-learn to split the data into a 70/30 split of training and test sets.

Next, we split the training set further into a 50/50 split of training and validation sets using the same function. Finally, we print out the shapes of each set to confirm that the data has been split correctly.

Note that in this example, we assume that the dataset has a binary classification target variable 'returned' indicating whether a customer returned an item or not. We drop this column from the feature data using `drop('returned', axis=1)` when splitting into training/validation/test sets. It's also important to set a `random_state` parameter to ensure reproducibility of the splits.

I generated a sample dataset to use in the previous example. Here is an example of how the dataset could look like:

```
customer_id,age,gender,location,item,purchase_amount,returned
1,25,Female>New York,Clothing,50.0,No
2,32,Male>Los Angeles,Electronics,100.0,Yes
3,45,Female>Chicago,Books,20.0,No
4,38,Male>Miami,Home goods,75.0,No
5,22,Female>Boston,Clothing,30.0,Yes
6,56,Male>San Francisco,Electronics,200.0,No
7,29,Female>Seattle,Home goods,80.0,Yes
8,41,Male>Dallas,Clothing,55.0,No
9,34,Female>Washington D.C.,Books,15.0,Yes
10,48,Male>Houston,Home goods,90.0,No
```

[Copy Code](#)

This dataset contains 10 customer transactions with demographic information (age, gender, and location), items purchased, purchase amount, and whether they returned any items. In this example, we split the data into training, validation, and test sets to build a machine learning model that predicts whether a customer returned an item based on their demographic information and purchase history.

```
Training set shape: (3, 6)
Validation set shape: (4, 6)
Test set shape: (3, 6)
```

[Copy Code](#)

The output shows the number of rows and columns in each set after splitting the data into training, validation, and test sets. Note that the actual shapes may differ depending on the size of your original dataset and the random seed used in the split.

In this example, we split the data into a 70/30 split of training and test sets using `test_size=0.3` in the `train_test_split` function. Then, we further split the training set into a 50/50 split of training and validation sets using the same function with `test_size=0.5`.

The output shows that the training set has 3 rows and 6 columns, the validation set has 4 rows and 6 columns, and the test set has 3 rows and 6 columns.

# Evaluation and usage phases

داده های تست را اینطوری  
تعریف کردیم:  
داده هایی که مدل قبل ندیده

فاز کارایی  
میخایم داده های تستمون  
را بش بدم ببینیم چه  
نتیجه ای میده؟

یه عملکردی به ما میده ما میخایم  
یه بازخوردی از مدلمون بگیریم  
ازش میپرسیم یاد گرفتی یا نه؟  
اگه یادنگر فته نحوه ای اموزشمون

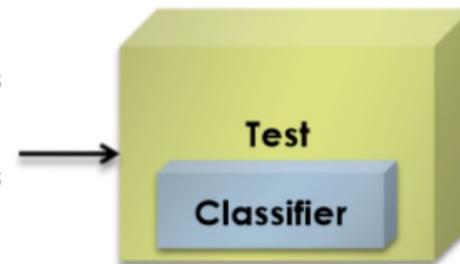
را مثلًا عوض کنیم یا داده ی  
یشتری بش بدم تا بهتر یادبگیره یا  
مدلمون را عوض کنیم

به این عملیات میگوییم  
evaluation یا ارزیابی

## 1-Test the classifier

این تست دیتاها را از قبل  
!!!!!!!  
 بش نشون ندیدم ها  
بنها دیتا های جدیده که مدل  
ندیده

Age	Income	Class label
27	28K	Budget-Spenders
25	36K	Big-Spenders
70	45K	Budget-Spenders
40	35k	Big-Spender



فیدبک میگیریم از مدلمون

## 2-If acceptable accuracy

### Unlabeled data

Age	Income
18	28K
37	40K
60	45K
40	36k

### Classified data

Age	Income	Class label
18	28K	Budget-Spenders
37	40K	Big-Spenders
60	45K	Budget-Spenders
40	36k	Budget-Spenders

# Evaluation and usage phases

مدل‌مون را چطوری ارزیابی کنیم؟  
 چندین روش برای ارزیابی کردن هست  
 اولین روش: پیداکردن میزان دقت یا accuracy  
 دومین روش: استفاده از ماتریس کانفیوژن confusion matrix  
 سومین معیار: error rate  
 تقسیم تعداد اشتباه‌ها به کل تعداد پیش‌بینی شده

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}.$$

True positive = TP

**Table 3.4.** Confusion matrix for a binary classification problem.

		Predicted Class		False Negative=FN
		Class = 1	Class = 0	
Actual Class	Class = 1	$f_{11}$	$f_{10}$	ایده‌آل ما اینه که یک هارا یک تشخیص بده و صفر‌ها را صفر
	Class = 0	$f_{01}$	$f_{00}$	
		False Positive=FP	True negative =TN	

$$\text{Error rate} = \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

Suppose we have a classification problem where we need to predict whether a customer will buy a product or not based on their demographic information. We have a dataset of 1000 customers, out of which we use 700 for training our model and the remaining 300 for testing. After training our model, we make predictions on the test set and get the following results:

- True positives (TP): 150
- False positives (FP): 30
- True negatives (TN): 100
- False negatives (FN): 20

To calculate the accuracy of our model, we can use the following formula:

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

Plugging in the values from our results, we get:

$$\text{Accuracy} = (150 + 100) / (150 + 100 + 30 + 20)$$

$$\text{Accuracy} = 0.833$$

This means that our model correctly predicted the outcome for 83.3% of the customers in the test set. We can use this accuracy score to evaluate the performance of our model and compare it with other models or benchmarks.

A confusion matrix is a performance evaluation metric used in machine learning to evaluate the accuracy of a classification model. It is a table that compares the actual values of the target variable with the predicted values of the target variable. The matrix consists of four different combinations of predictions and actual outcomes: true positive (TP), false positive (FP), false negative (FN), and true negative (TN).

Here's an example: let's say we have built a binary classifier that predicts whether an email is spam or not. We have tested the model on a dataset of 100 emails, where 60 are non-spam (class 0) and 40 are spam (class 1). The model has predicted 45 non-spam emails correctly, 15 non-spam emails incorrectly as spam, 30 spam emails correctly, and 10 spam emails incorrectly as non-spam.

The confusion matrix for this example would be:

	Predicted Not Spam	Predicted Spam
Actual Not Spam	TN = 45	FP = 15
Actual Spam	FN = 10	TP = 30

Here, TN stands for true negatives, which represents the number of non-spam emails that were correctly predicted as non-spam. FP stands for false positives, which represents the number of non-spam emails that were incorrectly predicted as spam. FN stands for false negatives, which represents the number of spam emails that were incorrectly predicted as non-spam. Lastly, TP stands for true positives, which represents the number of spam emails that were correctly predicted as spam.

From the confusion matrix, we can calculate several performance metrics such as accuracy, precision, recall, and F1 score. Accuracy measures the overall performance of the model, while precision and recall measure the model's ability to correctly predict positive cases and identify all positive cases, respectively. F1 score is a weighted average of precision and recall. These metrics can help us understand how well our model is performing and identify areas for improvement.

Performance metrics in machine learning are used to evaluate the performance of a machine learning model. These metrics help data scientists to determine how well their model is performing and identify areas for improvement. Different machine learning tasks require different performance metrics. Here are some common performance metrics in machine learning along with examples:

**Accuracy:** The most basic performance metric that measures the percentage of correctly predicted instances among all the instances in the dataset. It's calculated by dividing the number of correct predictions by the total number of predictions. Example: If a model correctly predicts 90 out of 100 instances, then the accuracy score would be 90%.

**Precision:** Measures the percentage of true positive predictions among all positive predictions made by the model. It's calculated by dividing the number of true positives by the sum of true positives and false positives. Example: If a model predicts 30 positives and 25 of them are actual positives, then the precision score would be 83.33%.

**Recall:** Measures the percentage of true positive predictions among all actual positive instances in the dataset. It's calculated by dividing the number of true positives by the sum of true positives and false negatives. Example: If there are 50 actual positives in the dataset and the model predicts 40 of them, then the recall score would be 80%.

**F1 Score:** The harmonic mean of precision and recall. It provides a balance between precision and recall. It's calculated as  $2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$ . Example: If the precision and recall scores are 0.85 and 0.75 respectively, then the F1 score would be 0.79.

**Mean Absolute Error (MAE):** A regression performance metric that measures the average absolute difference between the predicted and actual values. Example: If the predicted value is 10 and the actual value is 7, then the absolute difference is 3. If the model makes 10 predictions with absolute differences of 3, then the MAE would be  $3/10 = 0.3$ .

**Root Mean Squared Error (RMSE):** Another regression performance metric that measures the square root of the average squared difference between the predicted and actual values. Example: If the predicted value is 10 and the actual value is 7, then the squared difference is 9. If the model makes 10 predictions with squared differences of 9, then the RMSE would be the square root of  $9/10 = 0.95$

Let's say we have a binary classification problem where we want to predict whether a patient has a disease or not. We have a dataset of 100 patients, where 60 patients do not have the disease and 40 patients have the disease. We use a machine learning model to make predictions on this dataset and obtain the following confusion matrix:

	<b>Predicted No</b>	<b>Predicted Yes</b>
<b>Actual No</b>	50	10
<b>Actual Yes</b>	5	35

From this confusion matrix, we can calculate the following metrics:

Accuracy: the proportion of correct predictions out of all predictions made by the model.

Accuracy = (True Positives + True Negatives) / Total Population

Accuracy =  $(50 + 35) / 100$

Accuracy = 0.85

So the accuracy of our model is 85%.

Precision: the proportion of true positives (patients predicted as having the disease who actually have the disease) out of all patients predicted as having the disease.

Precision = True Positives / (True Positives + False Positives)

Precision =  $35 / (35 + 10)$

Precision = 0.78

So the precision of our model is 78%.

Recall: the proportion of true positives out of all actual positives (patients who actually have the disease).

Recall = True Positives / (True Positives + False Negatives)

Recall =  $35 / (35 + 5)$

Recall = 0.88

So the recall of our model is 88%.

F1 score: a combined metric that takes into account both precision and recall.

F1 Score =  $2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$

F1 Score =  $2 * (0.78 * 0.88) / (0.78 + 0.88)$

F1 Score = 0.83

So the F1 score of our model is 0.83.

In machine learning, we typically divide our available data into three sets: training set, validation set, and test set.

The purpose of the training set is to train the model. The purpose of the validation set is to tune hyperparameters or select the best model among multiple possible models. The purpose of the test set is to evaluate the final performance of the model on unseen data.

The key difference between the validation set and the test set is their usage in the machine learning process. The validation set is used during the training process to validate different models' performance on unseen data and select the best performing one. In contrast, the test set is used only once at the end of the training process to evaluate the model's generalization ability on previously unseen data.

Here's an example to illustrate the difference:

Suppose you have a dataset of images of cats and dogs. You can divide this data into training, validation, and test sets. You can use 70% of the data for training (training set), 15% of the data for model selection (validation set), and the remaining 15% for evaluating the final performance of the model (test set).

During the training process, the model is trained on the training set, and its performance is evaluated on the validation set. Multiple models with different hyperparameters can be trained and validated using the validation set, and the best performing model is selected for final evaluation.

Once the final model is selected, it is evaluated on the test set. This provides an unbiased estimate of the model's performance on completely unseen data.

# Classification—A Two-Step Process

- Model construction: describing a set of predetermined classes
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the **class label attribute**
  - The **set of tuples** used for model construction is **training set**
  - The model is represented as **classification rules**, **decision trees**, or **mathematical formulae**
- Model usage: for classifying future or **unknown objects**
  - Estimate accuracy of the model
    - ◆ The **known label of test sample** is **compared** with the **classified result** from the model
    - ◆ **Accuracy** rate is the percentage of test set samples that are correctly classified by the model
    - ◆ **Test set** is **independent** of training set (otherwise overfitting)
  - If the accuracy is acceptable, use the model to **classify new data**
- Note: If the **test set** is used to **select models**, it is called **validation (test) set**

شبکه های عصبی فرمول ریاضی برآمون  
در مبارزه در کلسفیکیشن  
مدلمن مثلا یه کلسفیکیشن روی است ینی  
ما میخایم قوانین را از داده ها استخراج کنیم

# Binary classification vs Multiclass classification

## Binary classification

- Classification tasks with **only two classes**, typically denoted by  $\{+,-\}$ ,  $\{+1,-1\}$ , or  $\{\text{Pos}, \text{Neg}\}$ .
- Example: **email spam detection**, **(pos/neg) sentiment analysis**.

## Multiclass classification

- Classification tasks with **more than two classes**.
- Example: **email topic detection**, **(pos/null/neg) sentiment analysis**.

سوال یا مساله:

یه نمودار قیمت دلار داریم میخایم ببینیم قیمتش بالا میره یا پایین در روز اینده این که جواب مساله دو تا حالت داره میشه باینری کلسیفیکیشن میتوونیم یه حالت دیگه هم اضافه کنیم به جواب ها قیمت ثابت بماند نسبت به روز قبل

پس الان سه تا جواب داره مساله میشه مولتی کلاس مسائل کلسیفیکیشن چند حالته را میشه با دو حالته هم حل کرد چطوری؟

مثلا بگیم قیمت ثابت است یا ثابت نیست؟  
اول اینو میپرسیم از ش

اگه گفت ثابت که هیچی اگه گفت غیر ثابت میگیم کم میشه یا زیاد؟ اینجا نیاز به یک کلسیفایر دیگه داریم که درباره ی بالا و پایین بودن قیمت تصمیم گیری کنه

# Classification Techniques

---

---

- Decision Tree based Methods
- Rule-based Methods
- Nearest-neighbor
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines
- Neural Networks, Deep Neural Nets
- Ensemble Classifiers
  - ◆ Boosting, Bagging, Random Forests
- And many more

ترکیب کلاسیفایر های  
مختلف باهم

---

---

# **DECISION TREE**

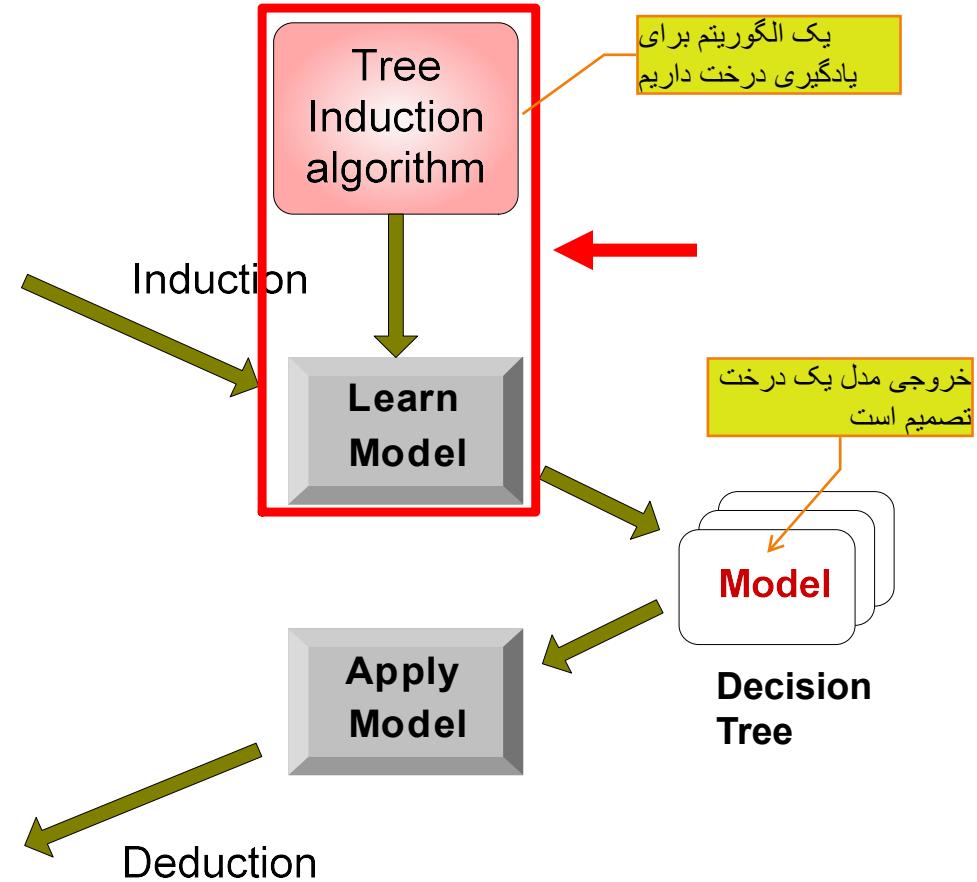
# Decision Tree Classification Task

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

Training Set

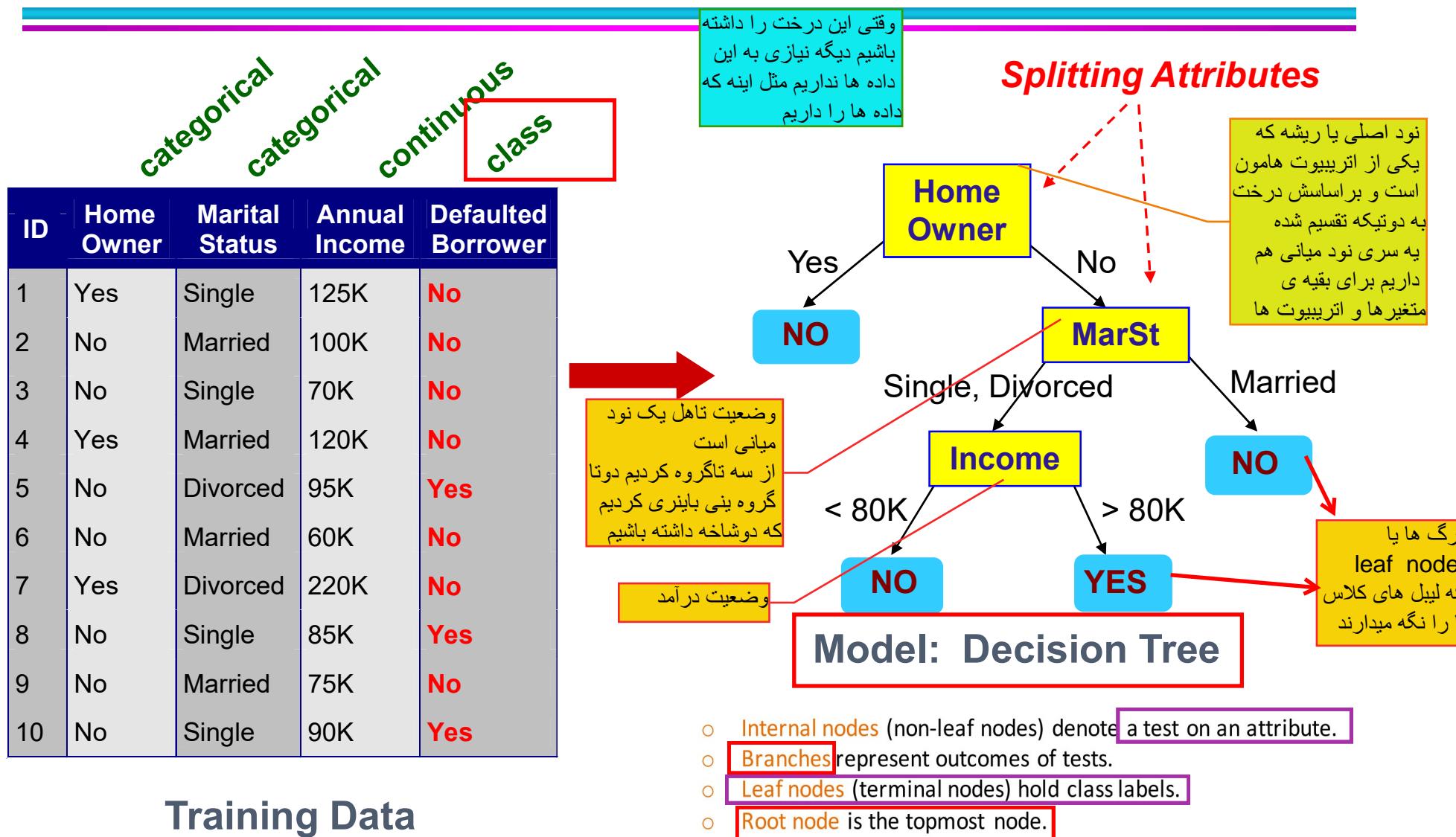
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



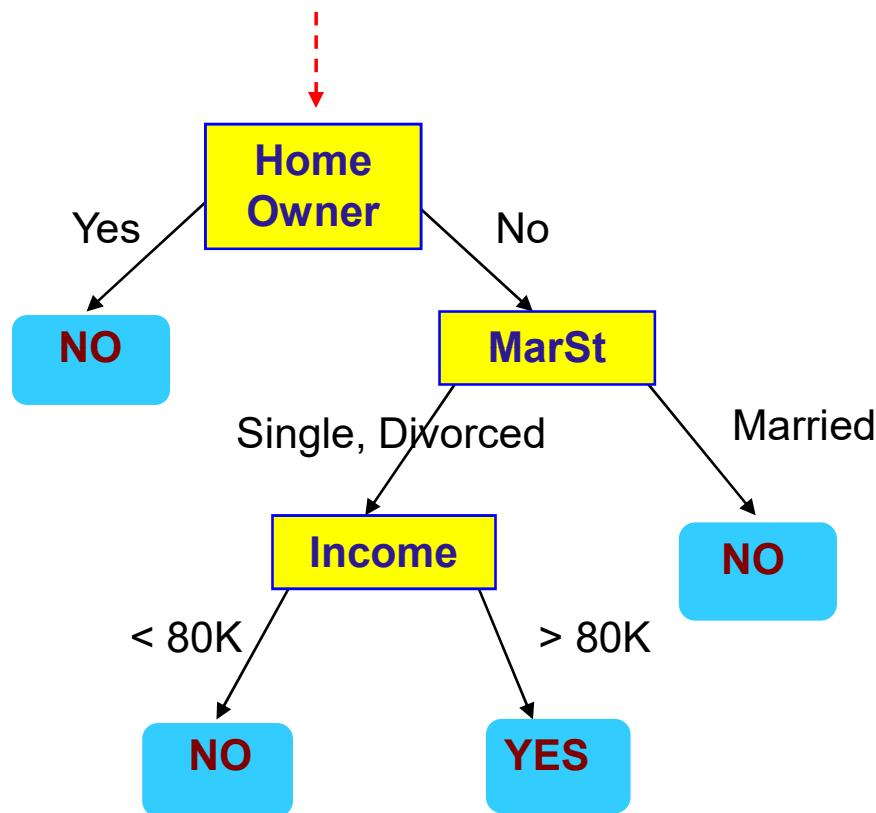
# Example of a Decision Tree

سوال یا مساله:  
اگه یه ادمی وام بگیره ایا پس میده یا نه؟



# Apply Model to Test Data

Start from the root of tree.

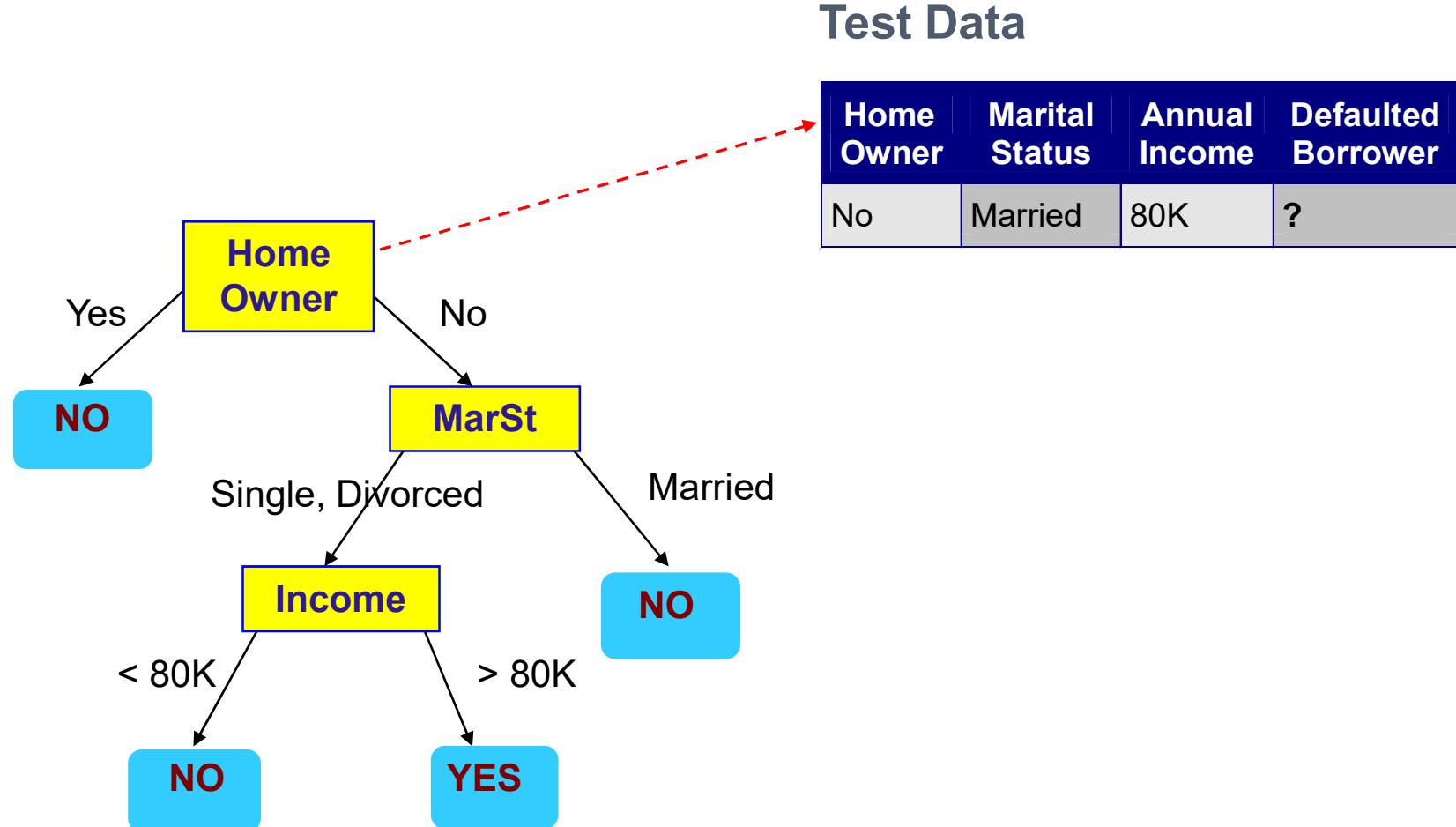


## Test Data

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

وقتی کلسیفیکیشن میکنیم لزوما به یک مدل نمیرسیم به یک درخت یکسان لزوما نمیرسیم بنی میتوانیم چندین مدل داشته باشیم روی داده هامون سوال ۱: چطوری این درخت را بدست اورد؟  
چطوری تقسیم بندی کرد شاخه ها را؟

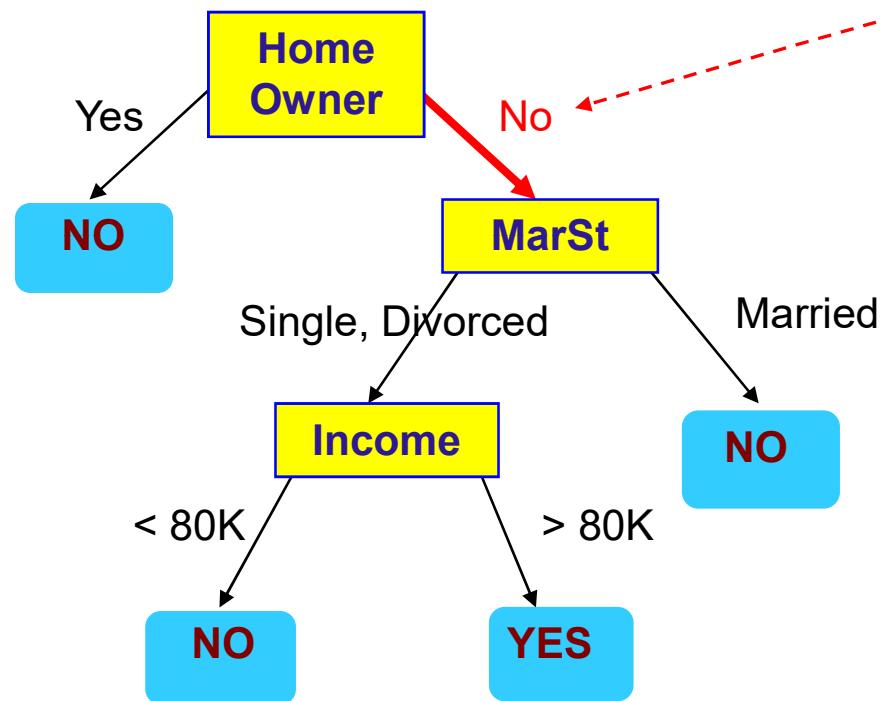
# Apply Model to Test Data



# Apply Model to Test Data

Test Data

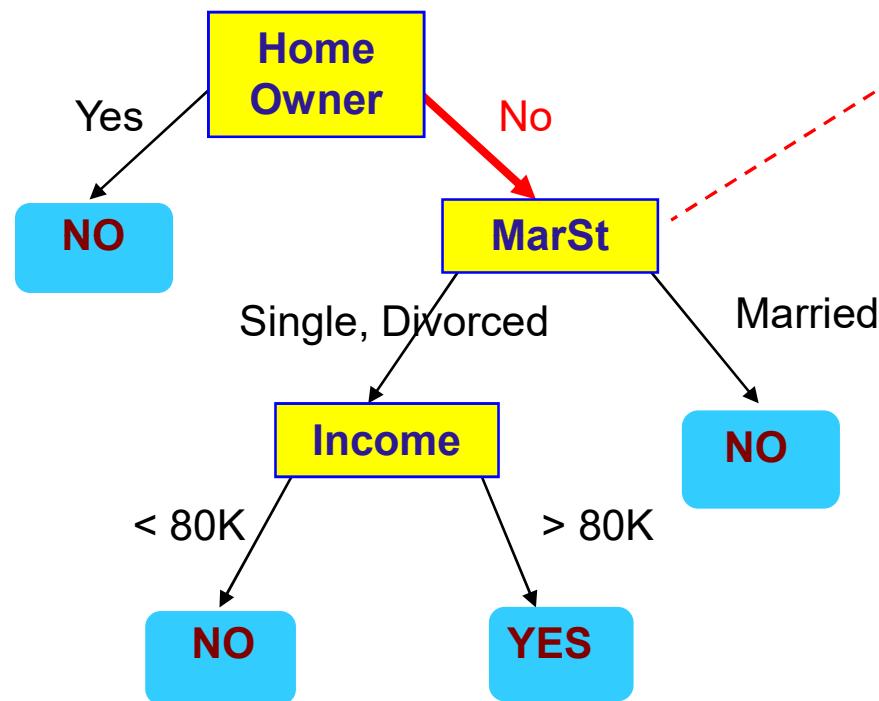
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



# Apply Model to Test Data

Test Data

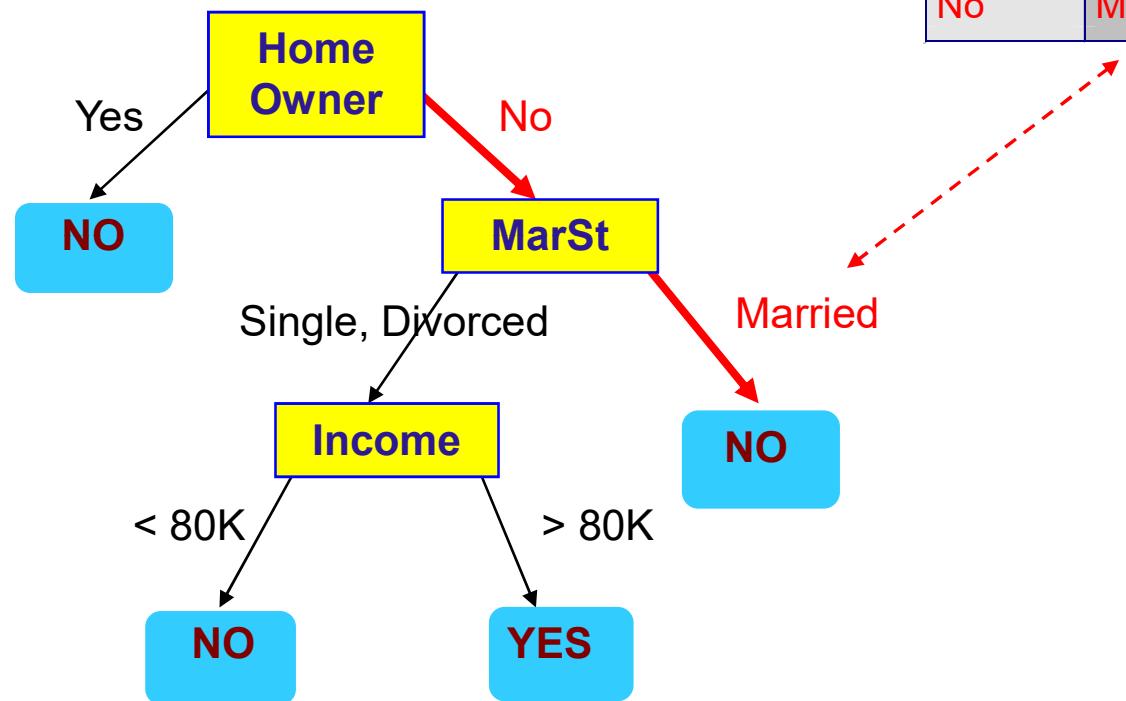
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



# Apply Model to Test Data

Test Data

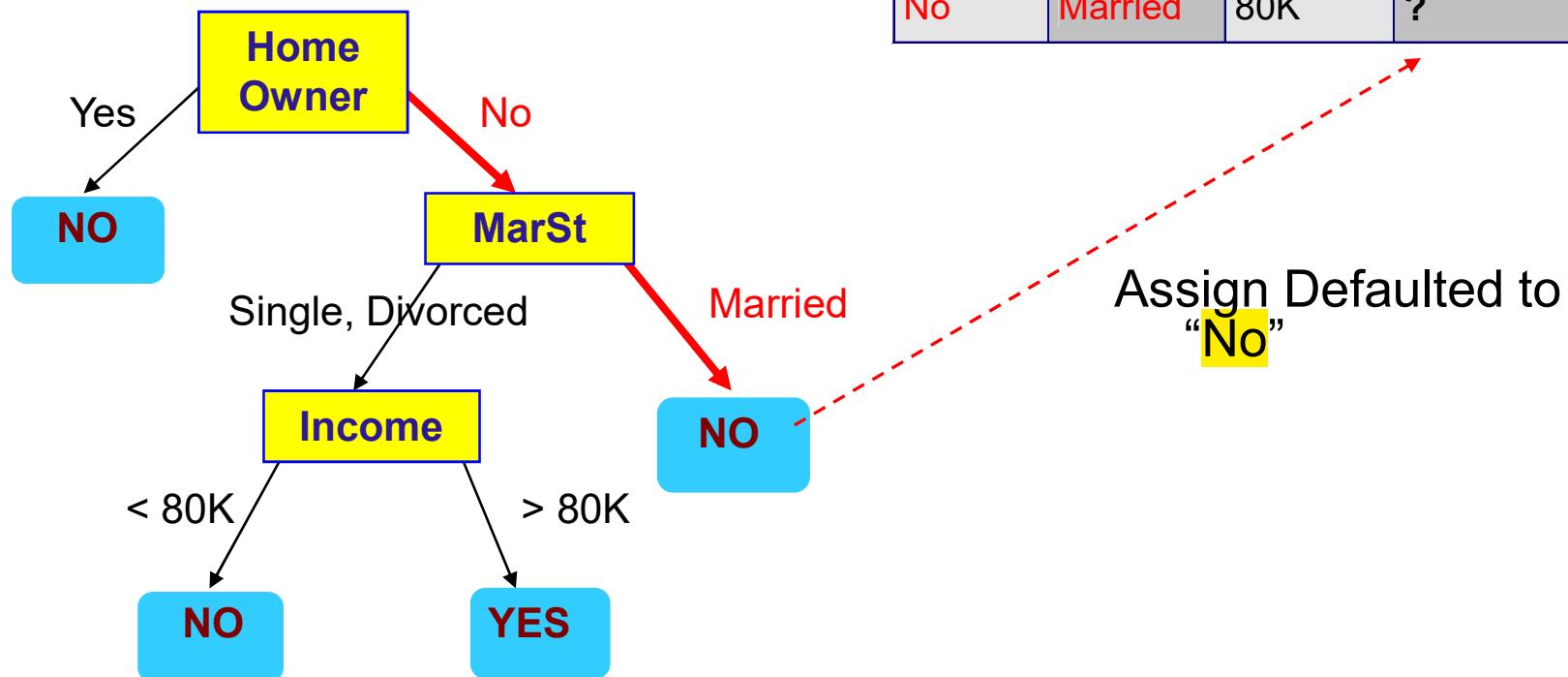
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



# Apply Model to Test Data

Test Data

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



Building an optimal decision tree for your dataset requires careful consideration of several factors. Here are some steps you can follow to build the best decision tree:

**Data Preparation:** Ensure that your data is properly preprocessed and cleaned. This includes removing missing values, handling outliers, and transforming categorical variables.

**Feature Selection:** Choose the most relevant features for your model. You can use techniques like correlation analysis, principal component analysis (PCA), or recursive feature elimination (RFE) to identify the most important features.

**Splitting Criteria:** Select the appropriate splitting criteria to partition your data into the subsets that contain the most homogeneous class labels. Commonly used splitting criteria include Gini index and entropy.

**Pruning:** Decide when to stop growing the tree by setting a stopping criterion to avoid overfitting. Pruning can improve the accuracy of the decision tree by removing irrelevant or redundant branches.

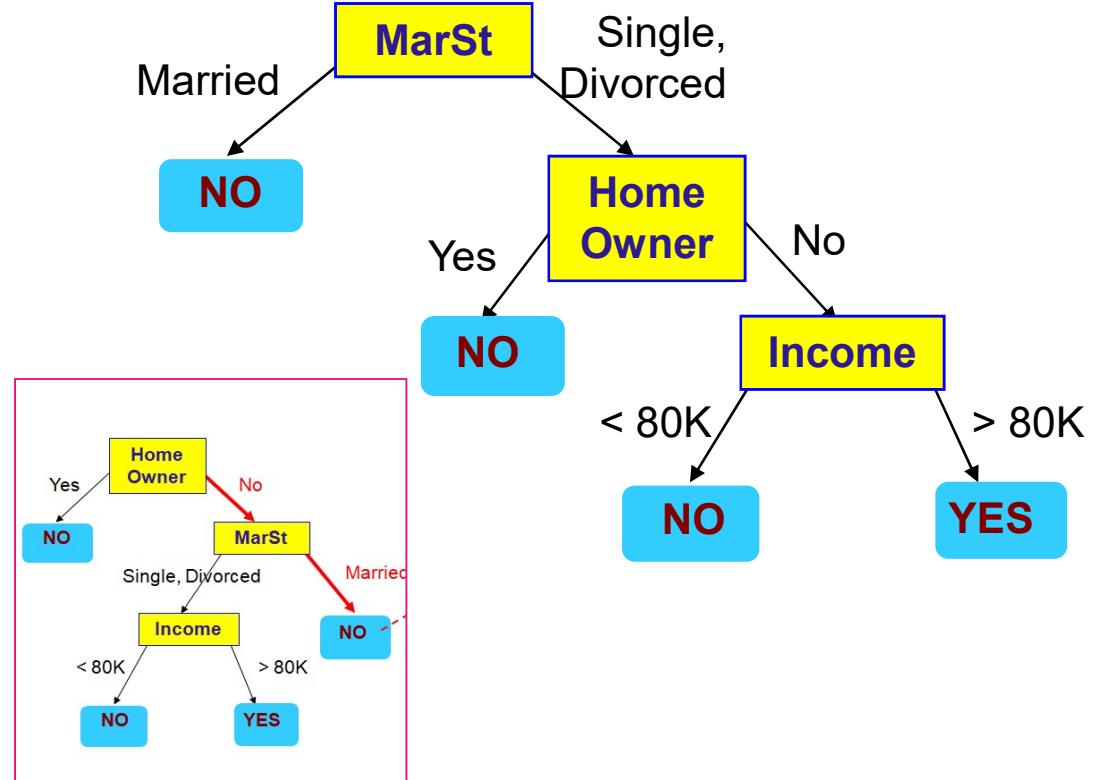
**Tuning Parameters:** Fine-tune the hyperparameters of your decision tree algorithm. This includes parameters like the maximum depth of the tree, minimum samples required to split a node, and minimum samples required in a leaf.

**Validation:** Evaluate the performance of your decision tree using techniques like cross-validation, holdout validation, or bootstrapping.

By following these steps, you can build the best possible decision tree for your dataset.

# Another Example of Decision Tree

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



There could be more than one tree that fits the same data!

# Decision Tree Induction

---

---

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

# General Structure of Hunt's Algorithm

- Let  $D_t$  be the set of training records that reach a node  $t$

- General Procedure:**

— If  $D_t$  contains records that belong to the same class  $y_t$ , then

$t$  is a leaf node labeled as  $y_t$

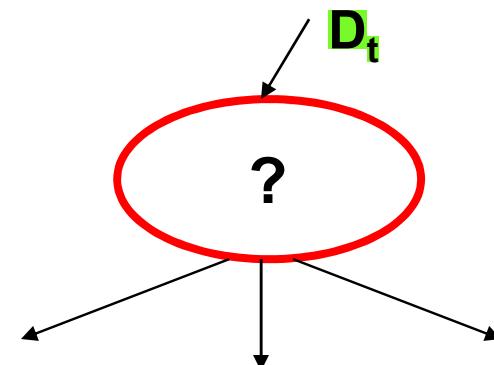
— If  $D_t$  contains records that belong to more than one class,

use an attribute test to split the data into smaller subsets.

Recursively apply the procedure to each subset.

اتributti که انتخاب میکنیم  
مساله را برآمودن میشکنند

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



# Hunt's Algorithm

Defaulted = No

(7,3)

(a)

۱. تا رکورد داریم که ۳ تاشون لیبل yes دارند و  
۷ تاشون لیبل no  
پس ۷ تا برای یک کلاسه و ۳ تا برای کلاس دیگه

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

# Hunt's Algorithm

Defaulted = No

(7,3)

(a)

اینجا رسیدیم به وضعیتی  
که همه داده هامون یک  
لیبل یا یک برچسب یا  
یک تارگت نهایی دارند  
بنی همه ی رکوردهایی  
که توی این شاخه اومدن  
لیبلشون no است  
پس این نود را به عنوان  
برگ درنظر میگیریم



(3,0)

(b)

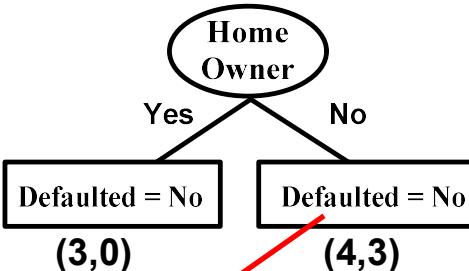
از این ۷تا، ۳تاشون برچسب yes  
دارند و ۴تاشون برچسب no  
اینجا چون دو تا برچسب دارن داده  
ها باید شاخه بندی کردن را ادامه  
بدم تا جایی که به برگ برسیم بنی  
باید اتریبیوت های بعدی را درنظر  
بگیریم

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

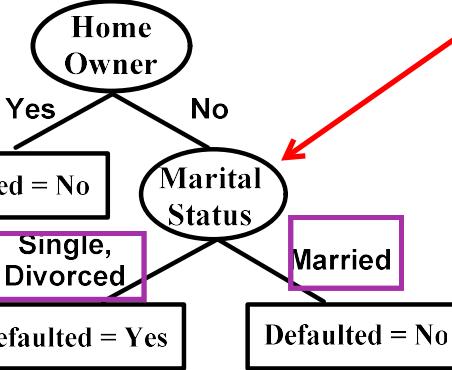
# Hunt's Algorithm

Defaulted = No  
(7,3)

(a)



(b)



اینجا هنوز عدم قطعیت داریم و برای اینکه بتوانیم تصمیم گیری کنیم برای داده ها که درنهایت بشون برچسب پس بدمیم یا نه باید براساس یک اتریبیوت دیگه هم تقسیم‌بندی کنیم

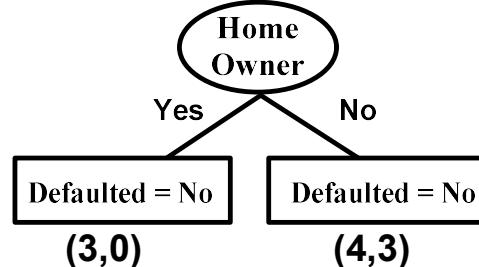
میاییم وضعیت تا هل را برآشون بررسی میکنیم

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

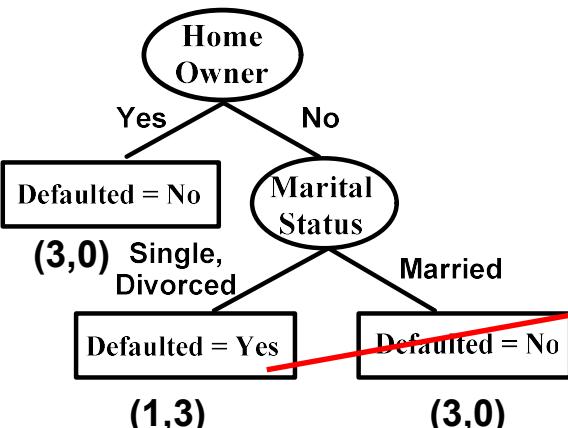
# Hunt's Algorithm

Defaulted = No  
(7,3)

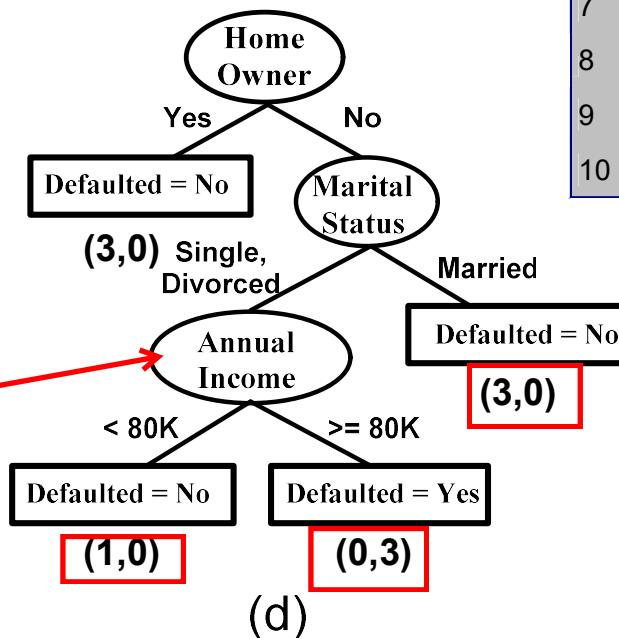
(a)



(b)



(c)



(d)

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

الآن قطعیت همه کامل شد یعنی  
همه حالت ها به برگ رسید

# Design Issues of Decision Tree Induction

- How should training records be split?

(splitting criterion)

1. Gini index
2. Entropy
3. information gain

برچه اساسی باید تقسیم بندی کنیم؟  
معیار تفکیک کردن

- How should the splitting procedure stop?

(stopping criterion)

این روند تقسیم کردن و شاخه بندی چه زمانی باید تمام شه؟  
معیار توقف

# Design Issues of Decision Tree Induction

---

---

- How should the splitting procedure **stop?**

(stopping criterion)

– Stop splitting if **all the records** belong to the **same class** or have **identical attribute values**

به شاخه ای برسیم که  
برچسب همه یکسان بشن

جایی منطق بنشیم که هیچ  
اتribute دیگه ای نیست

– There are **no remaining attributes** for further partitioning

یه راگیری باید میگرفتیم که مثلاً اونایی که تعداد لبیل بس  
بیشتر دارند شاخه به اونا تعلق داره

– There are **no samples left** – majority voting on the **parent's samples** is employed.

# Design Issues of Decision Tree Induction

- How should training records be split?  
(splitting criterion)
  - Method for expressing test condition
    - ◆ depending on attribute types
  - Measure for evaluating the goodness of a test condition

سوال:  
عدد ۸۰ هزار را از کجا  
ورد برای درآمد که براساس  
اون اوMD اسپلیت کرد؟  
یا از کجا فهمید که باید مجرد  
هارا بفرسته توی یه شاخه  
و بقیه که شامل متاهل و  
طلاق گرفته بودن رو  
بفرسته توی شاخه ی دیگه؟

# Methods for Expressing Test Conditions

---

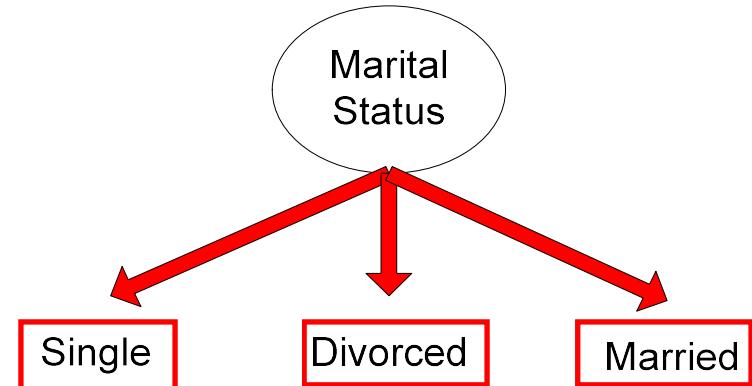
---

- Depends on attribute types
  - Binary
  - Nominal
  - Ordinal
  - Continuous

# Test Condition for Nominal Attributes

- **Multi-way split:**

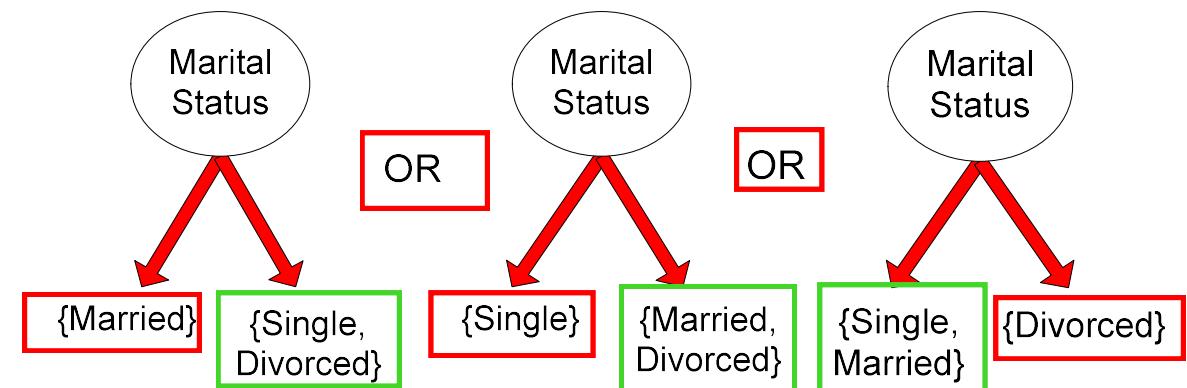
- Use as many partitions as distinct values.



- **Binary split:**

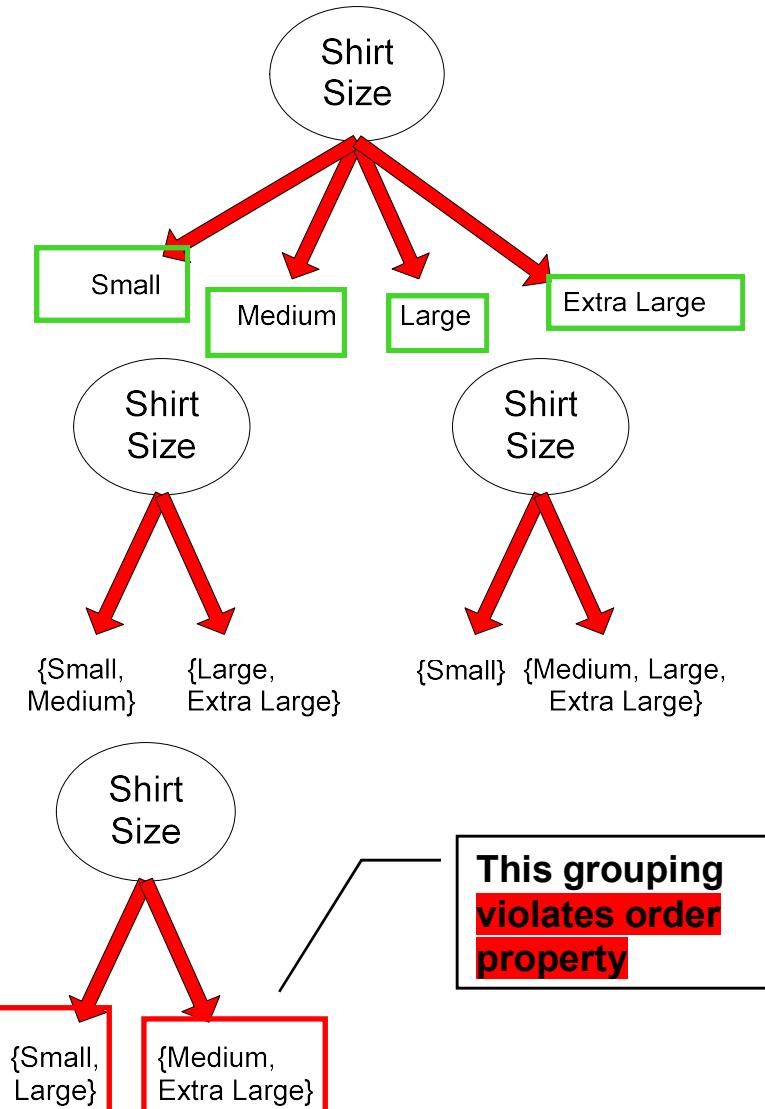
- Divides values into two subsets

معیار تفکیک کردن  
علاوه بر بستگی داشتن به نوع اتریبیوت به  
خود اتریبیوت هم بستگی داره اینکه چه  
اتریبیوتی را انتخاب کنیم برای اسپلیت کردن؟  
کدوم را بگذاریم برای نود روت؟ بعدش کدوم  
انتخاب شه؟  
نحوه ی انتخاب کردن مقادیر برای جدا کردن  
مثل اینکه برای درامد ۸۰ را بگذاریم یا ۸۱  
متاهم ها با مجرد ها باشن یا با طلاق گرفته  
های؟  
مسئله دوم اینکه کدام اتریبیوت را زودتر  
بگذاریم؟

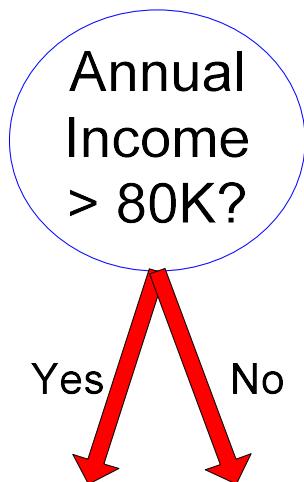


# Test Condition for Ordinal Attributes

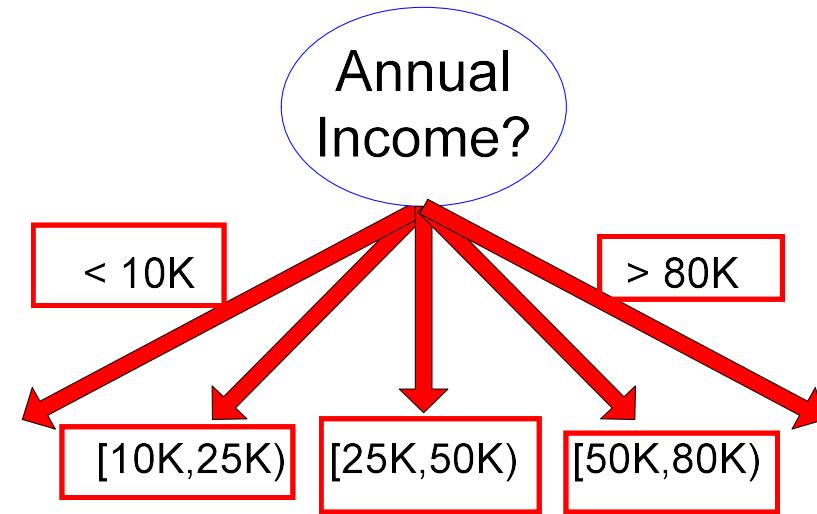
- **Multi-way split:**
  - Use as many partitions as distinct values
- **Binary split:**
  - Divides values into two subsets
  - **Preserve order property** among attribute values



# Test Condition for Continuous Attributes



(i) Binary split



(ii) Multi-way split

# Splitting Based on Continuous Attributes

- Different ways of handling
  - Discretization to form an ordinal categorical attribute

Ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.

    - ◆ Static – discretize once at the beginning
    - ◆ Dynamic – repeat at each node
  - Binary Decision:  $(A < v)$  or  $(A \geq v)$ 
    - ◆ consider all possible splits and finds the best cut
    - ◆ can be more compute intensive

بازه بندی کردن مقادیر

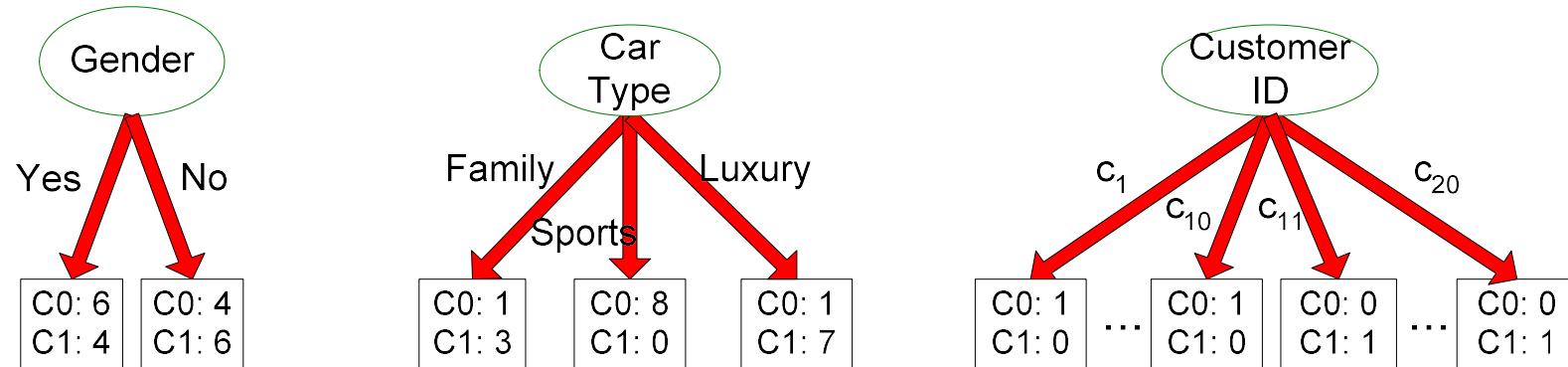
# How to determine the Best Split

۲۰ تا رکورد داریم و دو تا کلاس  
۱۰ تا از رکوردها برای کلاس  
صفر است و ۱۰ تا برای یک

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

۳. انتخاب داریم  
برای نود روت  
۱. جنسیت بشه  
۲. نوع ماشین  
۳. ایدی مشتری

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



**Which test condition is the best?**

2/1/2021

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

بیشتر از همه خالص و  
پیور بکنه داده هارا

یه معیاری میخایم که به تصمیم گیریمون کمک کنه مثلاً تقسیم  
بندی را پیورتر بکنه مثلاً اینطوری نباشه که اول کار داده ها  
مون ۵۰ ۵۰ باشن و بعد از تقسیم بندی هم هنوز اون ابهامه تویی  
تقسیم بندیمون باشه اینطوری هیچی از تقسیم بندی نفهمیدیم و  
انگار اصلاً تقسیم بندی نکردیم پس دنبال یه معیاری هستیم که  
میزان خالص بودن تقسیم بندی را اندازه بگیره و هر کدام خالص  
تر بود انتخاب کنیم

# Measures of Node Impurity

- Gini Index

$$Gini\ Index =$$

$$1 - \sum_{i=0}^{c-1} p_i(t)^2$$

به ازای کل کلاس هایی  
که توی نود داریم.

Where  $p_i(t)$  is the frequency  
of class  $i$  at node  $t$ , and  $c$  is  
the total number of classes

- Entropy

$$Entropy =$$

$$-\sum_{i=0}^{c-1} p_i(t) \log_2 p_i(t)$$

احتمال هر کلاسی که داریم  
در یک نود مشخص مثل  $t$

- Misclassification error

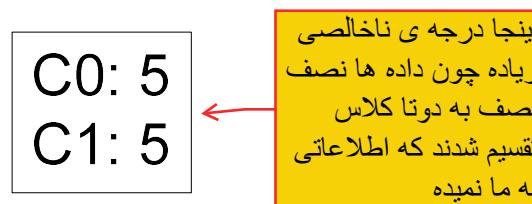
$$Classification\ error =$$

$$1 - \max[p_i(t)]$$

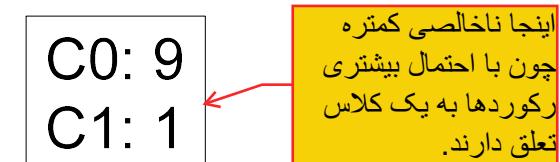
ماکس احتمال رخدادن  
یک کلاس توی یک نود

# How to determine the Best Split

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



High degree of impurity



Low degree of impurity

# Finding the Best Split

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

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

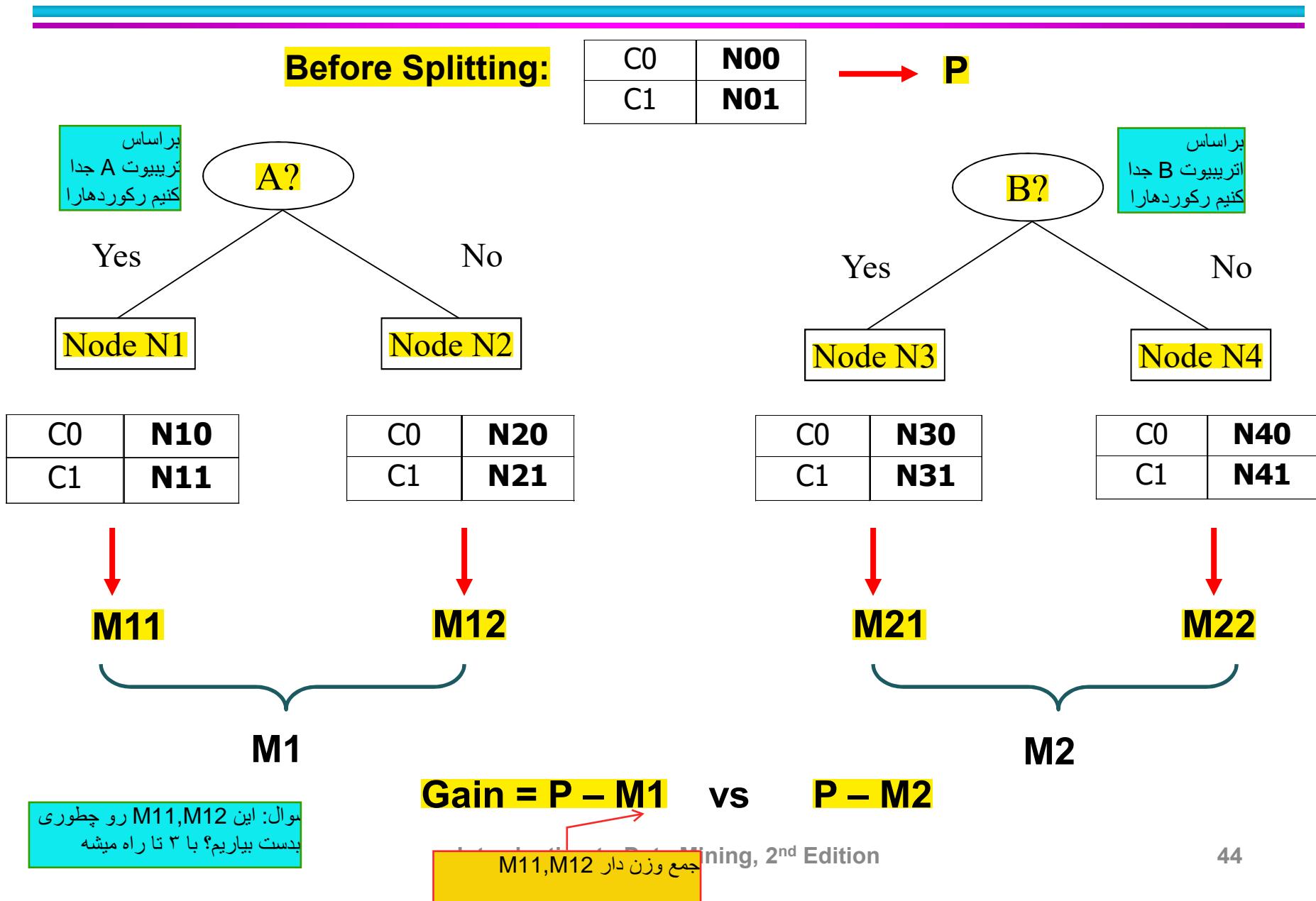
میزانی که از ناچالصی  
داده هامون کم شده میشه  
گین ما یا دستاورد ما  
هرچی این میزان بیشتر  
باشه بهتره

بهترین split اونیه که  
1. بعد از انجام دادنش، بیشترین گین رو بدست  
بیاریم  
یا میشه بگیری  
2. بعد از اسپلیت شدن، کمترین میزان ناچالصی  
را بمون بد

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

اونی که کمترین  $M$  را  
میده انتخاب میکنیم.

# Finding the Best Split



این معیار جینی ایندکس داره ناخالصی رو نشون میده ما هم که به دنبال خالص شدن تقسیم بندی هامون هستیم پس بهترین حالتش اینه که جینی ایندکس صفر بشه یعنی ناخالصی نداشته باشیم و بدترین حالتش اینه که احتمال همه ی کلاس ها توانی یک نود یکسان بشه یعنی توزیع یکسان بین کلاس ها

# Measure of Impurity: GINI

## Gini Index for a given node $t$

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

اگه احتمال همه کلاس ها  
برابر باشه:  
 $c * (1/c)^2 =$   
 $c * (1/c^2) = 1/c$   
gini index =  $1 - 1/c$

Where  $p_i(t)$  is the frequency of class  $i$  at node  $t$ , and  $c$  is the total number of classes

- Maximum of  $1 - 1/c$  when records are equally distributed among all classes, implying the least beneficial situation for classification
- Minimum of 0 when all records belong to one class, implying the most beneficial situation for classification

اگه احتمال یکی از کلاس ها یک باشه  
و بقیه صفر باشه:  
gini index = 0  
یعنی قطعیت داریم و مثلًا برچسب همه  
ی دیتاها مون یه چیز یکسان شده

# Measure of Impurity: GINI

- Gini Index for a given node t :

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

- For 2-class problem ( $p, 1 - p$ ):

binary classification

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

بهترین حالت که  
ناخالصی صفر است.

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

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

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

بدترین حالت که رکوردها به صورت  
مساوی بین کلاس ها توزیع شدند.  
 $p(c1) = p(c2) = 3/6 = 1/2$   
 $gini\ index = 1 - 1/c$   
 $c = 2$   
چون ۲ تا کلاس داریم پس  
 $gini\ index = 1 - 1/2 = 1/2$

# Computing Gini Index of a Single Node

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

C1	0
C2	6

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

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

C1	1
C2	5

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

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

C1	2
C2	4

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

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

## Computing Gini Index for a Collection of Nodes

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

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

جینی هر پارتیشن را در  
احتمال اون تعداد  
رکوردی که رفتد توی  
اون پارتیشن ضرب  
میکنیم و باهم جمع میکنیم

where,

$n_i$  = number of records at child  $i$ ,

$n$  = number of records at parent node  $p$ .

جمع وزن دار جینی ایندکس ها

### Example 1:

Suppose we have a dataset with 100 samples and two classes (A and B). We want to split the data based on a feature X that can take on two values (0 or 1). The Gini index for the original dataset is:

$$\text{Gini}(D) = 1 - (50/100)^2 - (50/100)^2 = 0.5$$

Now, let's say that when X=0, there are 40 samples of class A and 10 samples of class B, and when X=1, there are 10 samples of class A and 40 samples of class B. The Gini index for each subset is:

$$\text{Gini}(D_0) = 1 - (40/50)^2 - (10/50)^2 = 0.32$$

$$\text{Gini}(D_1) = 1 - (10/50)^2 - (40/50)^2 = 0.32$$

To calculate the Gini index for the split, we take a weighted average of the Gini indices for each subset:

$$\text{Gini}_{\text{split}} = (50/100)\text{Gini}(D_0) + (50/100)\text{Gini}(D_1) = 0.32$$

If we compare this to the original Gini index of 0.5, we can see that the split has reduced the impurity of the data.

### Example 2:

Suppose we have a dataset with 100 samples and three classes (A, B, and C). We want to split the data based on a feature X that can take on three values (0, 1, or 2). The Gini index for the original dataset is:

$$\text{Gini}(D) = 1 - (30/100)^2 - (40/100)^2 - (30/100)^2 = 0.66$$

Now, let's say that when X=0, there are 20 samples of class A, 5 samples of class B, and 5 samples of class C. When X=1, there are 5 samples of class A, 20 samples of class B, and 5 samples of class C. When X=2, there are 5 samples of class A, 5 samples of class B, and 20 samples of class C. The Gini index for each subset is:

$$\text{Gini}(D_0) = 1 - (20/30)^2 - (5/30)^2 - (5/30)^2 = 0.53$$

$$\text{Gini}(D_1) = 1 - (5/30)^2 - (20/30)^2 - (5/30)^2 = 0.53$$

$$\text{Gini}(D_2) = 1 - (5/30)^2 - (5/30)^2 - (20/30)^2 = 0.53$$

To calculate the Gini index for the split, we take a weighted average of the Gini indices for each subset:

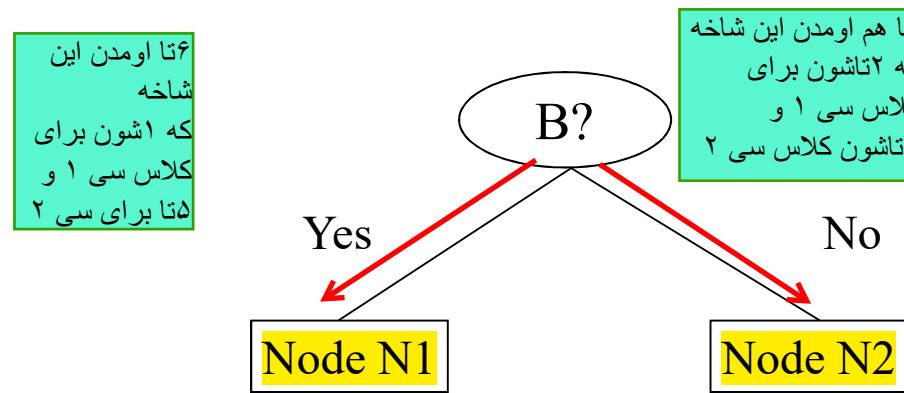
$$\text{Gini}_{\text{split}} = (30/100)\text{Gini}(D_0) + (30/100)\text{Gini}(D_1) + (40/100)\text{Gini}(D_2) = 0.53$$

If we compare this to the original Gini index of 0.66, we can see that the split has reduced the impurity of the data.

# Binary Attributes: Computing GINI Index

- Splits into two partitions (child nodes)
- Effect of Weighing partitions:
  - Larger and purer partitions are sought

$$p1 = 7/12 \\ p2=5/12 \\ \text{gini index} = 1 - ( (7/12)^2 + (5/12)^2 ) = 1 - (0.34 + 0.17) = 1 - 0.51 = 0.49$$



**Gini(N1)**

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

**Gini(N2)**

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

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

**Weighted Gini of N1 N2**

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

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

محاسبه حالت وزن دار

این مقدار بمون کمک میکنه بین یه اتریبیوت و اتریبیوت های دیگه مقایسه کنیم

# Categorical Attributes: Computing Gini Index

- For each distinct value, gather counts for each class in the dataset
- Use the count matrix to make decisions

۳ تا فرزند درست  
میشه از این اسپلیت  
کردن پس ۳ تا جینی  
خواهیم داشت که باید  
در احتمال پارتویشن  
بندی شاخه ها هم  
ضرب کنیم که بشه  
کل جینی این اسپلیت

Multi-way split

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

Two-way split  
(find best partition of values)

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

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

Which of these is the best?

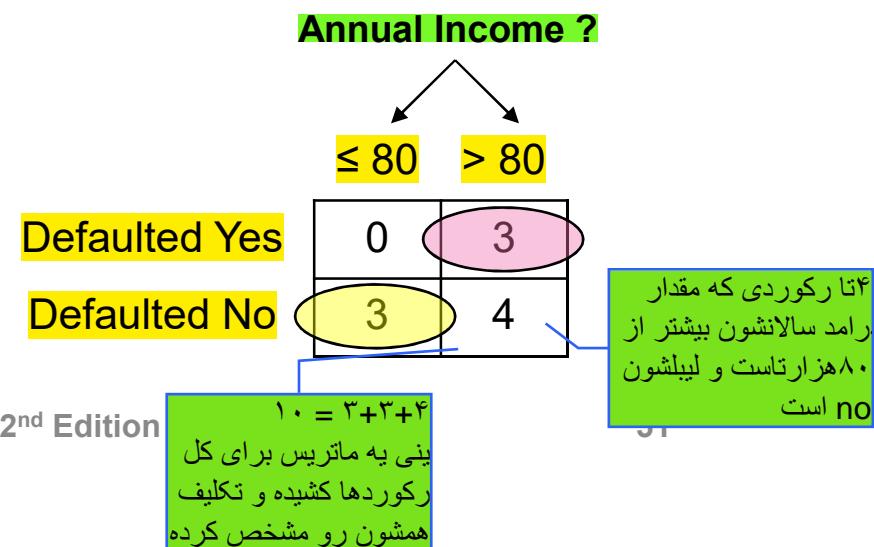
از بین این ۳ تا حالت اونی که کمترین  
جینی را میده انتخاب میکنیم. یعنی  
حالات multi way split

جینی تک تک این حالت هارو بدست میاریم، جینی پرنت یا والد هم  
که داریم اونی را انتخاب میکنیم که بیشترین گین را بده بمون

# Continuous Attributes: Computing Gini Index

- Use Binary Decisions based on one value
- Several Choices for the splitting value
  - Number of possible splitting values = Number of distinct values
- Each splitting value has a count matrix associated with it
  - Class counts in each of the partitions,  $A \leq v$  and  $A > v$
- Simple method to choose best  $v$ 
  - For each  $v$ , scan the database to gather count matrix and compute its Gini index
  - Computationally Inefficient! Repetition of work.

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



# Continuous Attributes: Computing Gini Index...

- For efficient computation: for each attribute,
  - Sort the attribute on values
  - Linearly scan these values, each time updating the count matrix and computing gini index
  - Choose the split position that has the least gini index

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

# Continuous Attributes: Computing Gini Index...

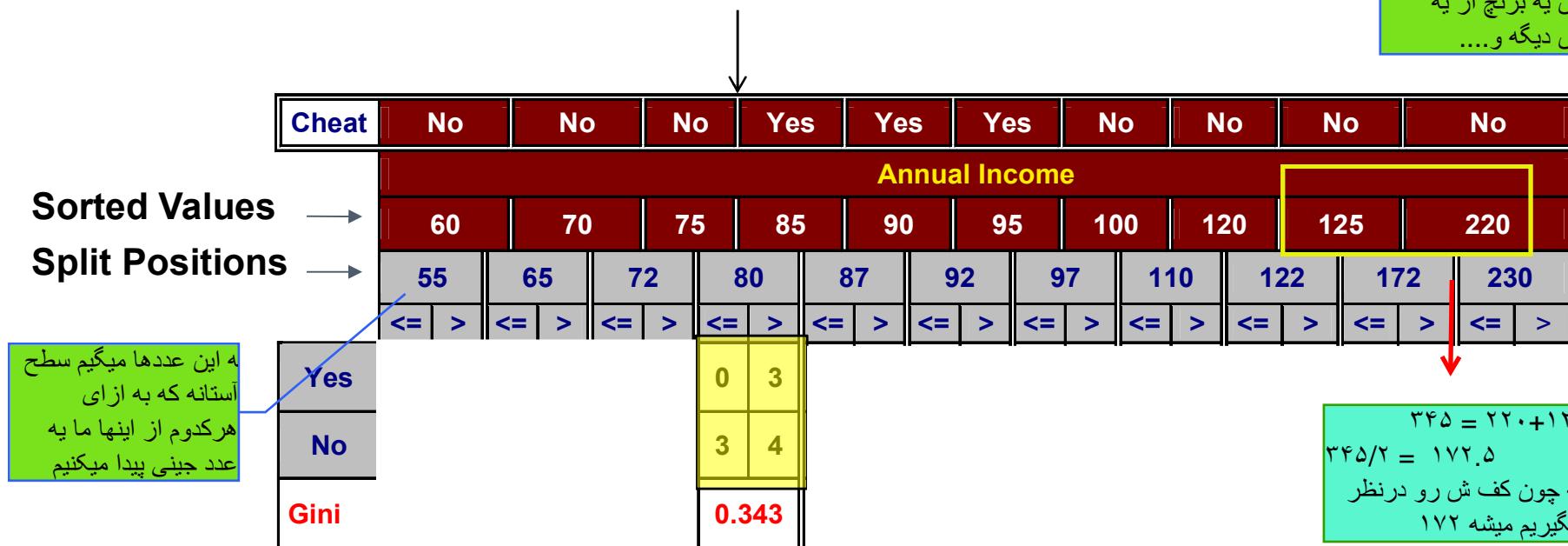
- For efficient computation: for each attribute,
  - Sort the attribute on values
  - Linearly scan these values, each time updating the count matrix and computing gini index
  - Choose the split position that has the least gini index

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

# Continuous Attributes: Computing Gini Index...

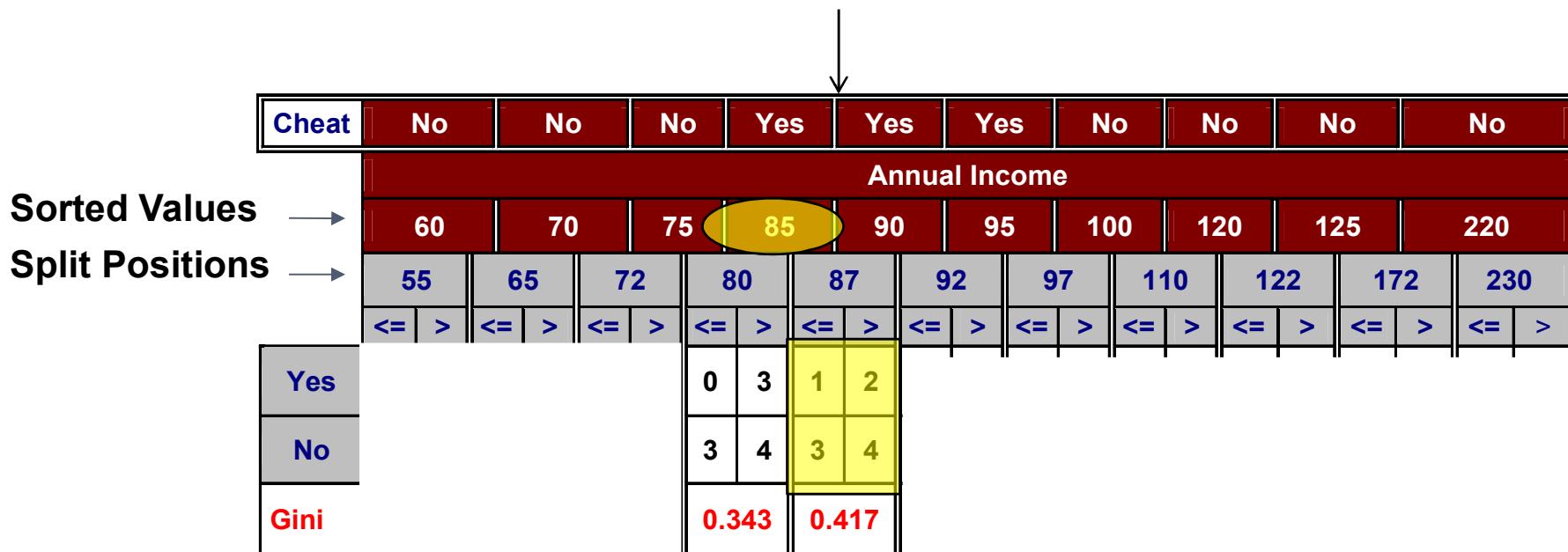
- For efficient computation: for each attribute,
  - Sort the attribute on values
  - Linearly scan these values, each time updating the count matrix and computing gini index
  - Choose the split position that has the least gini index

کم کردن عدم قطعیت  
افزایش قطعیت مثلاً یه  
برنج بشه واقعاً از یه  
کلاس یه برنج از یه  
کلاس دیگه و....



# Continuous Attributes: Computing Gini Index...

- For efficient computation: for each attribute,
  - Sort the attribute on values
  - Linearly scan these values, each time updating the count matrix and computing gini index
  - Choose the split position that has the least gini index



# Continuous Attributes: Computing Gini Index...

- For efficient computation: for each attribute,
  - Sort the attribute on values
  - Linearly scan these values, each time updating the count matrix and computing gini index
  - Choose the split position that has the least gini index

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

# Measure of Impurity: Entropy

راجع به عدم قطعیت  
دارایم حرف میزندیم  
فراش انتروپی به معنای  
فراش عدم قطعیت است

- Entropy at a given node  $t$

$$\text{Entropy} = - \sum_{i=0}^{c-1} p_i(t) \log_2 p_i(t)$$

Where  $p_i(t)$  is the frequency of class  $i$  at node  $t$ , and  $c$  is the total number of classes

- ◆ Maximum of  $\log_2 c$  when records are equally distributed among all classes, implying the least beneficial situation for classification
  - ◆ Minimum of 0 when all records belong to one class, implying most beneficial situation for classification
- 
- Entropy based computations are quite similar to the GINI index computations

# Computing Entropy of a Single Node

$$\text{Entropy} = - \sum_{i=0}^{c-1} p_i(t) \log_2 p_i(t)$$

C1	0
C2	6

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

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

فرایش انتروپی که  
معادل افزایش عدم  
قطعیت است و ما  
دوست نداریم (:

C1	1
C2	5

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

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

C1	2
C2	4

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

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

# Computing Information Gain After Splitting

## Information Gain:

$$Gain_{split}$$

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

Parent Node,  $p$  is split into  $k$  partitions (children)

$n_i$  is number of records in child node  $i$

نود پرنت: انتروپی  
قبل از اینکه این  
اتribیوت را انتخاب  
کنیم چقدر بوده؟

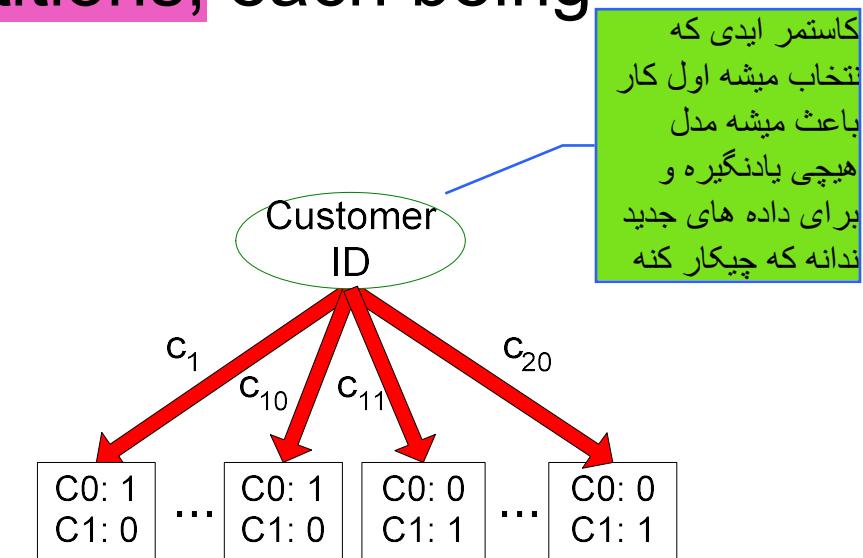
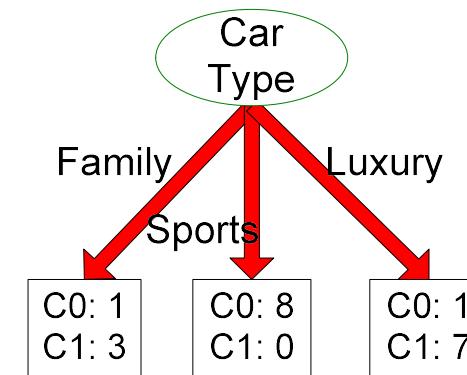
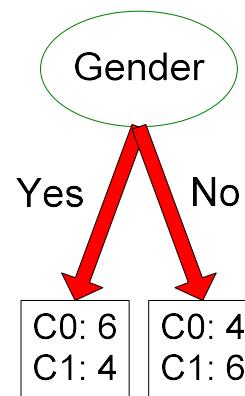
انتروپی چقدر کاهش پیدا کرده؟ یعنی چقدر  
به قطعیت رسیدیم بعد از تقسیم بندی  
براساس اون اtribیوت  
ما میخایم هرچی در درخت پایین تر میریم  
قطعیتمون بیشتر بشه دیگه

انتروپی بعد از  
انتخاب نود برای  
تصمیم گیری  
چون با چندتا  
مجموعه سروکار  
داریم باید وزن  
هر شاخه را حساب  
کنیم

- Choose the split that achieves most reduction (maximizes GAIN)
- Used in the ID3 and C4.5 decision tree algorithms
- Information gain is the mutual information between the class variable and the splitting variable

# Problem with large number of partitions

- Node impurity measures tend to prefer splits that result in large number of partitions, each being small but pure



- Customer ID has highest information gain because entropy for all the children is zero

مشکل اینکه ایدی رو به عنوان اتریبیوت انتخاب کنیم  
چیه؟ داده‌ی جدید که بیاد نمیتوونیم تویی یه دسته  
بگذاریمش چون ایدیش فرق داره میخاد تو یه دسته جدید  
بره نه دسته‌های قبلی پس به دسته بندی داده‌ها کمک  
نمیکنه

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

ینی عدم قطعیت  
داریم چون هر شاخه  
توش یه رکورد قرار  
گرفته

# Gain Ratio

Gain Ratio:

یه شاخصی از تعداد  
شاخه ها اضافه  
میکنیم

اتریبیوت هایی که باعث میشه داده ها به تعداد  
زیادی از شاخه ها افزایش بشن و در هر شاخه یه  
تعداد برابری کلاس داریم

$$Gain\ Ratio = \frac{Gain_{split}}{Split\ Info}$$

یه نسبتی از گین است

$$Split\ Info = - \sum_{i=1}^k \frac{n_i}{n} \log_2 \frac{n_i}{n}$$

برای کاستمر آیدی  
مقدارش بزرگ میشه  
و باعث کم شدن  
مقدار گین کلی میشه  
بنی جریمه میکنه

Parent Node,  $p$  is split into  $k$  partitions (children)

$n_i$  is number of records in child node  $i$

داره انتروپی شاخه  
رو حساب میکنه

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

# Gain Ratio

- Gain Ratio:

$$Gain\ Ratio = \frac{Gain_{split}}{Split\ Info}$$

$$Split\ Info = \sum_{i=1}^k \frac{n_i}{n} \log_2 \frac{n_i}{n}$$

Parent Node,  $p$  is split into  $k$  partitions (children)

$n_i$  is number of records in child node  $i$

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

$$\text{SplitINFO} = 1.52$$

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

$$\text{SplitINFO} = 0.72$$

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

$$\text{SplitINFO} = 0.97$$

$$\begin{aligned} n1/n &= 8/(8+2+10) = 8/20 \\ n2/n &= 12/20 \\ 8/20 * \log 8/20 + 12/20 * \log 12/20 &= -0.528 - 0.438 = -0.966 \end{aligned}$$

Suppose we have a dataset with 100 examples, where 80 examples belong to class A and 20 belong to class B. Let's split this dataset based on an attribute X that has three possible values: X1, X2, and X3. If we calculate the information gain for each value of X, we get:

- For X1, the dataset splits into 60 examples of class A and 20 examples of class B. The information gain is 0.539.
- For X2, the dataset splits into 10 examples of class A and 10 examples of class B. The information gain is 0.208.
- For X3, the dataset splits into 10 examples of class A and 0 examples of class B. The information gain is 0.206.

Now, let's calculate the split information for each value of X based on the number of examples:

- For X1, the split information is  $1 \log(1) + 3/5 \log(3/5) + 2/5 \log(2/5) = 0.971$ .
- For X2, the split information is  $1 \log(1) + 1 \log(1) = 0$ .
- For X3, the split information is  $1 \log(1) + 0 \log(0) = 0$ .

Finally, we can calculate the gain ratio for each value of X by dividing the information gain by the split information:

- For X1, the gain ratio is  $0.539/0.971 = 0.556$ .
- For X2, the gain ratio is undefined since the split information is 0.
- For X3, the gain ratio is undefined since the split information contains a  $\log(0)$ .

Therefore, based on the gain ratio criterion, we would choose X1 as the attribute to split the dataset.

Suppose we have a dataset with two binary features (A and B) and a binary target variable (Y), where 10 instances have A=0,B=0, 20 instances have A=1,B=0, 15 instances have A=0,B=1, and 5 instances have A=1,B=1. Let's compute the gain ratio for each possible split:

Split on feature A:

Subgroup 1: A=0 (25 instances)

Y=0: 10 instances

Y=1: 15 instances

Subgroup 2: A=1 (25 instances)

Y=0: 20 instances

Y=1: 5 instances

$$\text{Entropy}(Y) = -(30/50)\log_2(30/50) - (20/50)\log_2(20/50) = 0.971$$

$$\text{Split information} = -(25/50)\log_2(25/50) - (25/50)\log_2(25/50) = 1.0$$

$$\text{Information gain} = 0.971 - ((25/50)*(-(10/25)\log_2(10/25) - (15/25)\log_2(15/25)) + (25/50)(-(20/25)\log_2(20/25) - (5/25)\log_2(5/25))) = 0.02$$

$$\text{Gain ratio} = 0.02 / 1.0 = 0.02$$

Split on feature B:

Subgroup 1: B=0 (30 instances)

Y=0: 20 instances

Y=1: 10 instances

Subgroup 2: B=1 (20 instances)

Y=0: 10 instances

Y=1: 10 instances

$$\text{Entropy}(Y) = -(30/50)\log_2(30/50) - (20/50)\log_2(20/50) = 0.971$$

$$\text{Split information} = -(30/50)\log_2(30/50) - (20/50)\log_2(20/50) = 0.971$$

$$\text{Information gain} = 0.971 - ((30/50)*(-(20/30)\log_2(20/30) - (10/30)\log_2(10/30)) + (20/50)(-(10/20)\log_2(10/20) - (10/20)\log_2(10/20))) = 0.019$$

$$\text{Gain ratio} = 0.019 / 0.971 = 0.0196$$

Based on the gain ratios, we would choose to split on feature A because it has a slightly higher gain ratio than feature B.

Suppose we have a dataset with three features (A, B, and C) and a binary target variable (Y), where 15 instances have A=0,B=0,C=0, 10 instances have A=1,B=0,C=0, 5 instances have A=0,B=1,C=0, 5 instances have A=0,B=0,C=1, 5 instances have A=1,B=1,C=0, and 20 instances have A=1,B=0,C=1. Let's compute the gain ratio for each possible split:

Split on feature A:

Subgroup 1: A=0 (25 instances)

Y=0: 15 instances

Y=1: 10 instances

Subgroup 2: A=1 (30 instances)

Y=0: 25 instances

Y=1: 5 instances

$$\text{Entropy}(Y) = -(30/55) \cdot \log_2(30/55) - (25/55) \cdot \log_2(25/55) = 0.961$$

$$\text{Split information} = -(25/55) \cdot \log_2(25/55) - (30/55) \cdot \log_2(30/55) = 0.994$$

$$\text{Information gain} = 0.961 - ((25/55) \cdot (-15/25) \cdot \log_2(15/25) - (10/25) \cdot \log_2(10/25)) + (30/55) \cdot (-25/30) \cdot \log_2(25/30) - (5/30) \cdot \log_2(5/30))) = 0.132$$

$$\text{Gain ratio} = 0.132 / 0.994 = 0.133$$

Split on feature B:

Subgroup 1: B=0 (35 instances)

Y=0: 25 instances

Y=1: 10 instances

Subgroup 2: B=1 (20 instances)

Y=0: 15 instances

Y=1: 5 instances

$$\text{Entropy}(Y) = -(30/55) \cdot \log_2(30/55) - (25/55) \cdot \log_2(25/55) = 0.961$$

$$\text{Split information} = -(35/55) \cdot \log_2(35/55) - (20/55) \cdot \log_2(20/55) = 0.928$$

$$\text{Information gain} = 0.961 - ((35/55) \cdot (-25/35) \cdot \log_2(25/35) - (10/35) \cdot \log_2(10/35)) + (20/55) \cdot (-15/20) \cdot \log_2(15/20) - (5/20) \cdot \log_2(5/20))) = 0.114$$

$$\text{Gain ratio} = 0.114 / 0.928 = 0.123$$

Split on feature C:

Subgroup 1: C=0 (35 instances)

Y=0: 25 instances

Y=1: 10 instances

Subgroup 2: C=1 (20 instances)

Y=0: 15 instances

Y=1: 5 instances

$$\text{Entropy}(Y) = -(30/55) \cdot \log_2(30/55) - (25/55) \cdot \log_2(25/55) = 0.961$$

$$\text{Split information} = -(35/55) \cdot \log_2(35/55) - (20/55) \cdot \log_2(20/55) = 0.928$$

$$\text{Information gain} = 0.961 - ((35/55) \cdot (-25/35) \cdot \log_2(25/35) - (10/35) \cdot \log_2(10/35)) + (20/55) \cdot (-15/20) \cdot \log_2(15/20) - (5/20) \cdot \log_2(5/20))) = 0.114$$

$$\text{Gain ratio} = 0.114 / 0.928 = 0.123$$

Based on the gain ratios, we would choose to split on feature A because it has the highest gain ratio.

Sure, here are some numeric examples of splitting decision trees based on entropy:

Suppose we have a dataset with two binary features (A and B) and a binary target variable (Y), where 10 instances have A=0,B=0, 20 instances have A=1,B=0, 15 instances have A=0,B=1, and 5 instances have A=1,B=1. Let's compute the entropy for each possible split:

- Split on feature A:

- Subgroup 1: A=0 (25 instances)

- Y=0: 10 instances

- Y=1: 15 instances

- Subgroup 2: A=1 (25 instances)

- Y=0: 20 instances

- Y=1: 5 instances

- Entropy(Y) =  $-(30/50)\log_2(30/50) - (20/50)\log_2(20/50) = 0.971$

- Information gain = Entropy(Y) -  $((25/50)*(-(10/25)\log_2(10/25) - (15/25)\log_2(15/25)) + (25/50)*(-(20/25)\log_2(20/25) - (5/25)\log_2(5/25)) = 0.321 - 0.02 = 0.301$

- Split on feature B:

- Subgroup 1: B=0 (30 instances)

- Y=0: 20 instances

- Y=1: 10 instances

- Subgroup 2: B=1 (20 instances)

- Y=0: 10 instances

- Y=1: 10 instances

- Entropy(Y) =  $-(30/50)\log_2(30/50) - (20/50)\log_2(20/50) = 0.971$

- Information gain = Entropy(Y) -  $((30/50)*(-(20/30)\log_2(20/30) - (10/30)\log_2(10/30)) + (20/50)*(-(10/20)\log_2(10/20) - (10/20)\log_2(10/20)) = 0.321 - 0.019 = 0.302$

Based on the information gains, we would choose to split on feature B because it has a slightly higher information gain than feature A.

Note that for entropy-based splitting, the choice of split can be affected by the size of the subgroups, while this is not an issue for gain ratio-based splitting.

## Measure of Impurity: Classification Error

---

---

- Classification error at a node  $t$

$$Error(t) = 1 - \max_i[p_i(t)]$$

- Maximum of  $1 - 1/c$  when records are equally distributed among all classes, implying the least interesting situation
- Minimum of 0 when all records belong to one class, implying the most interesting situation

# Computing Error of a Single Node

$$Error(t) = 1 - \max_i[p_i(t)]$$

C1	0
C2	6

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

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

C1	1
C2	5

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

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

ارور ناشی  
از افزایش  
عدم قطعیت  
اره زیادتر  
میشه

C1	2
C2	4

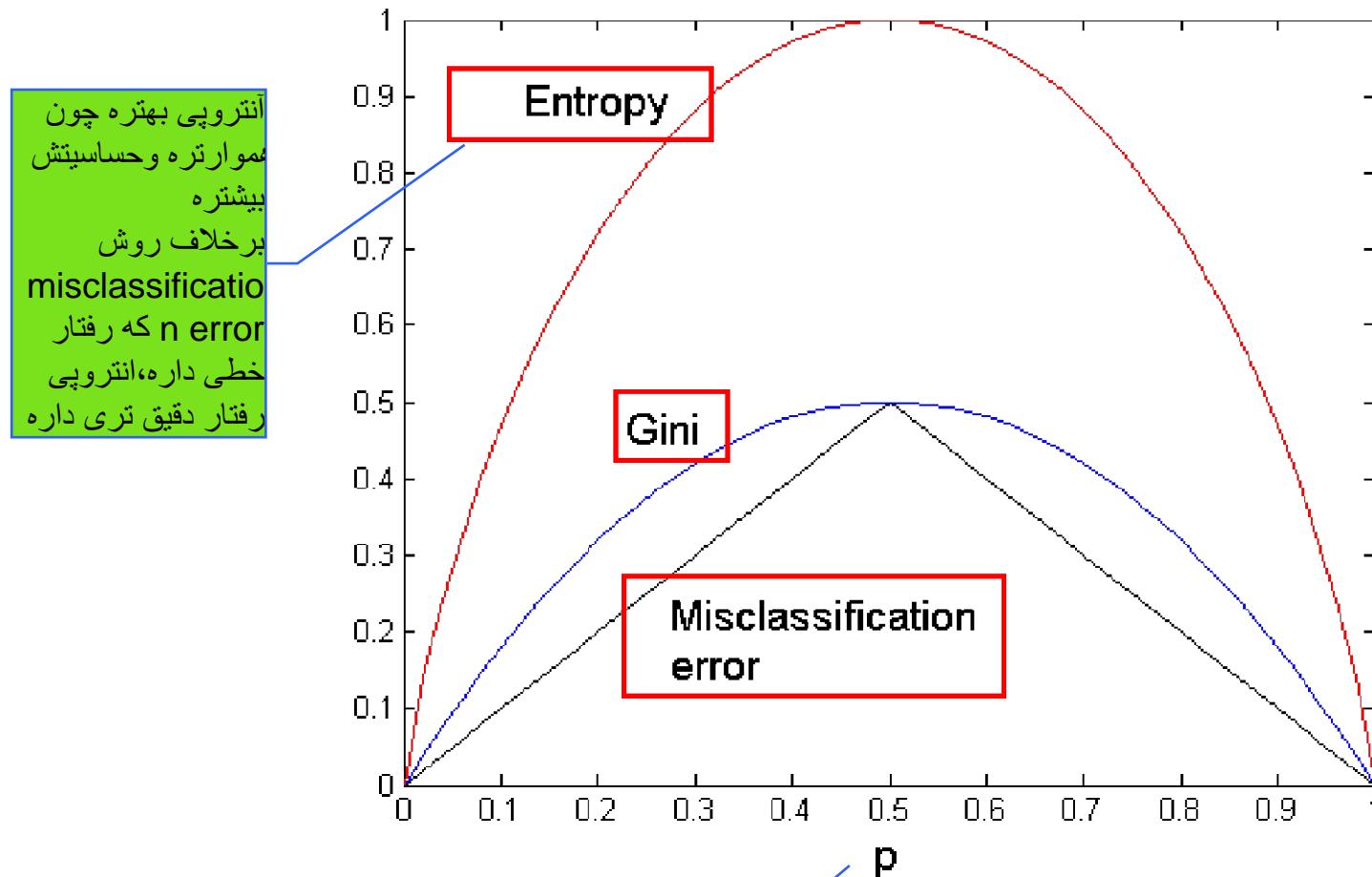
$$P(C1) = 2/6 \quad P(C2) = 4/6$$

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



# Comparison among Impurity Measures

For a 2-class problem:



### 1. Gini index:

The Gini index is a measure of the impurity of a set of instances. It is defined as the probability of misclassifying a random instance in the dataset if it were randomly labeled according to the class distribution in the dataset. The Gini index ranges from 0 to 1, where 0 indicates a pure node (all instances belong to the same class) and 1 indicates a completely impure node (instances are equally distributed among all classes). The Gini index is faster to compute than entropy and is preferred when the class distribution is balanced.

### 2. Entropy:

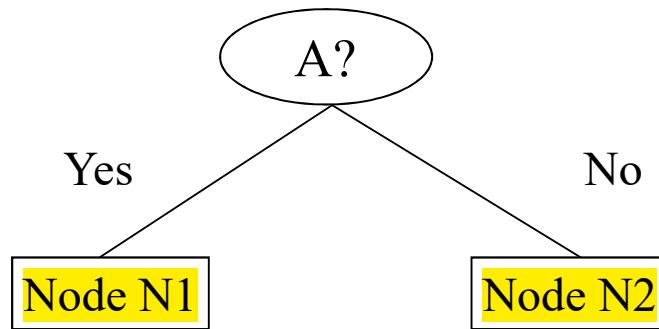
Entropy is a measure of the impurity of a set of instances. It is defined as the sum of the negative logarithms of the class probabilities. The entropy ranges from 0 to 1, where 0 indicates a pure node and 1 indicates a completely impure node. Entropy is slower to compute than the Gini index but is preferred when the class distribution is imbalanced.

### 3. Classification error:

Classification error is a measure of the impurity of a set of instances. It is defined as the proportion of instances in a node that do not belong to the majority class. The classification error ranges from 0 to 1, where 0 indicates a pure node and 1 indicates a completely impure node. Classification error is the simplest impurity measure to compute but is less sensitive to changes in the class distribution than the Gini index and entropy.

In summary, the choice of impurity measure depends on the characteristics of the dataset, such as the class distribution and the size of the dataset. The Gini index is preferred when the class distribution is balanced, while entropy is preferred when the class distribution is imbalanced. Classification error is the simplest impurity measure to compute but is less sensitive to changes in the class distribution than the other measures.

# Misclassification Error vs Gini Index



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

**Gini(N1)**

$$= 1 - (3/3)^2 - (0/3)^2 \\ = 0$$

**Gini(N2)**

$$= 1 - (4/7)^2 - (3/7)^2 \\ = 0.489$$

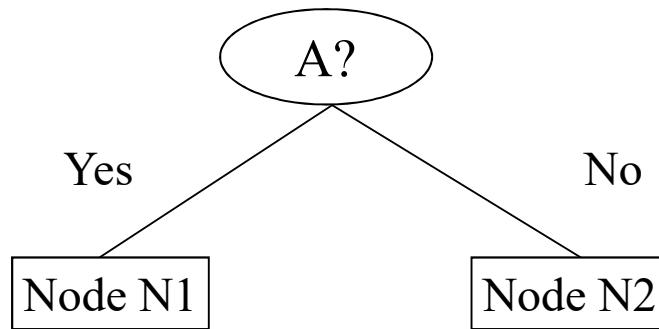
	<b>N1</b>	<b>N2</b>
C1	3	4
C2	0	3
<b>Gini=0.342</b>		

**Gini(Children)**

$$= 3/10 * 0 \\ + 7/10 * 0.489 \\ = 0.342$$

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

# Misclassification Error vs Gini Index



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

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

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

**Misclassification error for all three cases = 0.3 !**

# Decision Tree Based Classification

## ● Advantages:

- Relatively inexpensive to construct
- Extremely fast at classifying unknown records
- Easy to interpret for small-sized trees
- Robust to noise (especially when methods to avoid overfitting are employed)
- Can easily handle redundant attributes
- Can easily handle irrelevant attributes (unless the attributes are interacting)

شناسایی اتریبیوت  
های نامرتب

یه سری اتریبیوت هستند  
که برای تصمیم گیری  
برآشون باید باهم بشون  
نگاه کرد جداجدا که نگاه  
کنیم نمیشه تصمیم گرفت  
برآشون به اینا  
interacting  
میگن attributes

## ● Disadvantages: .

- Due to the greedy nature of splitting criterion, interacting attributes (that can distinguish between classes together but not individually) may be passed over in favor of other attributed that are less discriminating.
- Each decision boundary involves only a single attribute

ویژگی های خوب درخت های تصمیم:

۱. خیلی راحت ساخته میشوند
۲. خیلی سریع آموزش میبینند

2/1/2021

۳. تفسیر پذیر هستند به خصوص درخت های کوچکتر و خصوصا برای افرادی که با ماشین لرنینگ آشنایی ندارند

۴. نسبت به نویز نسبتاً رباتست و مقاوم است چون با خود رکوردها کار نمیکنیم و بالاحتمال هاشون کار میکنیم

۵. جاهایی که اتریبیوت های اضافه داریم میتوانه بگه این اضافه است و کاری بش نداشته باش به دلیل شباهت رفتارهای اتریبیوت ها

پس برای شناسایی اتریبیوت های اضافه مفید است

how decision trees can handle redundant attributes?

Decision trees can handle redundant attributes in different ways. One common approach is to use feature selection techniques to identify and remove redundant attributes before building the tree. This can help reduce the complexity of the tree and improve its accuracy.

Another approach is to use pruning techniques after building the tree to eliminate redundant branches or subtrees that do not contribute significantly to the classification performance. Pruning can help simplify the tree and reduce overfitting, which may occur when the tree is too complex and captures noise or irrelevant features in the data.

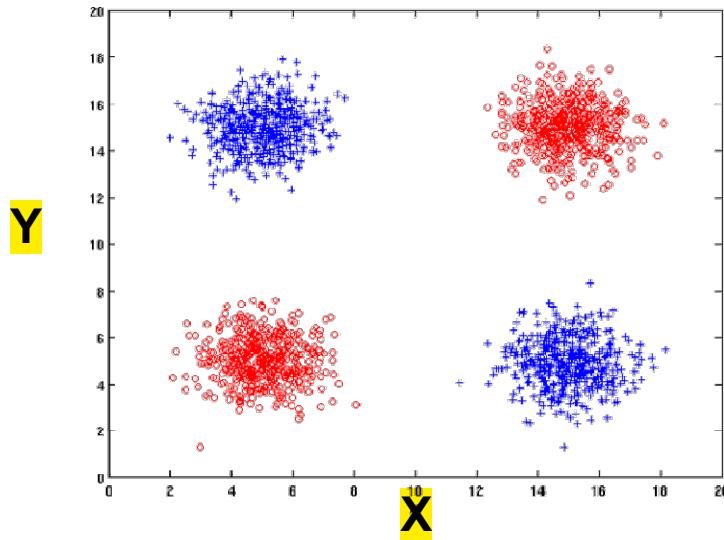
Moreover, some decision tree algorithms such as C4.5 or CART use a measure of information gain or impurity reduction to select the best split at each node. These measures take into account both the relevance and redundancy of the attributes, and aim to find splits that maximize the overall accuracy of the tree while minimizing redundancy.

Single attribute-based decision boundaries refer to the use of a single feature or attribute to make decisions in a decision tree. In other words, at each node in the tree, only one feature is used to split the data into two or more groups based on some threshold value.

For example, consider a decision tree that predicts whether or not a customer will buy a product based on their age. The first node of the tree may split the data into two groups: customers younger than a certain age and customers older than that age. This is an example of a single attribute-based decision boundary.

Single attribute-based decision boundaries are commonly used in decision trees because they are easy to interpret and visualize. However, as I mentioned earlier, they have some limitations in terms of their ability to capture complex relationships and handle missing data.

# Handling interactions



+ : 1000 instances

o : 1000 instances

Entropy (X) : 0.99

Entropy (Y) : 0.99

## Disadvantages:

- Decision trees are prone to overfitting, which means the model may perform well on the training data but poorly on the test data.
- Small changes in the data can lead to large changes in the decision tree.
- Decision trees can be biased towards features with many levels or features that provide more information.
- They are not suitable for learning complex patterns or relationships in data.
- Decision trees can be unstable because small variations in the data can result in a completely different tree.

Let's say we have a dataset that includes information about customers at a car dealership. The dataset includes the following attributes:

Age: the age of the customer in years

Income: the annual income of the customer in thousands of dollars

Gender: the gender of the customer (either male or female)

Car type: the type of car the customer is interested in (e.g., sedan, SUV, truck)

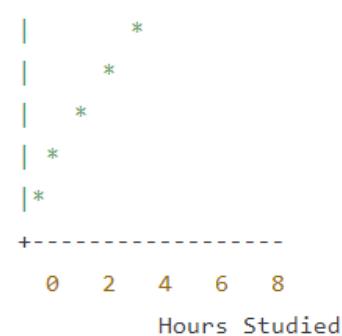
The outcome variable is whether or not the customer ultimately purchases a car from the dealership.

One way that age and income may interact with each other is that older customers with higher incomes may be more likely to purchase expensive cars like SUVs or luxury sedans. To capture this interaction in a decision tree classifier, we could create a new feature called "age times income" that represents the product of age and income.

We could then build a decision tree that splits the data based on this new feature. For example, the decision tree might split the data into two groups: one group where "age times income" is greater than a certain value (indicating older customers with higher incomes), and another group where it is less than that value (indicating younger customers with lower incomes).

Within each of these groups, the decision tree could further split the data based on other attributes like gender or car type. For example, within the group of older, high-income customers, the decision tree might split the data based on gender (i.e., male vs. female) to see if there are any gender differences in car preferences.

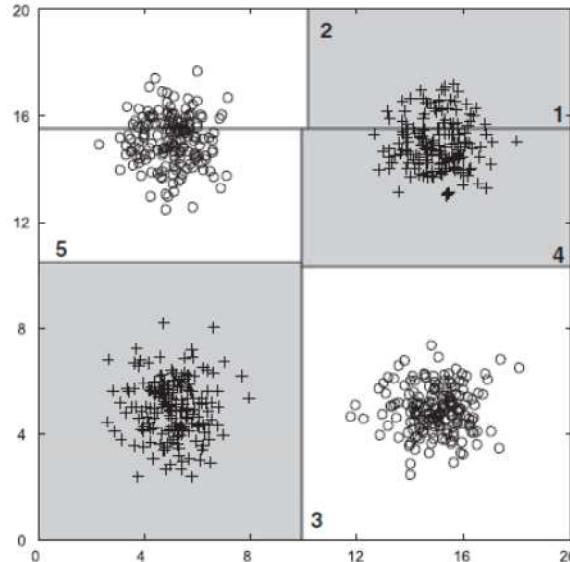
By incorporating interactions between attributes in this way, decision tree classifiers can more accurately model complex relationships between variables and make more accurate predictions about the outcome variable.



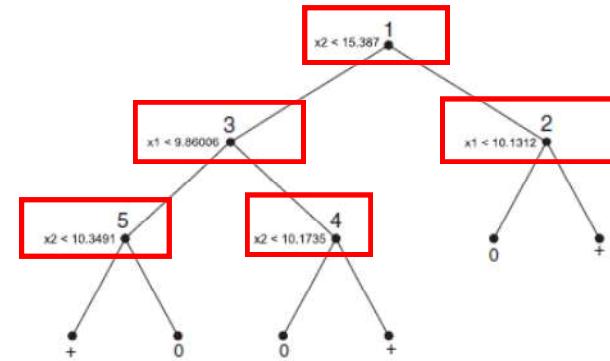
In this scatter plot, we have two attributes on the x-axis and y-axis: "Hours Studied" and "Exam Score". The data points represent students' exam scores based on how many hours they studied.

There is clearly a positive correlation between hours studied and exam score. However, there is also an interaction between these attributes - the slope of the line changes at different points along the x-axis. For example, for students who studied less than 4 hours, there is a steep increase in exam score as the hours studied increases. However, for students who studied more than 4 hours, the increase in exam score is more gradual. This suggests that the relationship between hours studied and exam score is not linear and there may be other factors at play that impact exam performance.

# Handling interactions



(a) Decision boundary for tree with 6 leaf nodes.



(b) Decision tree with 6 leaf nodes.

Figure 3.28. Decision tree with 6 leaf nodes using X and Y as attributes. Splits have been numbered from 1 to 5 in order of other occurrence in the tree.

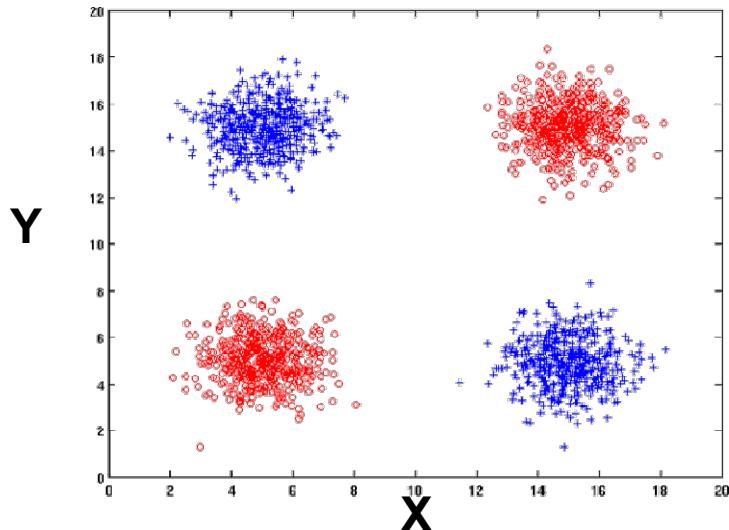
let's consider an example where we are trying to build a decision tree to predict whether or not a customer will purchase a product based on their age and income. Suppose we decide to use a single attribute-based decision boundary at the root node of the tree based on age alone.

If we split the data into two groups based on a threshold age of 30, then all customers younger than 30 would be in one group and all customers older than 30 would be in the other group. Let's suppose that the younger group has a higher proportion of non-purchasers and the older group has a higher proportion of purchasers.

However, if we had used both age and income as features to split the data, we may have found that for customers under 30 with high incomes, they are more likely to purchase the product while for customers over 30 with low incomes, they are less likely to purchase the product.

In this case, using a single attribute-based decision boundary based on age alone would not capture the true relationship between age, income, and purchasing behavior, resulting in a less accurate decision tree model.

# Handling interactions given irrelevant attributes



+ : 1000 instances

o : 1000 instances

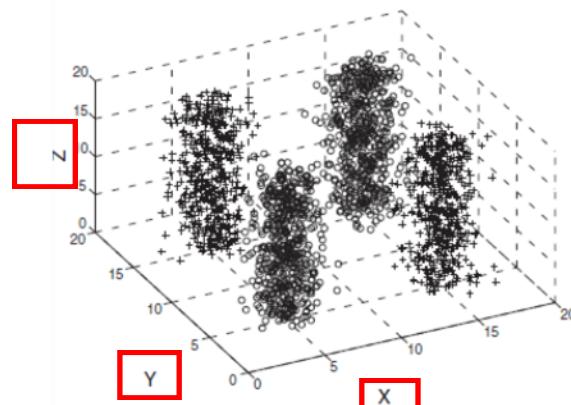
Adding Z as a noisy attribute generated from a uniform distribution

Entropy (X) : 0.99

Entropy (Y) : 0.99

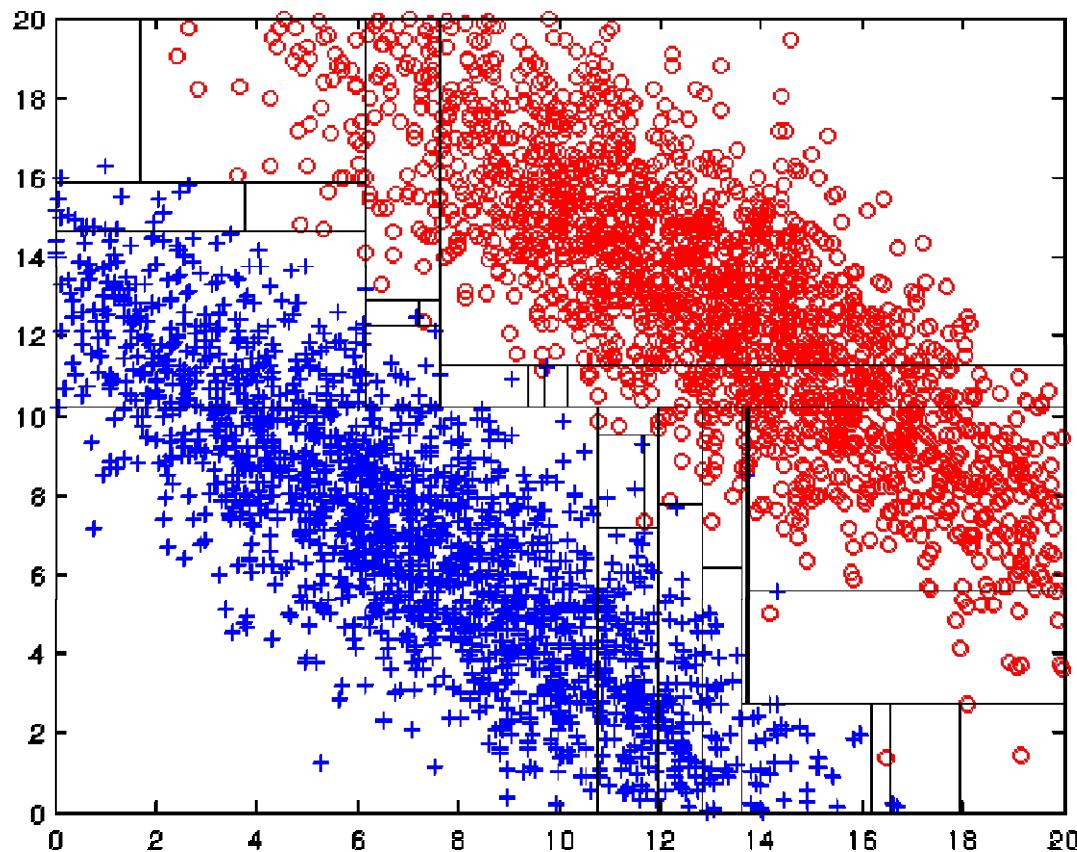
Entropy (Z) : 0.98

Attribute Z will be chosen for splitting!



(a) Three-dimensional data with attributes  $X$ ,  $Y$ , and  $Z$ .

# Limitations of single attribute-based decision boundaries



Both **positive (+)** and **negative (o)** classes generated from skewed Gaussians with centers at (8,8) and (12,12) respectively.

دوتا کلاس ابی و قرمز داریم که کلاس ابی هایکس و وای هاشون کم باشه کلاس قرمزها ایکس و وای هاشون زیاد باشه اگه این مقادیر را به درخت بدیم چیکار میکنه؟ باید دنبال حد استانه ای بگرده که این حد ها را بیشتر از هم متمایز کنه بعد که حد استانه مشخص شد داده هارو در شاخه های درخت پخش میکنه

مشکل درخت تو این شکل چیه؟

باید بیست بار یا سی بار از دوتا اتریبوتوی که داریم با حداستانه های مختلف استفاده کنه تا بتوانه شبکه بندی و گروه بندی کنه داده هارا نحوه ی نگاه کردنش و شبکه بندی کردنش به این شکل به درد این مساله نمیخوره

Generalization error, also known as out-of-sample error, is a measure of how well a machine learning model performs on unseen data. In other words, it measures the ability of the model to generalize beyond the training data and make accurate predictions on new, previously unseen data.

Generalization error occurs when a model is trained too well on the training data, leading it to be overfit and not able to generalize to new data. This can happen if the model is too complex or if there is not enough diverse training data. On the other hand, if a model is underfit, it may not perform well even on the training data, let alone on unseen data.

The goal of machine learning is to minimize the generalization error by finding the right balance between model complexity and training data size and diversity. This is typically achieved through techniques such as cross-validation and regularization.

# **OVERFITTING MODEL SELECTION EVALUATION**

# Classification Errors

- **Training errors:** Errors committed on the **training set**

چند نوع خطا داریم  
خطای داده های آموزشی  
مدل من پس از ساخته  
شدن روی خود داده های  
آموزشی چه خطایی  
داره؟

- **Test errors:** Errors committed on the **test set**

خطاهای آموزشی: خطاهای مرتب در مجموعه آموزشی  
خطاهای تست: خطاهای مرتب شده در مجموعه تست  
خطاهای تعمیم: خطای مورد انتظار یک مدل در انتخاب  
تصادفی رکوردها از همان توزیع

- **Generalization errors:** Expected error of a model over random selection of records from same distribution

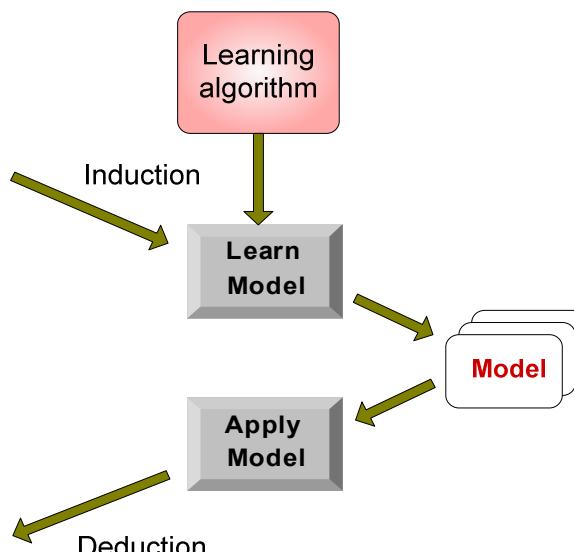
خطای عمومی یا عام  
ما دنبال این هستیم که یه  
تخمینی از خطای عام پیدا  
کنیم  
خطای عام ینی به طور  
متوسط مدل ما چقدر میتوانه  
وی کلاسیفیکیشن خوب عمل  
کنه؟

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

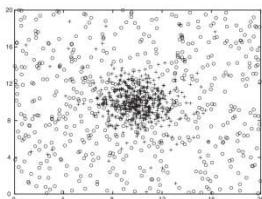
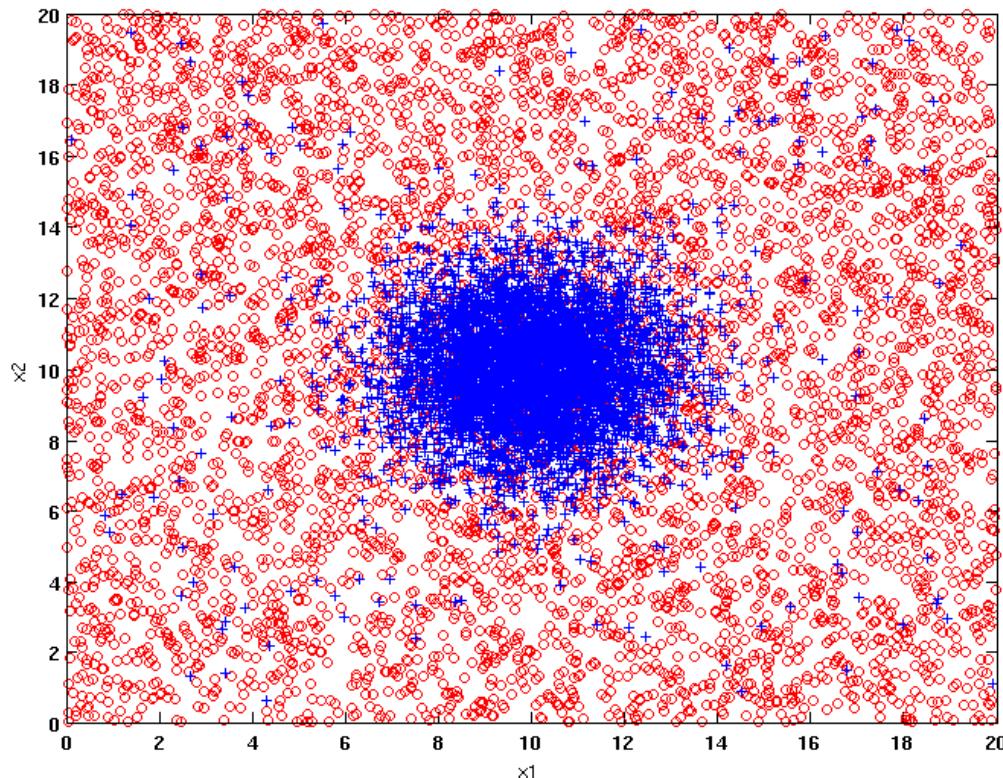
Training Set

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

Test Set



# Example Data Set



**Two class problem:**

**+** : 5400 instances

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

- 400 noisy instances added

**o** : 5400 instances

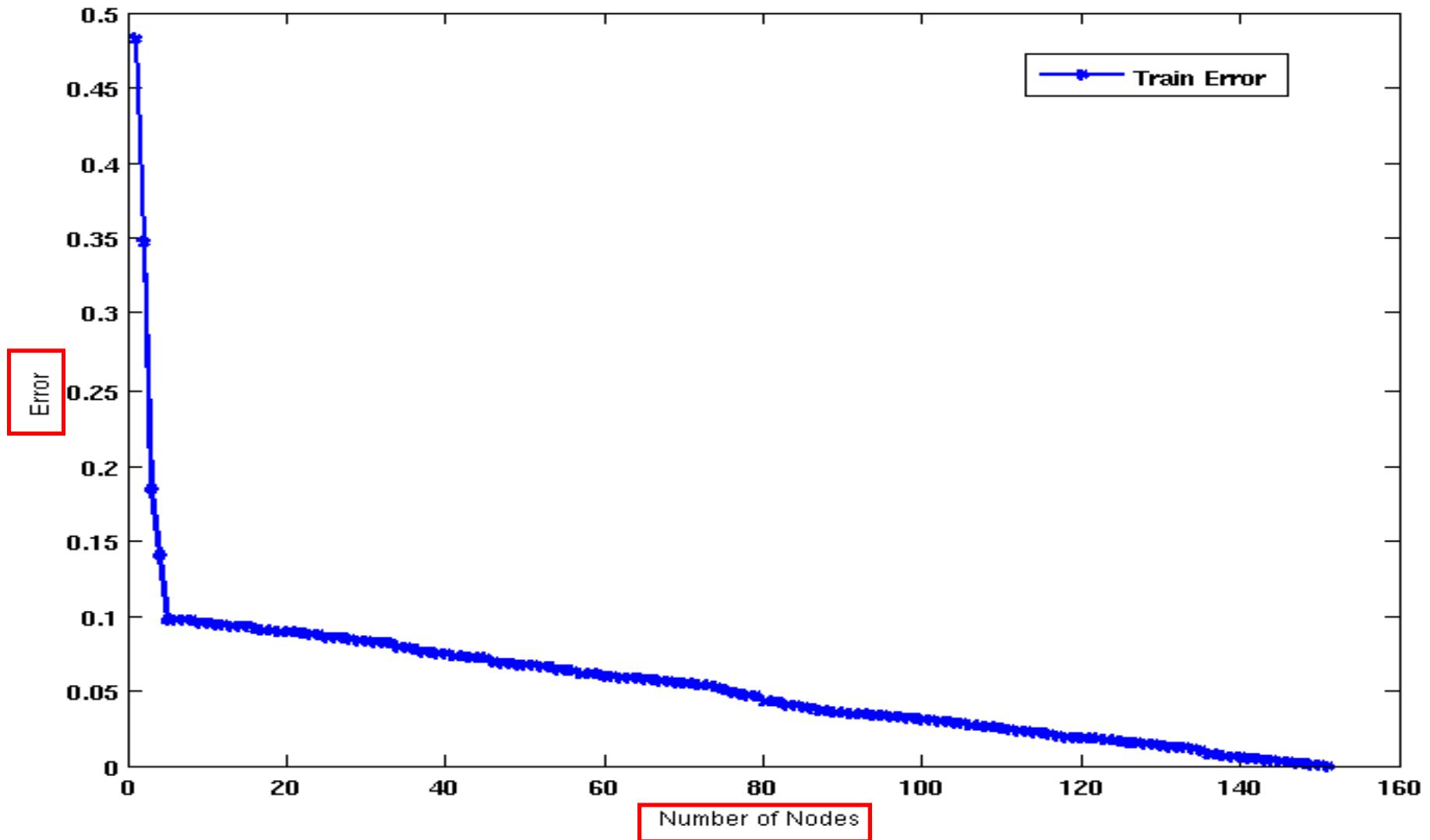
- Generated from a uniform distribution

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

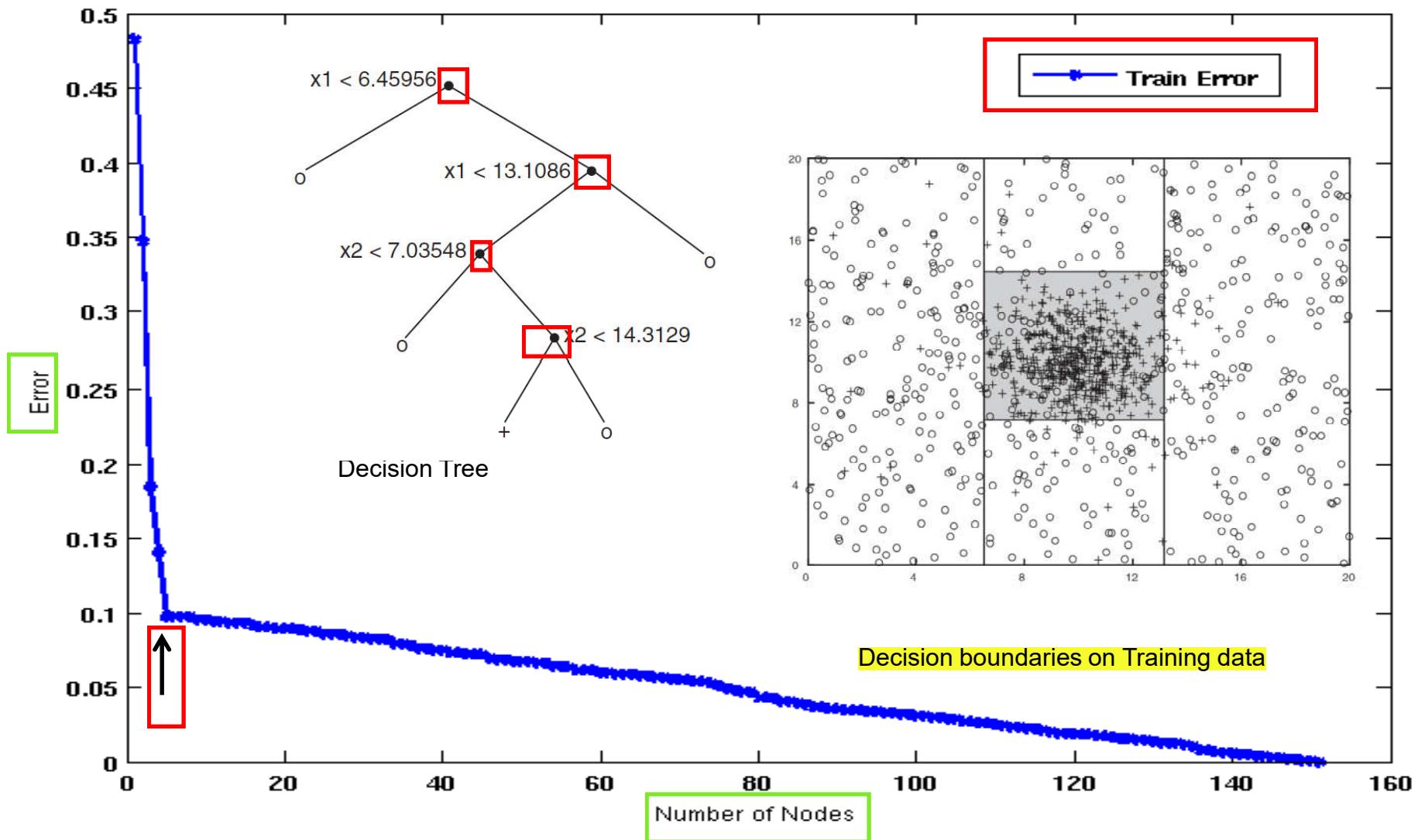
# Increasing number of nodes in Decision Trees

---

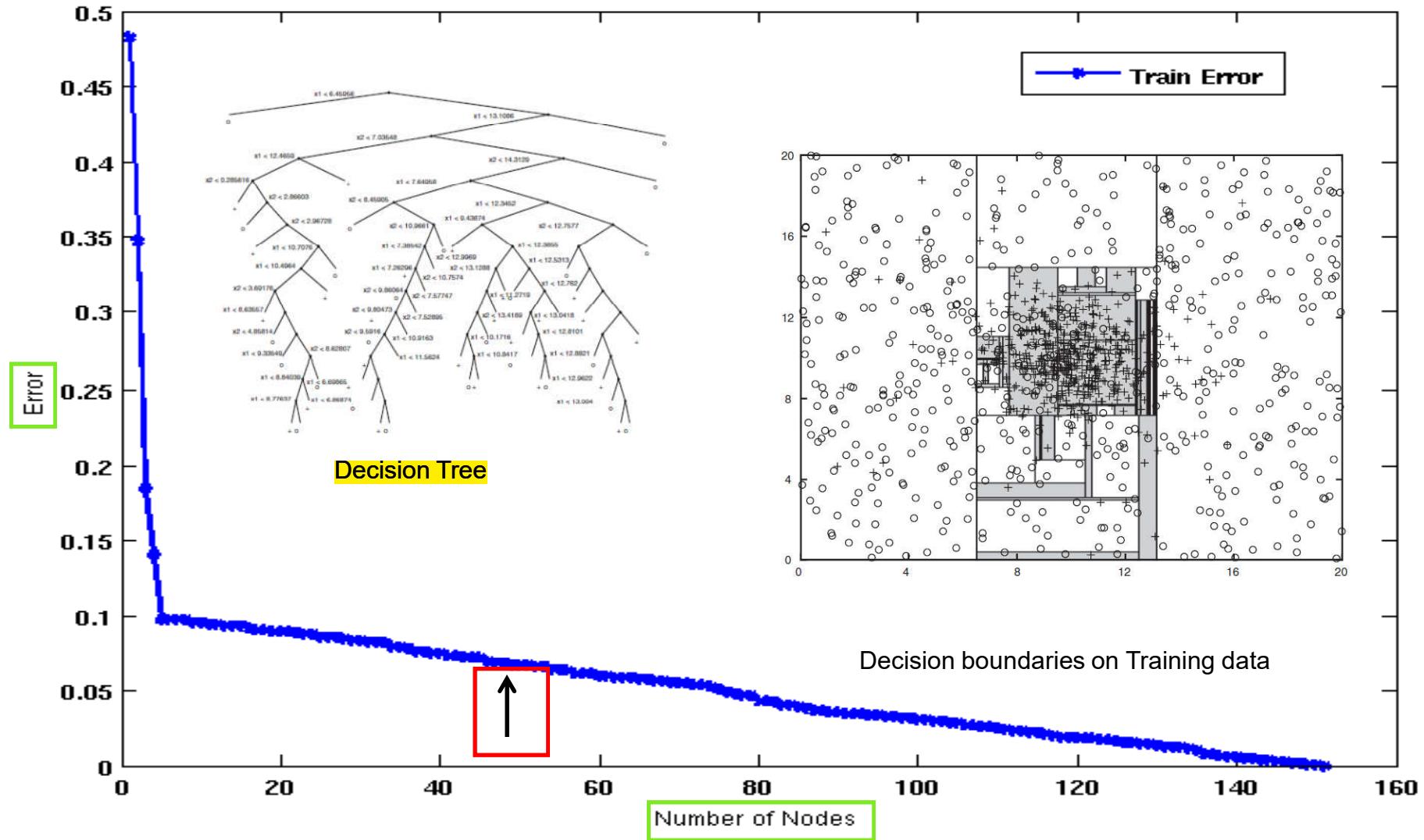
---



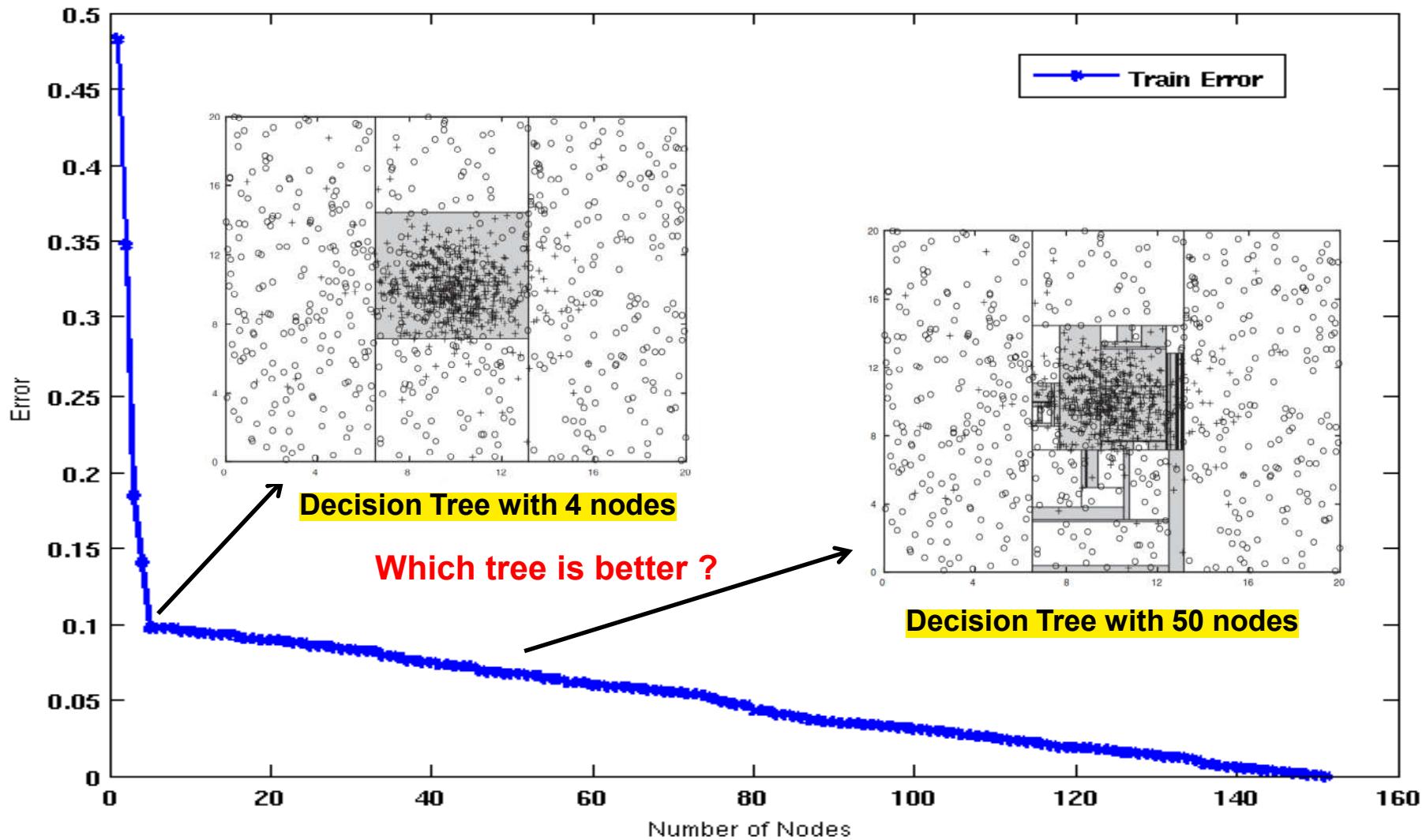
# Decision Tree with 4 nodes



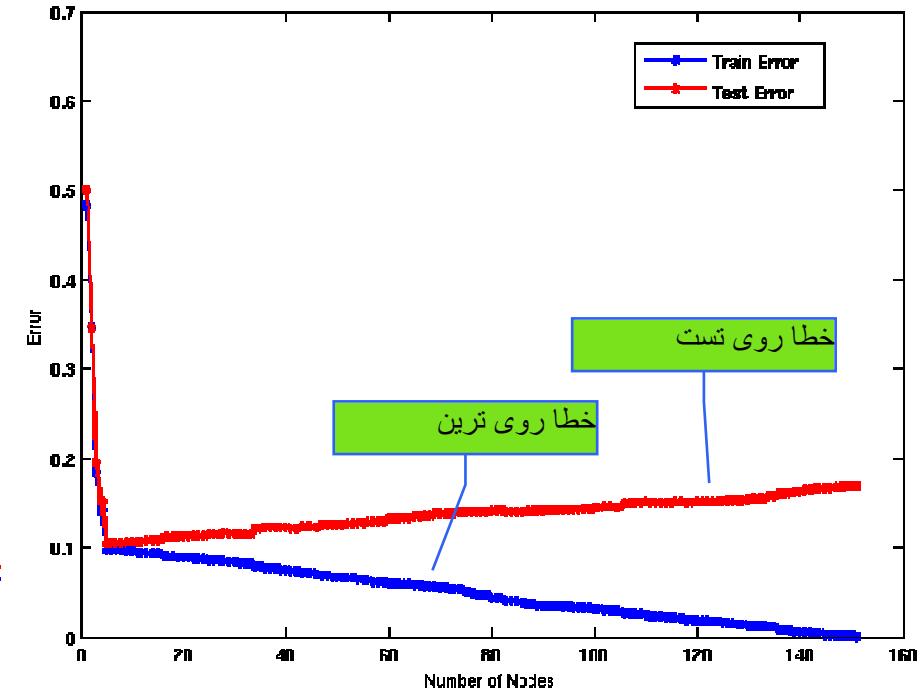
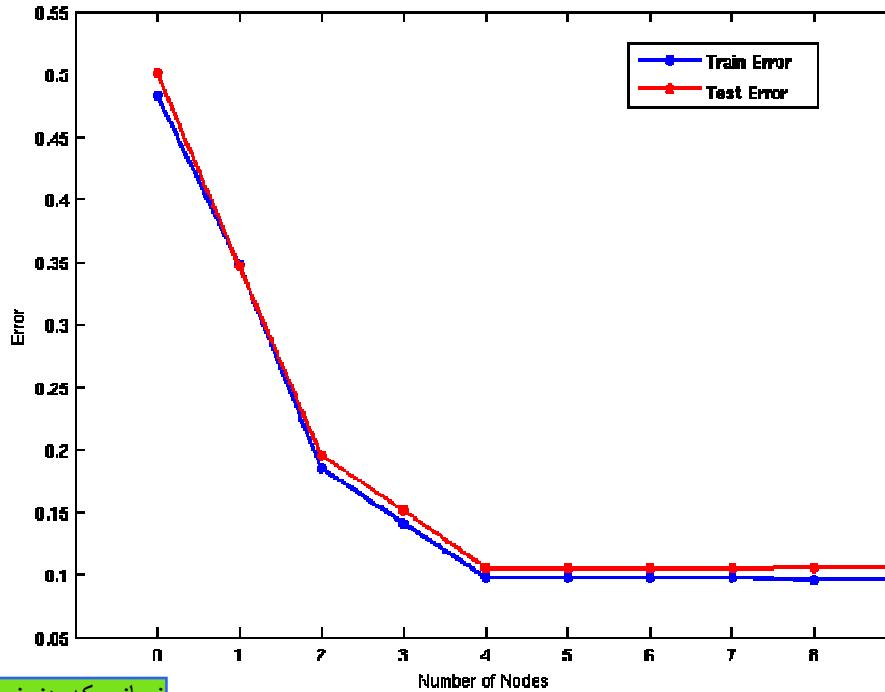
# Decision Tree with 50 nodes



# Which tree is better?



# Model Underfitting and Overfitting



زمانی که هنوز تقریباً  
چیزی پیدا نگرفتیم از داد  
ها هم خطای اموزشمن  
بالاست هم تست

- As the model becomes more and more complex, test errors can start increasing even though training error may be decreasing

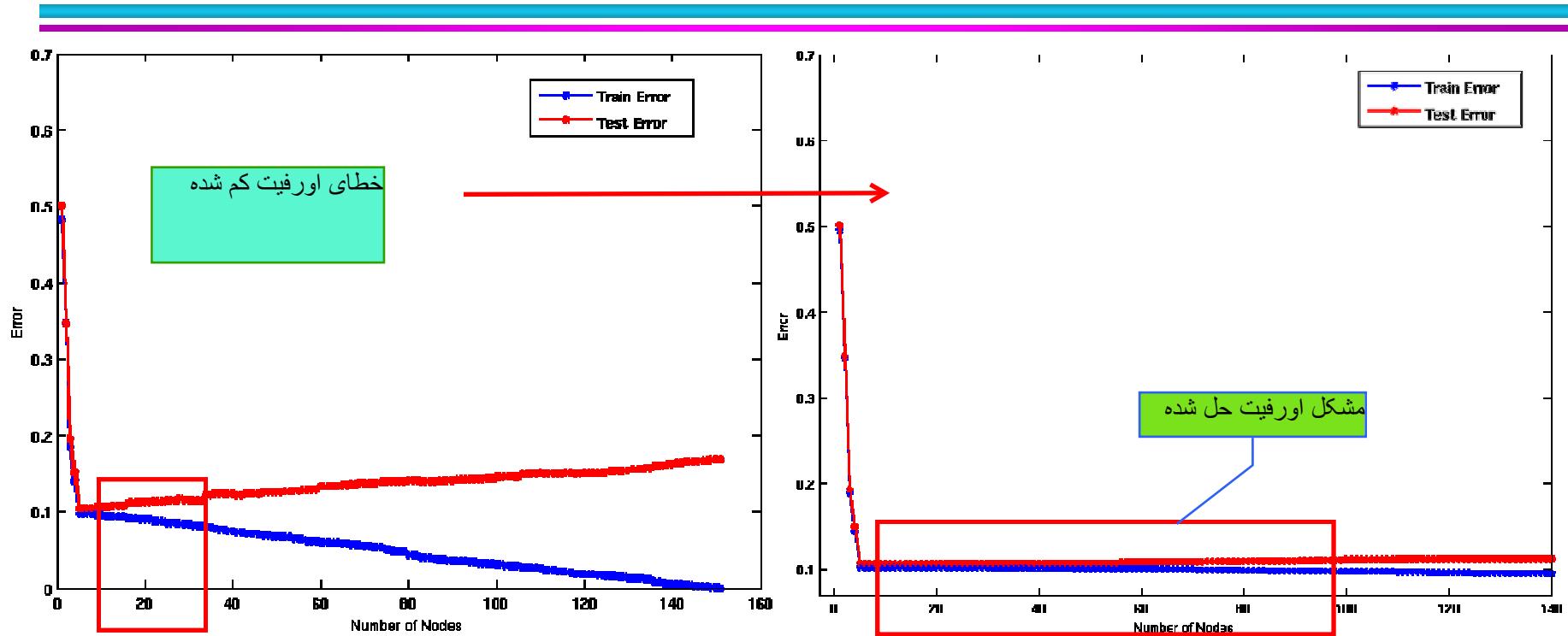
**Underfitting:** when model is too simple, both training and test errors are large

**Overfitting:** when model is too complex, training error is small but test error is large

در مدل های کلسیفیکیشن، تا یه جایی خطای داده های تست و ترین هردو شون کاهش پیدا میکنند ولی از به جایی به بعد خطای داده های تست افزایش پیدا میکنه ولی خطای داده های ترین همچنان داره کم میشه <><> اینجا جاییه که میگیم داریم دچار overfit میشیم یعنی داریم داده های ترین را حفظ میکنیم

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

# Model Overfitting – Impact of Training Data Size



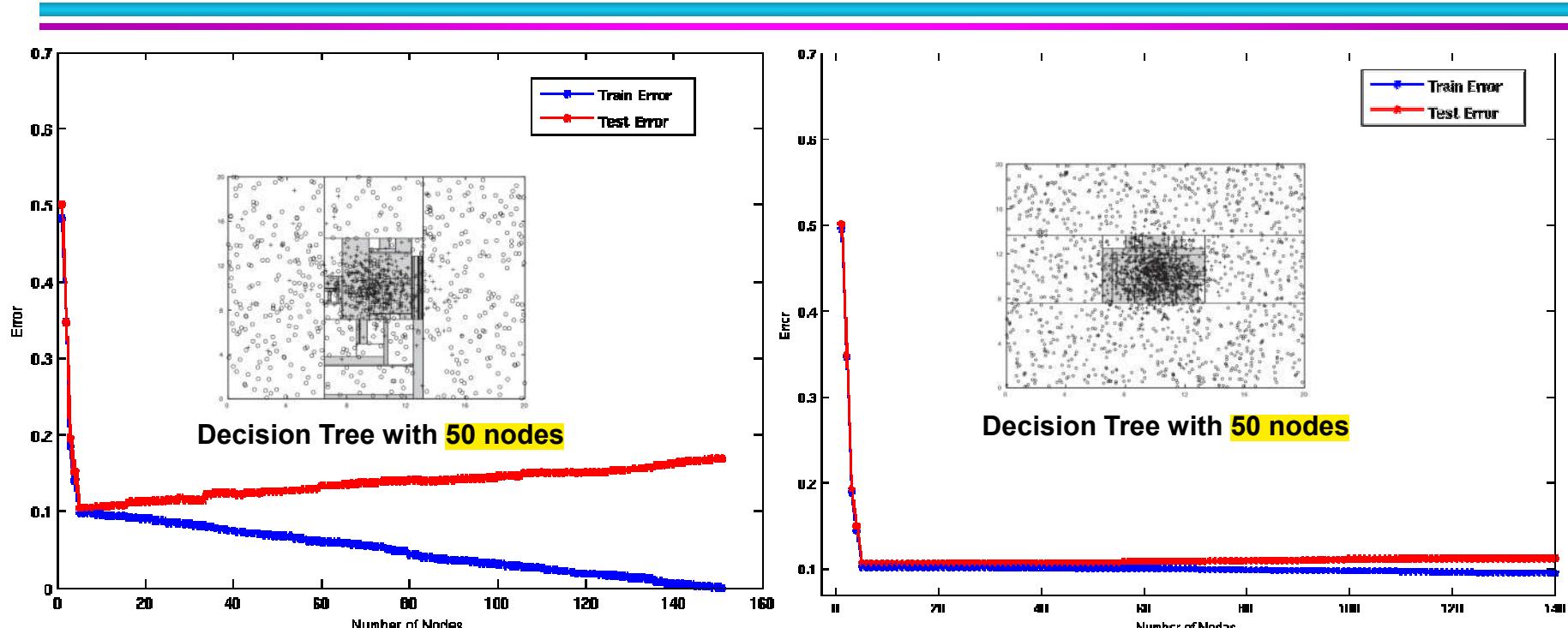
Using twice the number of data instances

- Increasing the size of training data reduces the difference between training and testing errors at a given size of model

به ۲۰ نود که میرسیم  
خطای ترین کم میشه  
همچنان ولی خطای تست  
افزایش پیدا میکنه

اگه به جای ۱۰ درصدی که برای داده های آموزشی کنار گذاشتیم، درصد بیشتری را برای آموزش اختصاص بدیم چه انفاقی میفته؟ الگوهایی که توی داده های آموزشی هستند بیشتر خودشان را نشان میدن و مدل دیرتر گیج میشه و احتمال اورفیت شدن کمتر میشه

# Model Overfitting – Impact of Training Data Size



Using twice the number of data instances

- Increasing the size of training data reduces the difference between training and testing errors at a given size of model

وقتی داده های اموزشی کم باشند نویز ها خودشون را پیشتر نشون میدن و بیشتر روی مدل تاثیر میگذارند یعنی بادیدن نمونه های مختلف سعی میکنند اون نمونه را مثل کحال خاص ببینند و برآش یه ایف جدگانه بگذارند یا یک برنج جدگانه برآش بسازند

# Reasons for Model Overfitting

- Not enough training data
- High model complexity
  - Multiple Comparison Procedure

مدل وقتی خیلی پیچیده بشه احتمال اینکه تک تصمیم گیری هارا غلط انجام بده خیلی زیاد میشه منظور از پیچیدگی چیه؟ مثلا در درخت ها منظور یا شاخص پیچیدگی میتونه تعداد نودها باشه در یک معادله مثلا تعداد پارامتر ها نشان دهنده ی پیچیدگی باشه

دلایل بیش از حد برآرash مدل  
داده های آموزشی کافی نیست  
پیچیدگی مدل بالا  
- روش مقایسه چندگانه

The impact of training data size on model overfitting can be significant. When the training set is too small relative to the complexity of the model and the variability of the data, overfitting is more likely to occur. Overfitting happens when a model is trained to fit the noise in the training data instead of the underlying patterns. This results in a high variance in the model's performance, meaning it will perform well on the training data but poorly on new data.

Increasing the size of the training set can help reduce the risk of overfitting by providing the model with more examples to learn from and better generalization of the underlying patterns. However, there is a point of diminishing returns where additional data may not provide any more benefit or may even hurt performance if the added data is noisy or irrelevant.

It's important to keep in mind that the impact of training data size on overfitting also depends on the complexity of the model being trained. A simpler model may require less data to generalize well, while a more complex model may need more data to avoid overfitting. Therefore, choosing an appropriate model complexity and having a sufficiently large and diverse training dataset are both important factors for building accurate machine learning models.

# Effect of Multiple Comparison Procedure

---

- Consider the task of predicting whether stock market will rise/fall in the next 10 trading days

- Random guessing:

$$P(\text{correct}) = 0.5$$

- Make 10 random guesses in a row:

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

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

# Effect of Multiple Comparison Procedure

---

- Approach:
  - Get 50 analysts
  - Each analyst makes 10 random guesses
  - Choose the analyst that makes the most number of correct predictions
- Probability that at least one analyst makes at least 8 correct predictions

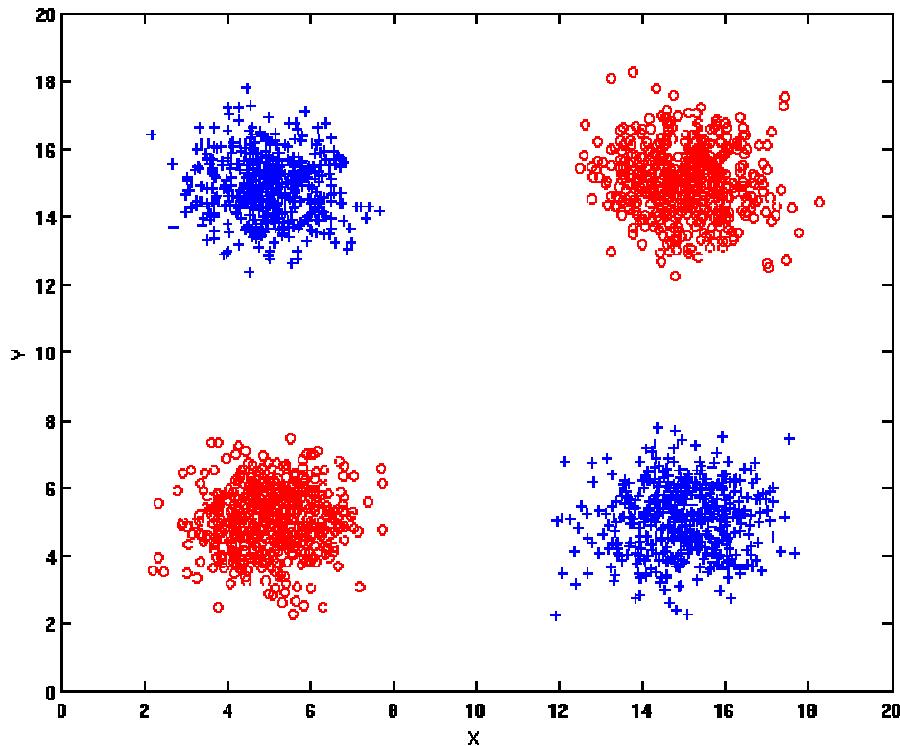
$$P(\# \text{correct} \geq 8) = 1 - (1 - 0.0547)^{50} = 0.9399$$

# Effect of Multiple Comparison Procedure

---

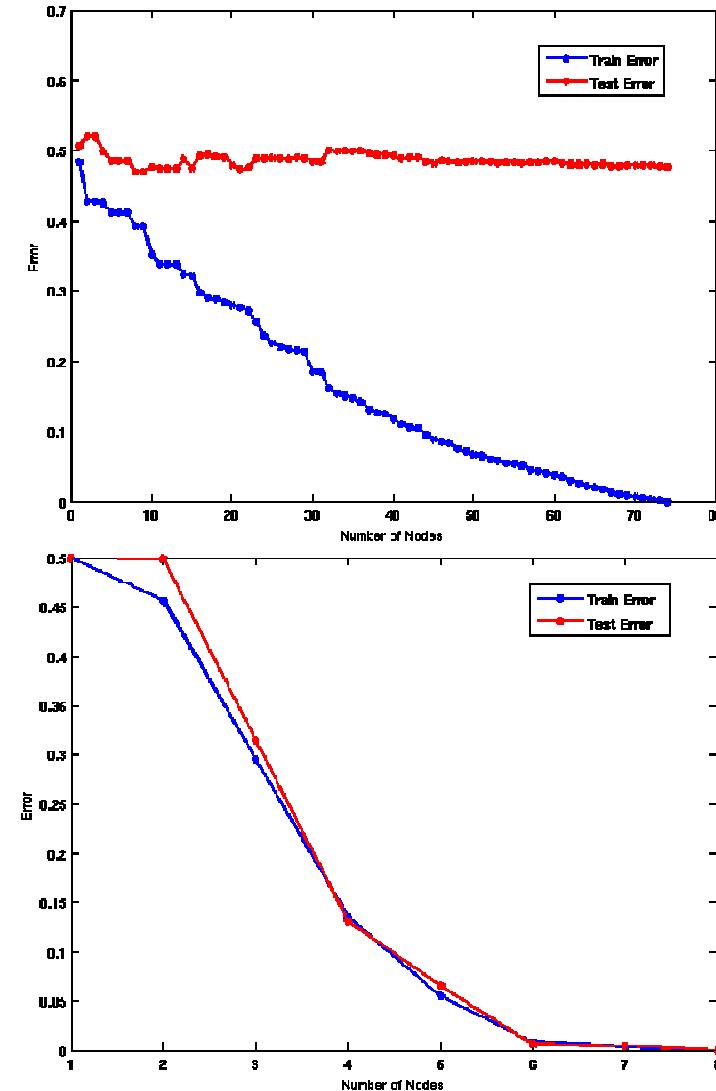
- Many algorithms employ the following greedy strategy:
  - Initial model:  $M$
  - Alternative model:  $M' = M \cup \gamma$ ,  
where  $\gamma$  is a component to be added to the model  
(e.g., a test condition of a decision tree)
  - Keep  $M'$  if improvement,  $\Delta(M, M') > \alpha$
- Often times,  $\gamma$  is chosen from a set of alternative components,  $\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_k\}$
- If many alternatives are available, one may inadvertently add irrelevant components to the model, resulting in model overfitting

# Effect of Multiple Comparison - Example



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

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



Using only X and Y as attributes

# Notes on Overfitting

---

---

- Overfitting results in decision trees that are more complex than necessary
- Training error does not provide a good estimate of how well the tree will perform on previously unseen records
- Need ways for estimating generalization errors(How?)

خطا روی داده های تستمون داره خوب عمل میکنه ولی روی داده هایی که مدل تا حالا ندیده چطوری عمل میکنه؟

نکاتی در مورد Overfitting

برازش بیش از حد منجر به درخت های تصمیم می شود که پیچیده تر از حد لازم هستند  
خطای آموزشی تخمین خوبی از عملکرد درخت در رکوردهای نادیده قبلی ارائه نمی دهد.  
به روش هایی برای تخمین خطاهای تعمیم نیاز دارید (چگونه?)

# Model Selection

میخایم بین مدل ها انتخاب کنیم که کومنشون  
برای داده های ما بهتره؟ مثلاً مدلی که ۴ تا نود  
داره بهتره یا اونی که ۵۰ تا نود داره بهتره؟

- ۲ تارویکرد یا راه حل برای  
انتخاب مدل هست:
۱. رویکرد داده محور که به  
مجموعه ای validation ارزیابی در نظر میگیریم
  ۲. رویکرد محاسباتی و فرمولی

## Purpose

- ensure that model is **not overly complex** (to avoid overfitting)
- Performed **during model building**

در حین بادگیری باید  
سریع راجع بش تصمیم  
بگیریم که کدام مدل  
بهتره؟

- Need to **estimate generalization error**
  - Using **Validation Set**
  - Incorporating **Model Complexity**

هدف  
اطمینان حاصل کنید که مدل بیش از حد پیچیده نیست (برای جلوگیری از  
برازش بیش از حد)  
در حین ساخت مدل اجرا می شود  
نیاز به تخمین خطای تعمیم  
استفاده از **Validation Set**  
گنجاندن پیچیدگی مدل

Estimating generalization error is important to ensure that a machine learning model can perform well on new, unseen data. One common approach to estimating generalization error is to split the available data into training and validation sets. The training set is used to train the model, while the validation set is used to evaluate its performance and estimate its generalization error.

To do this, we typically train the model on the training set and then use the validation set to measure its performance. We can repeat this process multiple times, using different subsets of the data for training and validation, to get an estimate of the model's average performance and generalization error. This process is known as cross-validation.

Another approach to estimating generalization error is to use a holdout set, which is a small portion of the available data that is not used during training or validation. After we have trained and validated our model, we can use the holdout set to test its performance on completely new, unseen data. This gives us an estimate of the model's true generalization error.

## Model Selection:

# Using Validation Set

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

## Model Selection:

# Incorporating Model Complexity

انتخاب بر اساس پیچیدگی مدل  
مدلی خوبه که کمتر پیچیده باشه مدلی که از همه ساده  
تره بهتره

با توجه به دو مدل از خطاهای تعمیم مشابه، باید مدل ساده‌تر را بر مدل پیچیده  
تر ترجیح داد  
- یک مدل پیچیده شناس بیشتری برای فیت شدن تصادفی دارد-

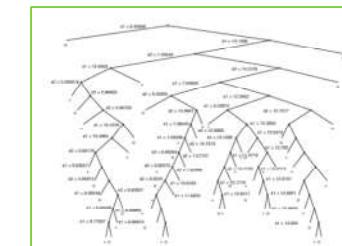
- Rationale: Occam's Razor

- Given two models of similar generalization errors,  
one should prefer the simpler model over the more complex model

- A complex model has a greater chance of being fitted accidentally
- Therefore, one should include model complexity when evaluating a model

بنابراین، هنگام ارزیابی یک مدل باید پیچیدگی مدل  
را لاحظ کرد

اندازه گیری پیچیدگی مدل به کلیفار برستگی دارد  
مثلاً توی درخت پیچیدگی را با عمقش مثلاً میسنجیم یا تعداد  
برگ هاش



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

خطای اموزشی

$$\alpha \times \text{Complexity(Model)}$$

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

جریمه در صورت افزایش پیچیدگی مدل

# Estimating the Complexity of Decision Trees

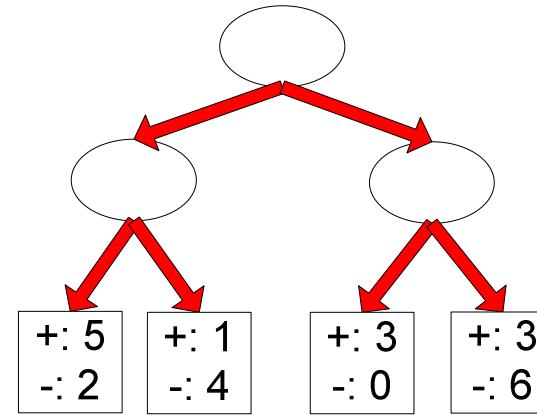
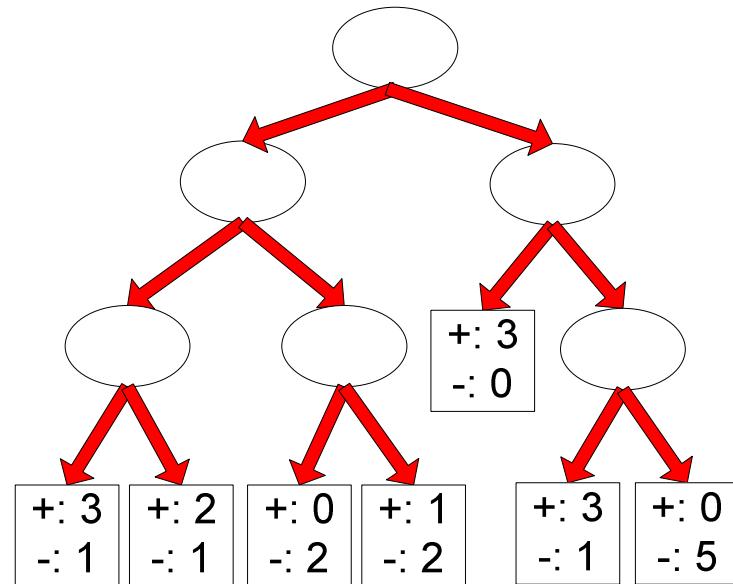
---

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

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

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

# Estimating the Complexity of Decision Trees: Example



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

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

$$\Omega = 1$$

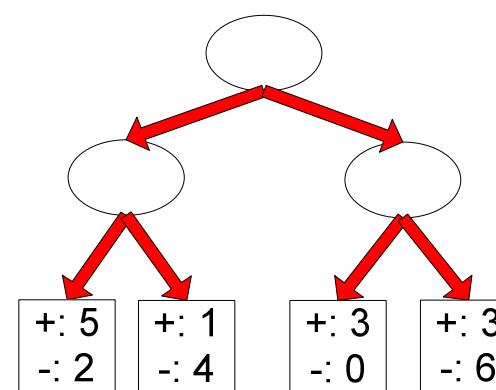
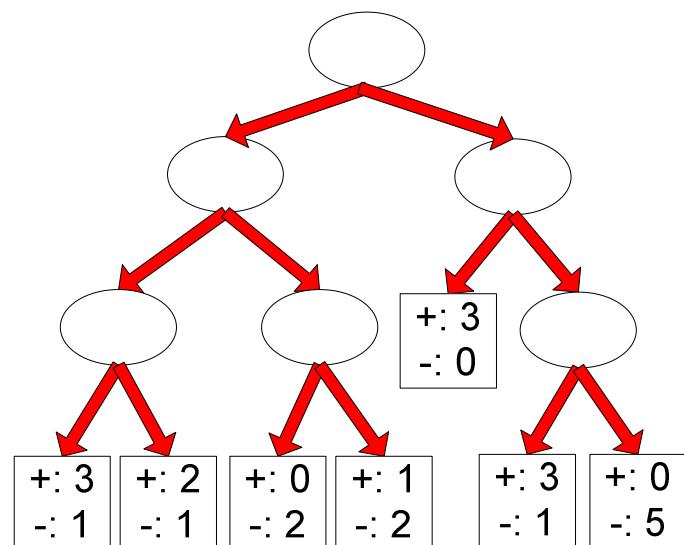
$$e_{\text{gen}}(T_L) = 4/24 + 1 * 7/24 = 11/24 = 0.458$$

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

# Estimating the Complexity of Decision Trees

- Resubstitution Estimate:

- Using training error as an **optimistic** estimate of generalization error
- Referred to as **optimistic error** estimate

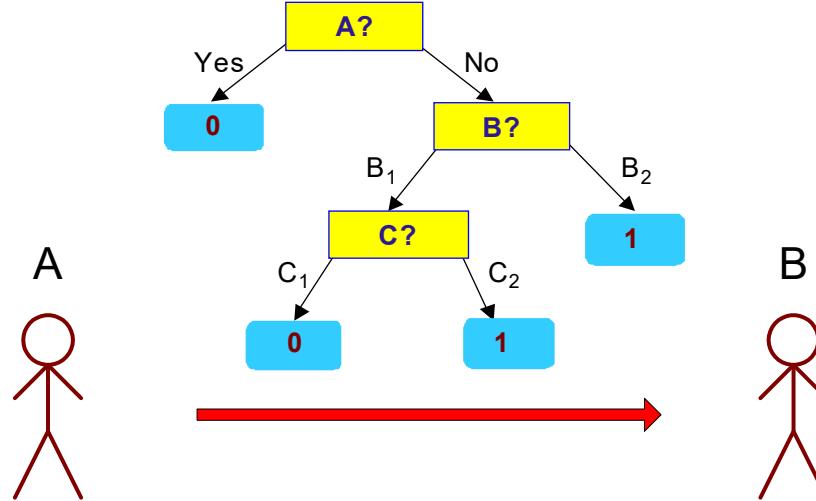


$$e(T_L) = 4/24$$

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

# Minimum Description Length (MDL)

X	y
$X_1$	1
$X_2$	0
$X_3$	0
$X_4$	1
...	...
$X_n$	1



X	y
$X_1$	?
$X_2$	?
$X_3$	?
$X_4$	?
...	...
$X_n$	?

- $\text{Cost}(\text{Model}, \text{Data}) = \text{Cost}(\text{Data}|\text{Model}) + \alpha \times \text{Cost}(\text{Model})$ 
  - Cost is the number of bits needed for encoding.
  - Search for the least costly model.
- $\text{Cost}(\text{Data}|\text{Model})$  encodes the misclassification errors.
- $\text{Cost}(\text{Model})$  uses node encoding (number of children) plus splitting condition encoding.

# Model Selection for Decision Trees

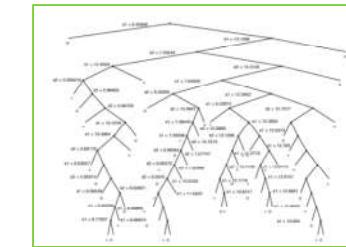
## ● Pre-Pruning (Early Stopping Rule)

- Stop the algorithm before it becomes a fully-grown tree

مثلما از بزرگ شدن درخت من راضی نیستیم باید هر شش کنیم مثلا بعضی جاها خوب نیست همون اول کار درخت های کوچیک برای مدلمن بسازیم و بهتره یه درخت بزرگ بسازیم بعد هر شش کنیم

- Typical stopping conditions for a node:

- ◆ Stop if all instances belong to the same class
- ◆ Stop if all the attribute values are the same



- More restrictive conditions:

- ◆ Stop if number of instances is less than some user-specified threshold
- ◆ Stop if class distribution of instances are independent of the available features (e.g., using  $\chi^2$  test)
- ◆ Stop if expanding the current node does not improve impurity measures (e.g., Gini or information gain).
- ◆ Stop if estimated generalization error falls below certain threshold

- شرایط محدودتر: اگر تعداد نمونه ها کمتر از آستانه تعیین شده توسط کاربر باشد، متوقف شود.  
اگر توزیع کلاس نمونه ها مستقل از ویژگیهای موجود باشد، متوقف شود.  
اگر گسترش گره فعلی معیار های ناخالصی را بهبود نمی بخشد (به عنوان مثال، جینی یا افزایش اطلاعات) متوقف شود.  
اگر خطای تعمیم تخمینی زیر آستانه معین قرار گرفت، متوقف کنید.

قبل از هرس (قانون توقف زودهنگام)  
- الگوریتم را قبل از اینکه به درختی کاملاً رشد کرده تبدیل شود متوقف کنید.  
- شرایط توقف معمول برای یک گره: اگر همه نمونه ها به یک کلاس تعلق دارند، توقف کنید.  
اگر همه اطلاعات ها مقادیر یکسان دارند، توقف کنید.

# Model Selection for Decision Trees

اجازه بده درخت خیلی بزرگ شه بعد پال ها و برگهای  
اضافه رو هرس کن و حذفشون کن

## ● Post-pruning

- Grow decision tree to its entirety
- Subtree replacement
  - ◆ Trim the nodes of the decision tree in a bottom-up fashion
  - ◆ If generalization error improves after trimming, replace sub-tree by a leaf node
  - ◆ Class label of leaf node is determined from majority class of instances in the sub-tree

پس از هرس

- درخت تصمیم را به طور کامل رشد دهید  
- جایگزینی درخت فرعی

گره های درخت تصمیم را به صورت پایین به بالا برش دهید  
اگر خطای تعمیم پس از اصلاح بهبود یافت، درخت فرعی را با یک گره برگ جایگزین کنید  
برچسب کلاس گره برگ از کلاس اکثر نمونه ها در زیر درخت تعیین می شود.

# Example of Post-Pruning

misclassification error =  $1 - \max(\pi(t))$   
 error =  $1 - \max(20/30, 10/30) = 1 - 20/30 = 10/30$

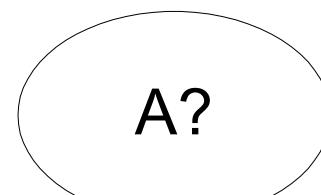
Training Error (Before splitting) =  $10/30$

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

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

اگه عمدہ ی لیبل هارا  
درنظر بگیریم که ۲۰ تا  
یس داریم پس کل را یس  
میگیریم پس خطای میشه  
 $10/30$   
چون ۱۰ تا از داده های  
اموزشمن را غلط گفتیم

لان برای این اوری که  
داریم آیا باید برنج جدید  
بسازیم برای تصمیم  
گیری درمورد اینها؟ یا نه  
مثلاً نود A را میتوانیم  
اضافه کنیم میتوانیم نکنیم



Class = Yes	8	Class = Yes	3	Class = Yes	4	Class = Yes	5
Class = No	4	Class = No	4	Class = No	1	Class = No	1

برچسب این شاخه چی  
میشه؟  
yes  
چون اکثریت یس هستند

برچسب این شاخه:

label = yes

label = yes

# Example of Post-Pruning

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

Training Error (Before splitting) = 10/30

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

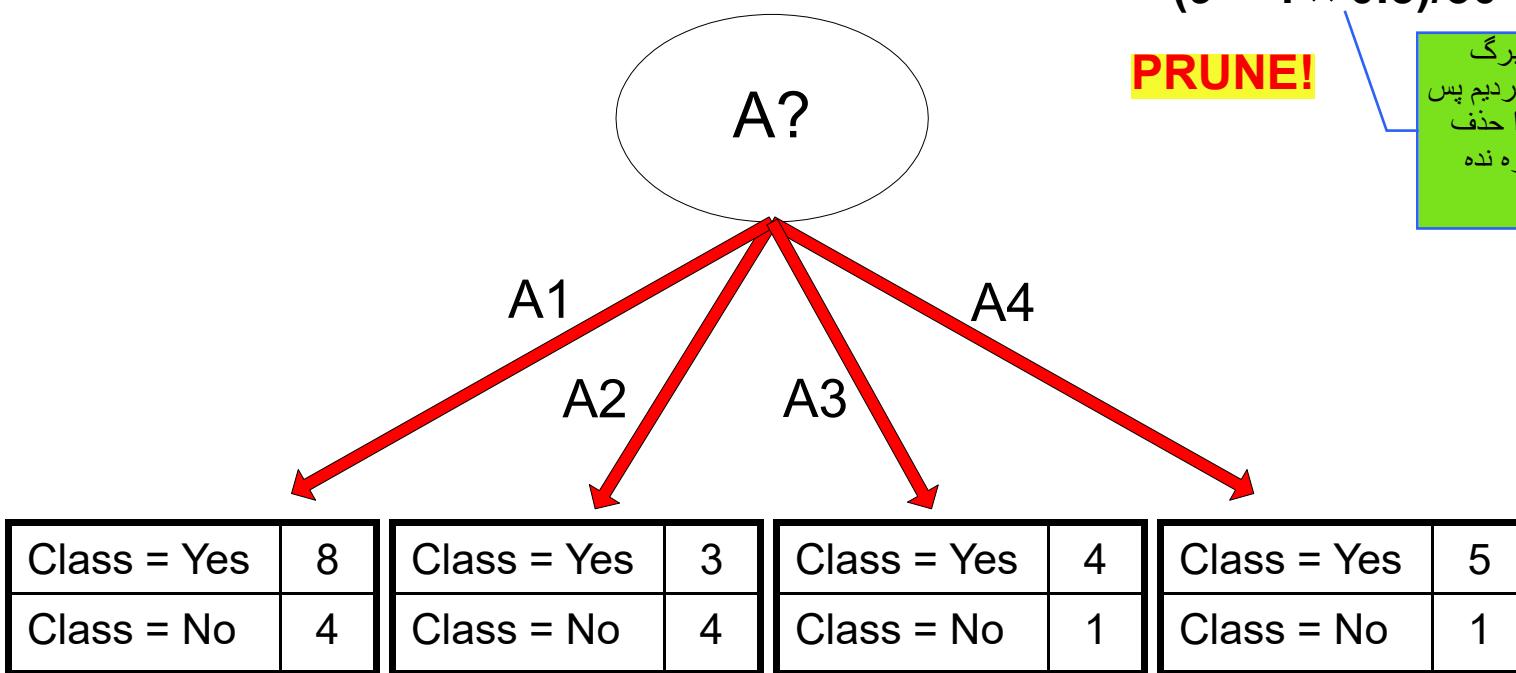
Training Error (After splitting) = 9/30

Pessimistic error (After splitting)

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

**PRUNE!**

جریمه‌ی تعداد برگ  
مایی که اضافه کردیم پس  
میگیم این نود را حذف  
کن یا اصلاً اجازه نده  
اضافه شه



# Examples of Post-pruning

## Decision Tree:

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

Subtree Raising

## Simplified Decision Tree:

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

Subtree Replacement

Suppose we have a dataset of 100 patients and their corresponding symptoms and diagnoses. We split this dataset into a training set of 80 patients and a validation set of 20 patients.

We use the training set to build a decision tree with 5 levels, which achieves 95% accuracy on the training data. However, we suspect that this tree may be overfitting, so we decide to perform post-pruning.

To do this, we start at the bottom of the tree and evaluate the cost complexity of pruning each leaf node using an alpha parameter. For simplicity, let's say that each leaf node has a cost complexity of 1.

We calculate the total cost complexity of the original tree by summing the individual cost complexities of all the leaf nodes, which is equal to 10 (since there are 10 leaf nodes in the tree).

Next, we try out different values of alpha and evaluate the validation accuracy of the pruned tree for each alpha value. Let's say we get the following results:

Alpha = 0: Pruned tree still has 5 levels, validation accuracy = 92%

Alpha = 1: Pruned tree has 4 levels, validation accuracy = 93%

Alpha = 2: Pruned tree has 3 levels, validation accuracy = 94%

Alpha = 3: Pruned tree has 2 levels, validation accuracy = 92%

Alpha = 4: Pruned tree has 1 level, validation accuracy = 90%

Based on these results, we choose an alpha value of 2, which gives us the simplest tree with the highest validation accuracy (94%). This new pruned tree has 3 levels and is our final decision tree.

Note that in practice, we might use cross-validation to evaluate the accuracy of the pruned tree more robustly and avoid overfitting on the validation set.

Let's say we have a decision tree with 1000 nodes, and it was trained using a dataset containing 10,000 samples. We want to use post-pruning to reduce the size of the decision tree while maintaining its accuracy.

To do this, we would first split our dataset into two parts: a training set and a validation set. Let's use an 80-20 split, which means that 8,000 samples would be used for training, and 2,000 samples would be used for validation.

We would then train our decision tree using the training set and perform pruning on the resulting tree using the validation set. One common post-pruning technique is reduced error pruning, which works as follows:

Starting at the leaves of the decision tree, replace each subtree with its majority class.

Evaluate the accuracy of the pruned tree on the validation set.

If the accuracy has improved, keep the pruned tree. Otherwise, restore the original subtree.

We repeat steps 1-3 until we can no longer improve the accuracy on the validation set.

Let's say that after pruning, our decision tree has 800 nodes instead of 1000. We can then test the accuracy of the pruned tree on a test set, which is a separate dataset from the training and validation sets. This will give us an estimate of how well our decision tree will perform on new, unseen data.

## differences between training set evaluation and testing set evaluation

Let's say we're building a machine learning model to classify images of cats and dogs. We have a dataset containing 10,000 images, with 5,000 images of cats and 5,000 images of dogs. We split the dataset into two parts: a training set containing 8,000 images (4,000 cats and 4,000 dogs) and a testing set containing 2,000 images (1,000 cats and 1,000 dogs).

During training, we use the training set to optimize the parameters of the model. For example, we might use an algorithm like logistic regression or a neural network, and we adjust the weights of the model until it produces accurate predictions on the training set.

After training, we want to evaluate the performance of the model on new, unseen data. This is where the testing set comes in. We use the testing set to measure how well our model generalizes to new data that it has not seen before. We can calculate metrics like accuracy, precision, recall, and F1 score based on the predictions the model makes on the testing set.

The key difference between training set evaluation and testing set evaluation is that training set evaluation measures how well the model fits the training data, while testing set evaluation measures how well the model generalizes to new, unseen data. It's important to perform both types of evaluation to ensure that the model is not overfitting to the training data and that it will perform well on real-world examples.

# Model Evaluation

ارزیابی مدل

هدف:  
- تخمین عملکرد کلسیفایر بر روی داده های دیده نشده  
قبلی (مجموعه تست)  
Holdout  
اعتبار سنجی متقابل

Purpose:

- To estimate performance of classifier on previously unseen data (test set)
- Holdout
- Cross validation

Holdout is a method for evaluating machine learning models that involves splitting the available data into two sets: a training set and a validation set (also called a holdout set or testing set).

In a holdout evaluation, we randomly split the available data into two parts, typically using a 70-30 or 80-20 split. The larger portion of the data is used to train the model, while the smaller portion is reserved for validation.

Once the model has been trained on the training set, it is evaluated on the validation set. The performance of the model on the validation set gives us an estimate of how well the model will perform on new, unseen data.

The advantage of using a holdout evaluation is that it provides an unbiased estimate of the model's performance, since the validation set was not used during training. This can help us avoid overfitting, which occurs when the model performs well on the training set but poorly on new data.

However, one potential disadvantage of holdout evaluation is that it may not be as statistically robust as other methods, such as cross-validation, especially if the dataset is small. In addition, the specific split of the data into training and validation sets can have a large impact on the estimated performance of the model. To address this issue, it's common to perform multiple rounds of holdout evaluation with different random splits of the data and average the results.

Overall, holdout evaluation is a useful and widely-used method for evaluating machine learning models, but it should be used in conjunction with other methods and best practices to obtain a more comprehensive performance assessment of the model.

Sure, here's an example of how to use holdout for evaluating a machine learning model in Python:

python Copy Code

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Load data
X, y = load_data()

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a logistic regression model on the training set
model = LogisticRegression()
model.fit(X_train, y_train)

# Evaluate the model on the testing set
score = model.score(X_test, y_test)
print(f"Accuracy: {score}")
```

In this example, we first load our data (not shown). We then split the data into a training set and a testing set using `train_test_split` from the `sklearn.model_selection` module. Here, we're using 20% of the data as our testing set.

Next, we train a logistic regression model on the training set with `LogisticRegression` from the `sklearn.linear_model` module. Finally, we evaluate the model on the testing set by computing the accuracy with `score`.

In the example I provided, the testing set and the validation set are actually the same thing.

The `train_test_split()` function from scikit-learn is commonly used to split a dataset into two parts: a training set and a testing set. The testing set is then used to evaluate the performance of the model on unseen data after it has been trained on the training set.

However, in some cases, we may want to tune the hyperparameters of our model or perform feature selection using a third subset of the data, which is called a validation set. In this case, we would typically split the original dataset into three parts: a training set, a validation set, and a testing set.

To split the dataset into three parts instead of two, you can modify the code as follows:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Load data
X, y = load_data()

# Split data into training, validation, and testing sets
X_train_val, X_test, y_train_val, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
X_train, X_val, y_train, y_val = train_test_split(X_train_val, y_train_val, test_size=0.25, random_state=42)

# Train a Logistic regression model on the training set
model = LogisticRegression()
model.fit(X_train, y_train)

# Evaluate the model on the validation set
val_score = model.score(X_val, y_val)
print(f"Validation Accuracy: {val_score}")

# Evaluate the model on the testing set
test_score = model.score(X_test, y_test)
print(f"Testing Accuracy: {test_score}")
```

In this modified example, we first split the data into a training/validation set (80% of the data) and a testing set (20% of the data). We then split the training/validation set further into a training set (60% of the original data) and a validation set (20% of the original data).

We train the logistic regression model on the training set and evaluate its performance on the validation set by computing the accuracy score. We use this score to tune the hyperparameters of our model or perform feature selection.

Finally, we evaluate the performance of the model on the testing set, which gives us an estimate of how well the model will generalize to new, unseen data.

Cross-validation is a technique used to evaluate the performance of a machine learning model by partitioning the data into multiple subsets (called folds), training the model on some of the folds, and testing it on the remaining fold. The process is repeated multiple times, with a different fold reserved for testing each time. This allows us to obtain more reliable estimates of the model's performance than a single train/test split would provide.

There are several types of cross-validation, but the most common one is k-fold cross-validation. Here's how it works:

The data is partitioned into k equal-sized folds.

For each fold i, the model is trained on the remaining k-1 folds.

The model is tested on the held-out fold i and the performance metric is recorded.

Steps 2-3 are repeated k times, with each fold being used exactly once as the test set.

The CV score is computed by averaging the performance metric over all k folds.

Here's an example of how to perform k-fold cross-validation in Python using scikit-learn:

```
from sklearn.model_selection import KFold
from sklearn.linear_model import LogisticRegression

# Load data
X, y = load_data()

# Define the number of folds
num_folds = 5

# Create a k-fold cross-validation object
kf = KFold(n_splits=num_folds)

# Initialize a list to store the validation scores
val_scores = []
```

```

# Train and validate the model on each fold
for train_idxs, val_idxs in kf.split(X):

    # Split the data into training and validation sets
    X_train, X_val = X[train_idxs], X[val_idxs]
    y_train, y_val = y[train_idxs], y[val_idxs]

    # Train a Logistic regression model on the training set
    model = LogisticRegression()
    model.fit(X_train, y_train)

    # Evaluate the model on the validation set
    val_score = model.score(X_val, y_val)
    val_scores.append(val_score)

# Compute the mean validation score over all folds
mean_val_score = np.mean(val_scores)

print(f"Mean cross-validation score: {mean_val_score}")

```

In this example, we first load our data (not shown). We define the number of folds (in this case, 5) and create a KFold object using scikit-learn's KFold function. We then loop over each fold and split the data into training and validation sets, as before. However, instead of computing a single test score, we compute a validation score for each fold and store it in a list. Finally, we compute the mean validation score over all folds.

Stratified cross-validation is a type of cross-validation technique used in machine learning to evaluate the performance of a model. In this technique, the data is divided into several subsets or folds, ensuring that each fold has a representative proportion of the different classes found in the dataset.

For example, let's say we have a dataset with 1000 instances, where 600 are labeled as class A and 400 are labeled as class B. To perform stratified cross-validation, we would split the dataset into several folds, such as 5 or 10, while maintaining the same proportion of classes in each fold. So, if we were using 5 folds, we would ensure that each fold had 120 instances of class A ( $600/5$ ) and 80 instances of class B ( $400/5$ ).

This approach ensures that the model is trained and evaluated on data that is representative of the overall distribution of classes in the dataset, which can lead to more accurate and reliable performance metrics.

here is an example of performing stratified cross-validation in Python using the StratifiedKFold function from the sklearn.model\_selection module:

```
from sklearn.model_selection import StratifiedKFold
import numpy as np

X = np.array([[1, 2], [3, 4], [5, 6], [7, 8], [9, 10]])
y = np.array([0, 0, 1, 1, 1])

skf = StratifiedKFold(n_splits=3)
for train_index, test_index in skf.split(X, y):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]
```

In this example, we're creating a dataset with 5 instances and two features, where the first two instances belong to class 0 and the remaining three instances belong to class 1. We're then using StratifiedKFold with n\_splits=3 to perform stratified 3-fold cross-validation on the dataset.

The split method of StratifiedKFold returns the indices of the training and testing data for each fold. We're then splitting the data into training and testing sets based on these indices and printing them out for each fold.

Repeated cross-validation is a technique used in machine learning to obtain a more robust estimate of model performance by repeating the cross-validation process multiple times using different random partitions of the data.

In repeated cross-validation, the dataset is randomly split into training and testing sets, and the model is trained and evaluated on each split. This process is then repeated for a specified number of times, with different random splits each time. The results from each repetition are then averaged to obtain an overall performance estimate.

Here's an example of how to perform repeated cross-validation in Python using scikit-learn:

```
from sklearn.model_selection import cross_val_score, RepeatedKFold
from sklearn.linear_model import LogisticRegression
import numpy as np

X = np.array([[1, 2], [3, 4], [5, 6], [7, 8], [9, 10]])
y = np.array([0, 0, 1, 1, 1])

model = LogisticRegression()
cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)
scores = cross_val_score(model, X, y, scoring='accuracy', cv=cv, n_jobs=-1)

print('Mean Accuracy: %.3f' % np.mean(scores))
```

In this example, we're using `RepeatedKFold` with `n_splits=10` and `n_repeats=3` to perform 30-fold cross-validation (10 splits, 3 repeats) on the dataset. We're also specifying `random_state=1` to ensure reproducibility of the results.

We're using logistic regression as our model and evaluating its accuracy using the accuracy metric. Finally, we're printing the mean accuracy score across all the repetitions.

`cross_val_score` is a function in scikit-learn that performs cross-validation to evaluate the performance of a machine learning model. It takes a model, input features, target variable, and a number of cross-validation parameters as input, and returns the performance score for each fold of the cross-validation.

The `cross_val_score` function works by splitting the data into k-folds (or more generally n-folds), where k is specified using the `cv` parameter. For each fold, it trains the model on  $k-1$  folds of the data and evaluates its performance on the remaining fold, returning the performance score for that fold. This process is repeated k times, with each fold being used as the test set once, resulting in k performance scores.

These individual performance scores can then be aggregated in various ways, such as taking their average or computing their standard deviation, to obtain an overall estimate of the model's performance.

Here is an example of how to use `cross_val_score` in Python to evaluate the performance of a logistic regression model using 10-fold cross-validation:

```
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
import numpy as np

X = np.array([[1, 2], [3, 4], [5, 6], [7, 8], [9, 10]])
y = np.array([0, 0, 1, 1, 1])

model = LogisticRegression()
scores = cross_val_score(model, X, y, cv=10)

print('Cross-Validation Scores:', scores)
print('Mean Performance Score:', np.mean(scores))
```

In this example, we're using `cross_val_score` to perform 10-fold cross-validation on a logistic regression model trained on the dataset `X` with corresponding labels `y`. The function returns an array of 10 performance scores, one for each fold of the cross-validation. We're printing these scores and also computing their mean to obtain an overall estimate of the model's performance.

# Model Evaluation: Holdout

---

---

- Holdout
  - Reserve  $k\%$  for training and  $(100-k)\%$  for testing
  - Random subsampling: repeated holdout

اینکه این  $k$  چی باشه رو  
میشه به صورت رندم  
انتخاب کرد چندین بار

# Model Evaluation: Cross-validation

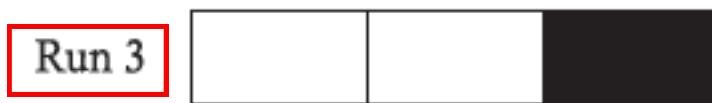
- Cross validation

- Partition data into  $k$  disjoint subsets
- $k$ -fold: train on  $k-1$  partitions, test on the remaining one
- Leave-one-out:  $k=n$

تعداد رکوردها یا تعداد  
اجکت هامون

## 3-fold cross-validation

$S_1$        $S_2$        $S_3$



مدام تکرار کنیم که یه مشکلی پیش میاد  
مثلا اگه داده هامون دو کلاسه باشن،  
برچسب هاشون متوازن نباشه ینی هامون  
قدر که داده ی کلاس یک داریم داده ی  
کلاس منهای یک نداشته باشیم  
وقتی داریم داده هامون رو پارتیشن بندی  
میکنیم ممکنه عده ش توی یک کلاس  
باشه و تعداد کمیش توی یه کلاس بره  
حتی ممکنه توی یک پارتیشن همه توی  
یک کلاس برن و توی اون پارتیشن اصلا  
کلاس دوم را نداشته باشیم

## تغییرات در اعتبار سنجی متقابل

# Variations on Cross-validation

اعتبار سنجی متقابل مکرر

- چند بار اعتبار سنجی متقابل را انجام دهید
- تخمینی از واریانس خطای تعیین می دهد

## ● Repeated cross-validation

- Perform cross-validation a number of times
- Gives an estimate of the variance of the generalization error

## ● Stratified cross-validation

طبقه بندی شده

Guarantee the same percentage of class labels in training and test

- Important when classes are imbalanced and the sample is small

## ● Use nested cross-validation approach for model selection and evaluation

از روش اعتبار سنجی متقابل تو در تو برای انتخاب و ارزیابی مدل استفاده کنید

ta Minim

اعتبار سنجی متقاطع طبقه بندی شده

- ضمانت درصد یکسان از برچسب کلاس در آموزش و آزمون
- زمانی که کلاس ها نامتعادل هستند و نمونه کوچک است مهم است

Instance-based learning is a type of machine learning algorithm that makes predictions based on a set of known data points, called instances. Instead of building a model to represent the relationships between input features and output targets, instance-based learning stores all the training examples and uses them to make predictions on new data.

To make a prediction for a new input instance, the algorithm searches the training set for instances that are similar to the new input and then uses the outputs associated with those similar instances to compute a prediction for the new instance. The similarity between instances is usually measured using a distance metric such as Euclidean distance or cosine distance.

Instance-based learning is particularly useful when the relationship between inputs and outputs is complex or unknown, or when the training data is noisy or incomplete. Because it does not require the construction of an explicit model, instance-based learning can be more computationally efficient and easier to implement than other machine learning algorithms.

# INSTANCE-BASED LEARNING

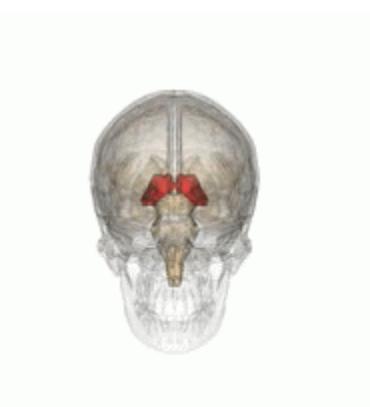
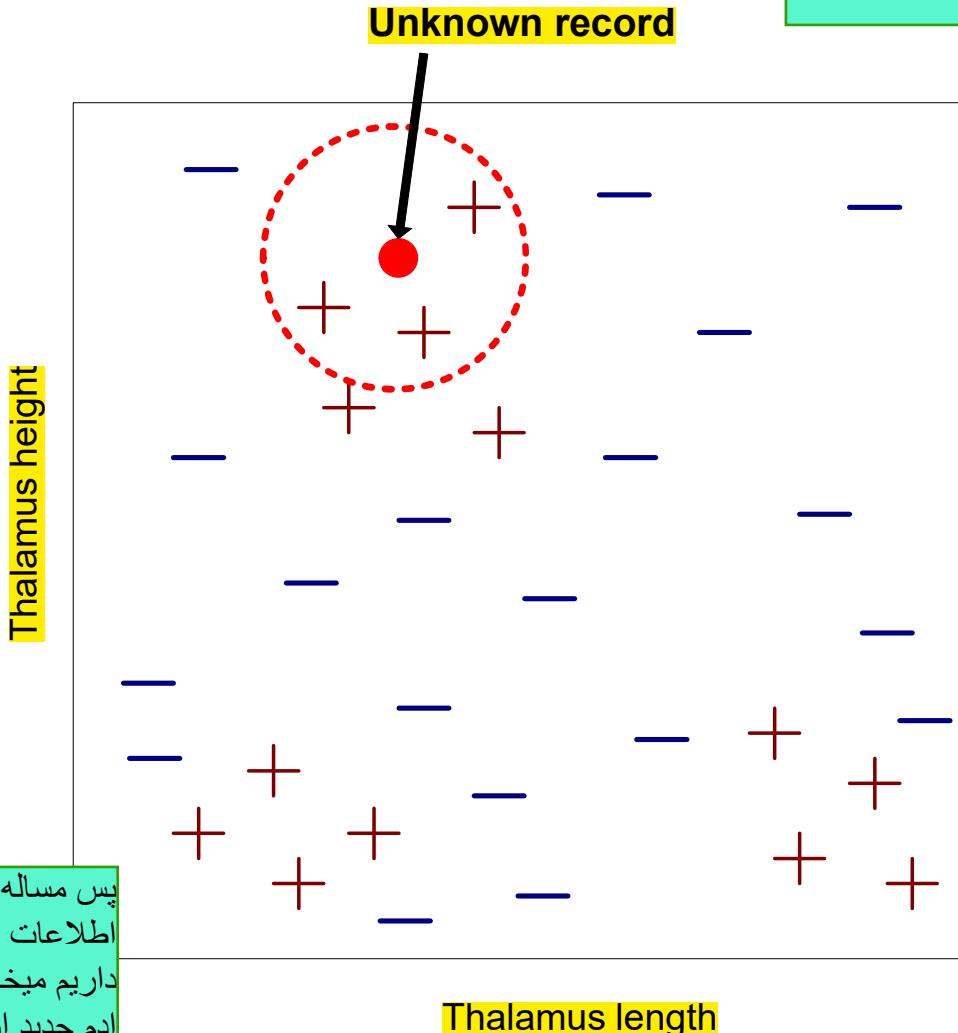
روش های مبتنی بر نمونه ها

از داده های اموزشی مستقیم برای پیش بینی نمونه تستی که ما بر چسبش را نمیدانیم سعی میکنیم کمک بگیره

ر درخت تصمیم ما سعی میکردیم داده ها را به فضایی مثل فضای مدبیریم و یک درختی بسازیم ولی اینجا به کمک خود داده ها تصمیم میگیریم

# Basic Idea\_

پزشک ها میگن مثلا اگه اندازه ی طول تalamوس زیاد شد فلان بیماری را داره یا اگه ارتفاعش زیاد شد فلان بیماری پزشک ها از ما یه سوالی پرسیدن: ما در بیماری های مختلف، طول و ارتفاع تalamوس شون را اندازه گرفتیم و میدانیم که مثلا افرادی که طول تalamوسشون کم و ارتفاعش زیاده آدم هایی هستند که بیماری دارند



[https://en.wikipedia.org/wiki/File:Thalamus\\_small.gif](https://en.wikipedia.org/wiki/File:Thalamus_small.gif)

پس مساله ی ما اینطوریه که یه سری اطلاعات راجع به یه سری بیمارو ادم سالم داریم میخاهیم تصمیم گیری کنیم که وقتی یه ادم جدید اوmd که ما طول و ارتفاع تalamوسش را داشتیم باید بش بگیم مریض است یا سالم؟ پس این فضا میشه داده های اموزش ما با روش های ماشین لرنینگ باید برچسب این ادم جدید را مشخص کنیم

# Nearest Neighbor Classifiers

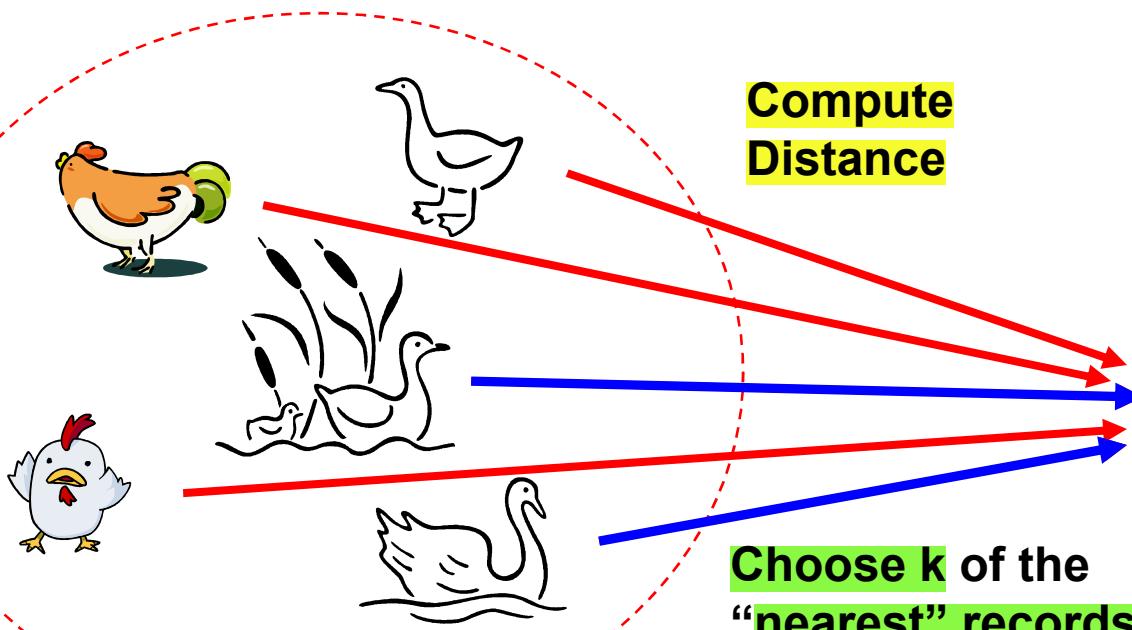
این روش میگه نگاه کن اون نمونه ای که برای تست اورده نزدیک هاش چه نمونه های دیگری هستند و از نمونه های نزدیکش کمک بگیر

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck

ایده پایه:  
- اگر مثل اردک راه میرود، مثل اردک کوک میکند، احتمالاً اردک است

پس ما میخاهم به کمک همسایه های نمونه تست، تصمیم گیری کنیم که برچسب داده ی جدید چیه؟  
به این روش تصمیم گیری میگیم  
nearest neighbor:  
classifier  
بنی نزدیک ترین همسایه ها

Training  
Records

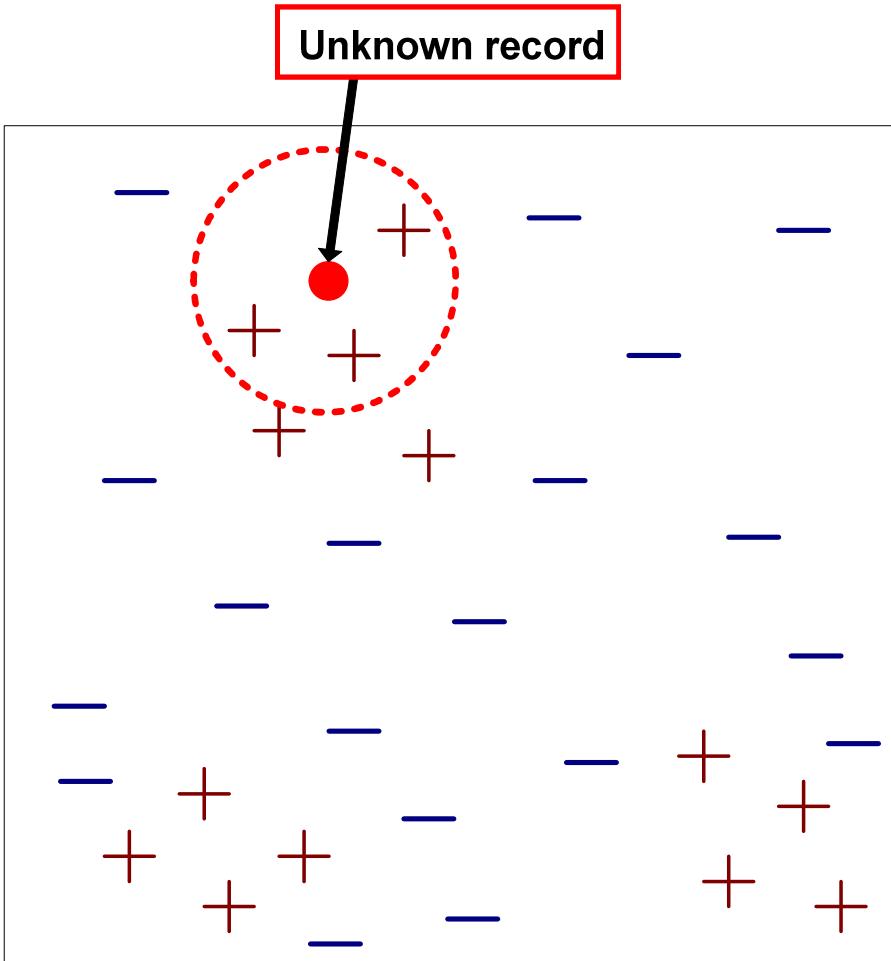


Test  
Record



میخاهم تصمیم بگیریم که  
این حیوان اردک است یا  
خرس است یا ....  
میبینیم شبیه چه حیوانی  
رفتار میکنه میگیم  
احتمالاً همونه

# Nearest-Neighbor Classifiers



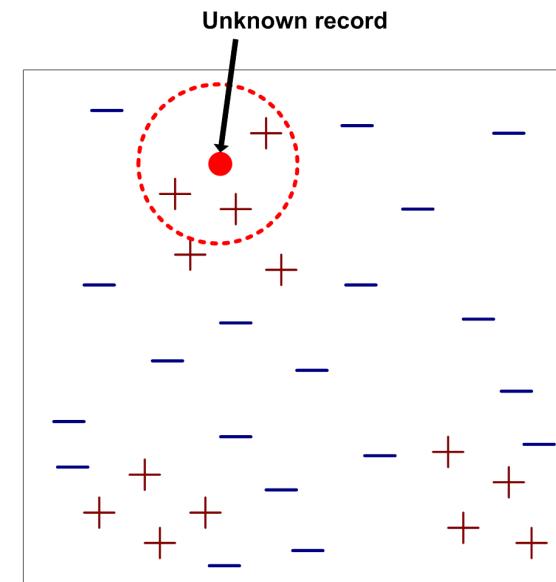
- Requires the following:
  - A set of labeled records (**Feature space**)
  - **Proximity metric** to compute distance/similarity between a pair of records
    - e.g., **Euclidean distance**
  - **The value of  $k$** , the number of nearest neighbors to retrieve
  - A **method** for using class labels of  $K$  nearest neighbors to **determine the class label** of unknown record (e.g., by taking majority vote)

چالش های این روش: ۱. چندتا از نزدیک ترین را درنظر بگیریم؟ مثلاً ۲تا از نزدیک ترین یا ۳تا یا ۴تا؟ پس مقدار  $k$  را باید معلوم کنیم.  
۲. الان فضای ۲بعدی است و میشه نزدیک ها را دید اگه فضا ۱۰۰ بعدی بود باید چیکار کنیم؟ اینجا تالاموس ۲ بعد داشت و راحت بود اگه ۵ بعد بود چی؟ نمیشه به همین راحتی رسم کرد و از روی شکل تصمیم گرفت  
پس باید روش های اندازه گیری نزدیک ترین ها را باید بهش یاد بدمیم .  
پس یه سری معیار برای فاصله باید داشته باشیم پس فاصله یابی هم یکی از چالش هاست  
۳. feature space یه چالش دیگه است ینی از بین فیچرهایی که داریم کدام ها را انتخاب کنیم؟ مثلاً فرض کنید بعلاوه ی طول و ارتفاع، یه سری ویژگی های دیگه هم از مساله بمون داده باشند پس انتخاب فیچرهای برای تصمیم گیری مهم است

# How to Determine the class label of a Test Sample?

- Take the majority vote of class labels among the k-nearest neighbors
- Weight the vote according to distance
  - weight factor,  $w = 1/d^2$

فرض کنید ما نزدیک ترین همسایه هامون را میدانیم کیا هستن حالا باید براساس این همسایه ها راجع به نقطه‌ی ناشناخته تصمیم گیری کنیم.  
یه راه اینه ببینیم توی این فاصله همسایه‌ها چه کلاس و برچسبی بیشتر از همه تکرار شده بینی رای گیری کنیم و ببینیم کدام کلاس برنده میشه.  
پس برچسب اون داده جدیده هم میداریم اون کلاسی که برنده شده.  
روش هوشمندانه تر اینه که درسته داریم به نزدیک ترین ها نگاه میکنیم ولی بهتره به صورت وزن دار نگاه کنیم.  
مثلابا یک وزنی از فاصله رای گیری کنیم از همسایه‌ها



Sure, let's walk through an example of finding the k-Nearest Neighbors (k-NN) for a test instance using the Euclidean distance metric.

Suppose we have a dataset with the following records:

Record	Feature 1	Feature 2	Class
A	2	3	Red
B	4	2	Red
C	6	4	Blue
D	4	5	Blue

We want to find the k=3 nearest neighbors of a test instance with features (3, 4). Here are the steps we could follow:

Compute the Euclidean distance between the test instance and each training instance. For example, the distances between the test instance (3, 4) and the training instances are:

```
distance(A) = sqrt((2-3)^2 + (3-4)^2) = sqrt(2)
distance(B) = sqrt((4-3)^2 + (2-4)^2) = sqrt(2)
distance(C) = sqrt((6-3)^2 + (4-4)^2) = 3
distance(D) = sqrt((4-3)^2 + (5-4)^2) = sqrt(2)
```

Sort the training instances by their distances to the test instance, in ascending order. In this example, the sorted list of training instances and their distances to the test instance are:

(A, sqrt(2))  
(B, sqrt(2))  
(D, sqrt(2))  
(C, 3)

Select the k nearest neighbors from the sorted list.

Since k=3, we select the first three instances from the sorted list:

(A, sqrt(2))  
(B, sqrt(2))  
(D, sqrt(2))

Determine the class of the test instance by majority vote among the k nearest neighbors.

In this case, all three nearest neighbors belong to the Red class, so we predict that the test instance also belongs to the Red class.

Therefore, using the k-NN algorithm with k=3 and the Euclidean distance metric, we predict that the class of the test instance (3, 4) is Red.

Sure, let's use the famous Iris dataset as an example for k-Nearest-Neighbor (k-NN) classification.

The Iris dataset contains measurements of the sepal length, sepal width, petal length, and petal width of 150 iris flowers belonging to three different species: Setosa, Versicolor, and Virginica. We will use these measurements to train a k-NN classifier that can predict the species of new iris flowers based on their measurements.

Here are the steps we would follow:

Load the dataset into memory.

Split the dataset into training and testing sets.

Choose a value for k (the number of nearest neighbors to consider).

For each test instance, compute the Euclidean distance between its feature vector and the feature vectors of all training instances.

Select the k instances with the smallest distances to the test instance.

Determine the class of the test instance by majority vote among the k nearest neighbors.

Repeat steps 4-6 for all test instances.

Compute the accuracy of the classifier by comparing its predictions with the true labels in the testing set.

Here is some sample Python code that implements this algorithm using scikit-learn library:

---

```
# import necessary libraries
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

# Load the iris dataset
iris = load_iris()

# split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target, test_size=0.2)

# create a k-NN classifier with k=3
knn = KNeighborsClassifier(n_neighbors=3)

# train the classifier on the training data
knn.fit(X_train, y_train)

# make predictions on the testing data
y_pred = knn.predict(X_test)

# compute the accuracy of the classifier
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

Handling missing values in k-NN can be challenging, but there are several approaches that can be used.

One approach is to simply remove any data points that have missing values from both the training and test sets. This is the simplest approach, but it can result in significant loss of data if many data points have missing values.

Another approach is to impute the missing values with some estimate before performing k-NN. One way to do this is to use mean or median imputation, where the missing value is replaced with the mean or median value of the feature in the training set. Another way is to use regression imputation, where a regression model is trained on the non-missing values of the feature and used to predict the missing values.

When imputing missing values, it is important to use only the training set to estimate the missing values, and not the test set. This is because using information from the test set during training can lead to overfitting and poor generalization performance.

Once the missing values have been imputed, k-NN can be applied as usual. However, it's important to keep in mind that imputing missing values can introduce bias into the model, so it's important to carefully consider the potential impact on the results.

To normalize a vector to have unit length, you need to divide each component of the vector by the magnitude of the vector. The magnitude of a vector is calculated as the square root of the sum of the squares of its components.

Here's an example: Suppose you have a vector  $v = [3, 4]$ . To normalize this vector, first you calculate its magnitude:

$$|v| = \sqrt{3^2 + 4^2} = 5$$

Then, you divide each component of the vector by its magnitude:

$$v_{\text{normalized}} = v/|v| = [3/5, 4/5]$$

Now, the resulting normalized vector has unit length, which means its magnitude is equal to 1. In mathematics, a vector is said to have unit length if its magnitude or length is equal to 1.

For example, if we have a two-dimensional vector  $v = [3, 4]$ , as I mentioned before, the magnitude of this vector is:

$$|v| = \sqrt{3^2 + 4^2} = 5$$

To normalize this vector, we divide each component by its magnitude to get the normalized vector  $v_{\text{normalized}} = [3/5, 4/5]$ . The magnitude of the normalized vector is:

$$|v_{\text{normalized}}| = \sqrt{(3/5)^2 + (4/5)^2} = 1$$

So, the normalized vector has unit length since its magnitude is equal to 1.

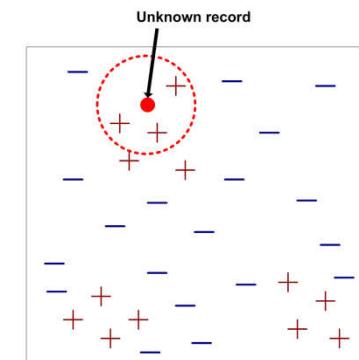
This concept of unit length is important in many areas of mathematics and science, including physics, engineering, and computer graphics, where it is used to represent directions and orientations of objects.

# Choice of proximity measure matters

- For documents, cosine is better than correlation or Euclidean

1 1 1 1 1 1 1 1 1 1 0	vs	0 0 0 0 0 0 0 0 0 0 1
0 1 1 1 1 1 1 1 1 1 1		1 0 0 0 0 0 0 0 0 0 0

Euclidean distance = 1.4142 for both pairs, but the cosine similarity measure has different values for these pairs.



معیارهای اندازه گیری فاصله وابسته به مساله است.  
انتخاب های ما مثل model selection است که قراره بین فرضیات مختلف یکی را  
انتخاب کنیم  
وقتی مدل های مختلف داریم کدام را انتخاب میکنیم؟ از داده ها ایده میگیریم  
روش اصلی: یه تعدادی از داده ها را جدا میکنیم و برای تصمیم گیری معیار فاصله  
ازشون استفاده میکنیم

here's an example of k-NN with cosine proximity:

Suppose we have a dataset of customer reviews for a set of products, where each review is represented as a vector of term frequencies. We want to use k-NN to classify new reviews as either positive or negative based on their similarity to the reviews in the training set.

To measure similarity between reviews, we can use cosine proximity, which measures the cosine of the angle between two vectors in a high-dimensional space. The cosine similarity ranges from -1 to 1, where -1 indicates opposite directions (maximal dissimilarity), 0 indicates orthogonality (no similarity), and 1 indicates parallel directions (maximal similarity).

Here's how k-NN with cosine proximity works:

Calculate the cosine similarity between the query review and each review in the training set.

Select the k reviews with the highest cosine similarity to the query review.

Compute the class label for the query review based on the majority vote of the k nearest neighbors.

Assign the query review to the predicted class.

For example, suppose we have a training set with three reviews:

Review 1: "This product is great and works well."

Review 2: "I don't like this product, it's too expensive."

Review 3: "This product is okay, but I've had better."

And suppose we want to classify the following query review:

Query Review: "I love this product, it's amazing!" We can represent each review as a vector of term frequencies, for example:

Review 1: [1, 1, 1, 1, 1, 0, 0, 0]

Review 2: [0, 0, 0, 1, 0, 1, 1, 1]

Review 3: [1, 0, 1, 1, 0, 0, 1, 0]

Query Review: [1, 0, 0, 1, 0, 0, 0, 1]

To use k-NN with cosine proximity, we can first normalize the vectors to have unit length:

Review 1: [0.41, 0.41, 0.41, 0.41, 0.41, 0.00, 0.00, 0.00]

Review 2: [0.00, 0.00, 0.00, 0.33, 0.00, 0.33, 0.33, 0.33]

Review 3: [0.53, 0.00, 0.53, 0.53, 0.00, 0.00, 0.53, 0.00]

Query Review: [0.71, 0.00, 0.00, 0.71, 0.00, 0.00, 0.00, 0.71]

Then we can calculate the cosine similarity between the query review and each training review:

Cosine Similarity:

Review 1: 0.55

Review 2: 0.13

Review 3: 0.49

Suppose we choose k=2. The two nearest neighbors to the query review are Review 1 and Review 3, both of which are positive. Thus, we predict that the query review is positive.

In the k-nearest neighbor (k-NN) algorithm, time series data can be used as input. Standardization is a preprocessing step that involves transforming the data to have a mean of 0 and a standard deviation of 1. This is also called z-score normalization.

In the context of the given statement, it means that before applying the k-NN algorithm to time series data, the time series values are first standardized by subtracting the mean value from each value and then dividing the result by the standard deviation of the values. This results in all the values being centered around zero with a consistent range. The purpose of doing this is to ensure that the different features or variables in the data are on a similar scale, so that they contribute equally to the distance metric used to find the k-nearest neighbors.

# Nearest Neighbor Classification

مثال:

- قد یک فرد ممکن است از 1.5 متر تا 1.8 متر متغیر باشد
- وزن یک فرد ممکن است از 90 پوند تا 300 پوند متغیر باشد
- درآمد یک فرد ممکن است از 10 هزار دلار تا 1 میلیون دلار متغیر باشد

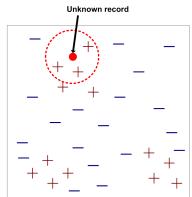
پیش پردازش داده ها اغلب مورد نیاز است  
- ممکن است برای جلوگیری از تسلط معیار های فاصله  
- توسط یکی از ویژگیها، ویژگیها باید مقیاس شوند

## ● Data preprocessing is often required

- Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes

### ◆ Example:

- height of a person may vary from 1.5m to 1.8m
- weight of a person may vary from 90lb to 300lb
- income of a person may vary from \$10K to \$1M



سری های زمانی معمولاً به گونه ای استاندارد می شوند که 0 به معنای انحراف معیار 1 باشد.

- Time series are often standardized to have 0 means a standard deviation of 1

فیچر هامون متفاوت هستند مثلاً یه فیچر جنسش طول است یکی سن و یکی وزن باشه یا مقیاس هاشون ممکن است فرق کنه مثلاً میگیم تالاموس چندمیلی متر است ولی مثلاً اگه بخاهیم راجع به شدت رنگ بافتش حرف بزنیم میگیم مثلاً ۰ تا ۲۵۵ است پس مقیاس فضایی که داریم بهش نگاه میکنیم کوچک و بزرگ است و باید در یک مقیاس بهش نگاه کنیم

وقتی داریم نزدیک ترین فاصله ها را حساب میکنیم این مقیاس های متفاوت کار را خراب میکنند. مثلاً فاصله ای اقلیدوی که میگیریم داریم به همه ی ویژگی هامون به یک شکل نگاه میکنیم در حالی که عدیزرسی ها توی فرمول فاصله موثرتر میشوند پس مقیاسشون میکنیم که جنسشون یکی شه

Yes, when we use the z-score normalization method to standardize a dataset, the resulting dataset will have a mean of 0. This is because the z-score formula involves subtracting the mean of the dataset from each data point, which centers the distribution around 0.

The formula for calculating the z-score of a data point  $x$  in a dataset with mean  $\mu$  and standard deviation  $\sigma$  is:  
$$z = (x - \mu) / \sigma$$

By subtracting the mean from each data point in the dataset, the new mean of the dataset becomes 0. Standardizing the data in this way allows us to compare and analyze different variables or features on a similar scale, regardless of their original units or scales.

When we apply the z-score normalization method to standardize a dataset, the resulting dataset will have a standard deviation of 1. This is because the z-score formula involves dividing each data point by the standard deviation of the dataset, which scales the data to have a standard deviation of 1.

By dividing each data point in the dataset by the standard deviation, the new standard deviation of the dataset becomes 1. Standardizing the data in this way allows us to compare and analyze different variables or features on a similar scale, regardless of their original units or scales.

Here is an example to demonstrate how applying the z-score normalization method results in a dataset with mean=0 and standard deviation=1:

Suppose we have a dataset of 5 numbers: {4, 6, 8, 10, 12}.

The mean of this dataset is:

$$\mu = (4 + 6 + 8 + 10 + 12) / 5 = 8$$

The standard deviation of this dataset is:

$$\sigma = \sqrt{((4-8)^2 + (6-8)^2 + (8-8)^2 + (10-8)^2 + (12-8)^2) / 5} = 2.83$$

Now, let's apply the z-score normalization method to standardize this dataset:

$$\text{z-score for 4: } (4 - 8) / 2.83 = -1.41$$

$$\text{z-score for 6: } (6 - 8) / 2.83 = -0.71$$

$$\text{z-score for 8: } (8 - 8) / 2.83 = 0$$

$$\text{z-score for 10: } (10 - 8) / 2.83 = 0.71$$

$$\text{z-score for 12: } (12 - 8) / 2.83 = 1.41$$

As you can see, after standardizing the dataset, the mean is now 0 and the standard deviation is now 1. This means that the dataset is now centered around 0 and scaled to have a consistent range, making it easier to compare and analyze the different values in the dataset.

The choice of the value of k in the k-NN algorithm can have a significant impact on the accuracy of the classifier. A small value of k (e.g., k=1) may result in overfitting, where the model is too sensitive to noise or outliers in the data, while a large value of k may result in underfitting, where the model oversimplifies the classification decision boundary and may miss important patterns in the data.

To find the best value of k for the k-NN classifier, we typically use a technique called cross-validation. The basic idea behind cross-validation is to split the available labeled data into two sets: a training set and a validation set. The model is trained on the training set, and its performance is evaluated on the validation set. This process is repeated several times with different splits of the data, and the average performance over all the splits is used as an estimate of the model's generalization performance.

Here are the general steps to find the best k for a k-NN classifier:

Split the labeled data into a training set and a validation set.

Choose a range of values for k to be tested.

For each value of k, train a k-NN classifier on the training set using that value of k.

Evaluate the performance of the classifier on the validation set using an appropriate metric, such as accuracy, precision, recall, F1 score, or area under the receiver operating characteristic (ROC) curve.

Repeat steps 1-4 several times with different splits of the data to get an estimate of the classifier's performance on unseen data.

Select the value of k that gives the best performance on the validation set.

It is important to note that the optimal value of k may vary depending on the specific dataset and problem being addressed, so it is important to choose a range of values for k that covers a broad range of possibilities in order to find the best value for the given problem.

Here's an example of how to use cross-validation to find the best value of k for a k-nearest neighbors classifier:

The GridSearchCV object automatically fits the model with each combination of hyperparameters in the parameter grid, and returns the best combination based on the validation score.

After running the code, you should see the output showing the best parameters (i.e., the value of k that maximized the validation score) and the corresponding score.

```
from sklearn.datasets import load_iris
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import GridSearchCV, train_test_split

# Load the iris dataset
iris = load_iris()

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target, test_size=0.2, random_state=42)

# Define the parameter grid
param_grid = {'n_neighbors': range(1, 11)}

# Create a k-NN classifier
knn = KNeighborsClassifier()

# Perform a grid search to find the best value of k
grid_search = GridSearchCV(knn, param_grid, cv=5)
grid_search.fit(X_train, y_train)

# Print the best parameters and score
print("Best parameters:", grid_search.best_params_)
print("Best score:", grid_search.best_score_)
```

**Best parameters: {'n\_neighbors': 6}**

**Best score: 0.975**

In this example, we used the Iris dataset and split it into training and testing sets with a 80/20 ratio. The best value of k was found to be 6 based on a 5-fold cross-validation procedure, with a corresponding validation score of 0.975.

Note that your actual results may vary slightly due to randomness in the train-test split and cross-validation folds.

# Nearest Neighbor Classification...

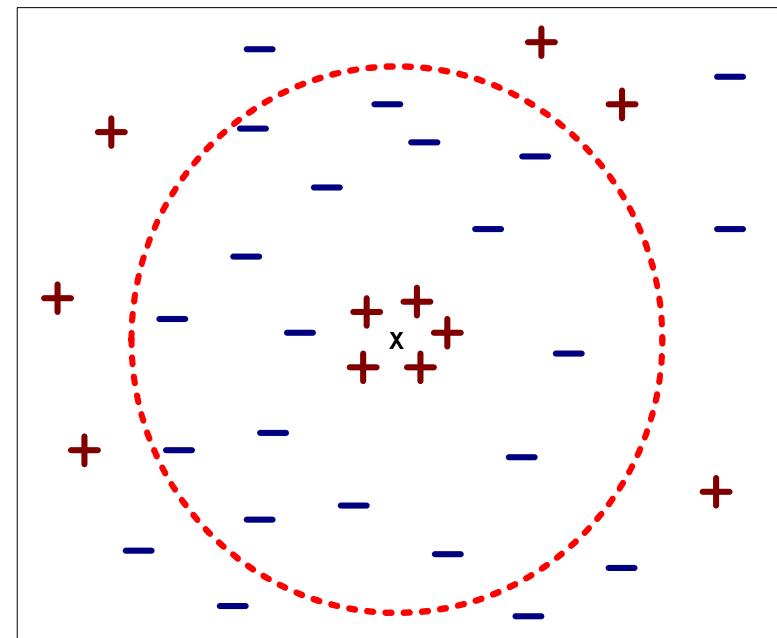
## ● Choosing the value of k:

- If  $k$  is too small, sensitive to noise points
- If  $k$  is too large, neighborhood may include points from other classes

اطلاعات کمی برای  
قضاؤت داریم استفاده  
میکنیم به نویز حساس  
میشه و ممکنه غلط  
قضاؤت کنیم

سوال:  
چندتا همسایه باید درنظر  
بگیریم؟

گه  $k$  را کوچک درنظر بگیریم یعنی تعداد همسایه ها را  
کم بگیریم، ینی اطلاعات منون برای قضاؤت کردن کم  
مثال: دادگاه های امریکا میگن شرایط این پرونده شبیه  
کدوم یکی از پرونده های قبلی است؟



به جای ۱۰ تا همسایه مثلا  
۲۰۰ کیفیم ممکنه همسایه ها به  
مورد ما مربوط نباشند  
دوباره قضاؤت منون اشتباه  
میشه  
پس یه تردید افی هست  
برای تعدادی که انتخاب  
میشه

# Nearest Neighbor Classification...

- Nearest neighbor classifiers are local classifiers

- They can produce decision boundaries of arbitrary shapes.

The statement "nearest neighbor classifiers are local classifiers" means that the k-nearest neighbor (k-NN) algorithm makes predictions based on local information, specifically by comparing each new instance to its k nearest neighbors in the training set.

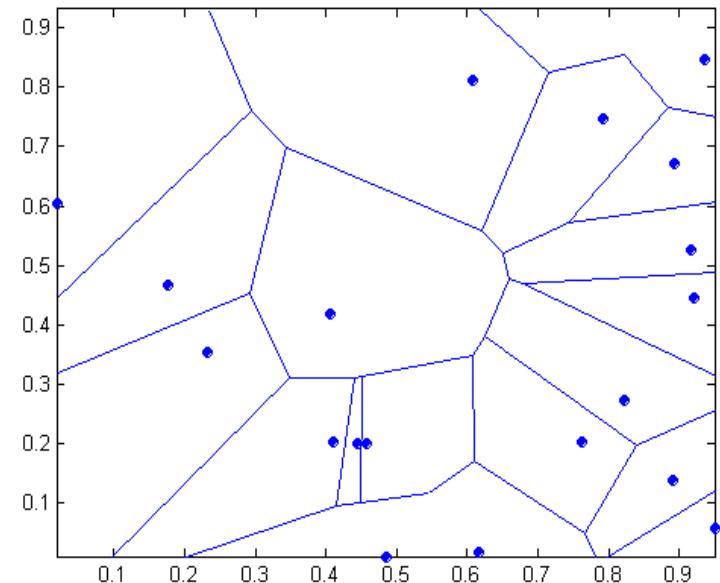
In other words, the k-NN classifier does not try to build a global model of the data distribution or decision boundary. Instead, it simply stores the entire training dataset and makes predictions for new instances based on the class labels of the k nearest training instances, which are typically measured using a distance metric such as Euclidean distance.

Because the k-NN algorithm operates locally, it can be sensitive to the structure of the data and the choice of k. Specifically, if the data has complex or nonlinear relationships, a small value of k may be more appropriate to capture the local structure, while a larger value of k may be more appropriate for smoother or simpler data distributions.

نحوه‌ی تصمیم‌گیری چطوری می‌شود  
نواحی مختلف؟ می‌شود به قانونی تعریف کرد؟  
وقتی که یک همسایه را در نظر بگیریم فقط



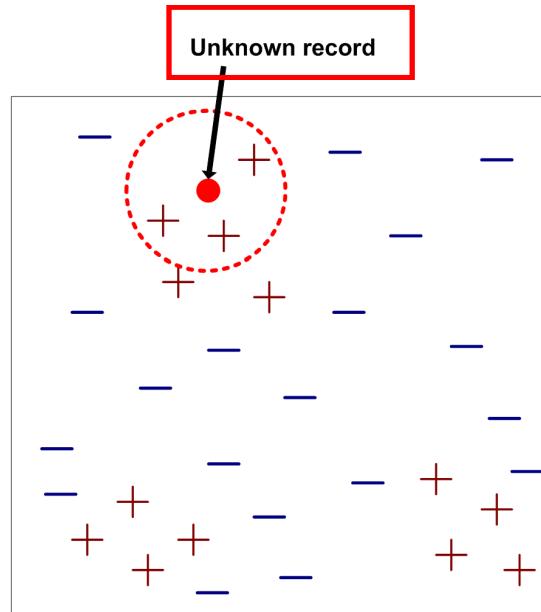
1-nn decision boundary is a Voronoi Diagram



# Nearest Neighbor Classification...

- How to handle missing values in training and test sets?

یه سری از فیچرهای را نداریم



مثالاً اگه طول های تالاموس را نداریم فقط با عرض ها فاصله سنگی را انجام بدیم در محاسبه ای نزدیکترین ها از طول مثلا استفاده نکن (برای کل داده ها!!!!!!)

در مسائل کلسیفیکیشن ممکن است با داده هایی رو برو باشیم که همه ای فیچر هاشون را ممکنه نداشته باشیم مثلاً یه تعداد بیمار هستند که فقط عرض تالاموسشون را داریم و طولش را نداریم یا اندازه گیری نکردن یا کردن و فهمیدن اشتباہ اندازه گرفتند پس با یه شرایطی رو برو هستیم که یه سری از مقادیر را نداریم سوال: آیا باید بیخیال اون داده ها بشیم یا مدلمون میتونه یه کاری برآشون بکنه؟ راه حل: فاصله ها را روی بقیه ای ویژگی ها بسنجم مثل اینه که از یک بعد خاص به داده ها نگاه کنیم این باعث میشه به یه نحوی یه جوابی بگیریم و بی جواب نمونیم که ینی داده هامون هدر نرفته

معیار فاصله حساب کردن را میبریم روی missing value ویژگی هایی که ندارند

# Nearest Neighbor Classification...

---

---

- How to handle missing values in training and test sets?
  - Proximity computations normally require the presence of all attributes
  - Some approaches use the subset of attributes present in two instances
    - ◆ This may not produce good results since it effectively uses different proximity measures for each pair of instances
    - ◆ Thus, proximities are not comparable

چگونه مقادیر از دست رفته در مجموعه های آموزشی و تست را مدیریت کنیم؟

- محاسبات مجاورت معمولاً به وجود همه صفات نیاز دارند

- برخی از رویکردها از زیرمجموعه ویژگی های موجود در دو مورد استفاده می کنند

این ممکن است نتایج خوبی ایجاد نکند زیرا به طور موثر از معیارهای مجاورت متفاوتی برای هر جفت نمونه استفاده می کند.

بنابراین، مجاورت ها قابل مقایسه نیستند

مثلاً اینجا سه بعدی  $x, y, z$  داریم و میدانیم که یکی از این بعدها نویز داره اگه فاصله را روی هر ۳تا بعد تعریف کنیم، فاصله ای ما همیشه به خاطر وجود این بعد نویزی یک مقدار ثابتی داره و یه بایاسی داره که با مقدار واقعی نمیخونه چون ما از این فاصله ها برای تصمیم گیری کلاس ها میخاهیم استفاده کنیم بهتره شاید این بعد را حذف کنیم و فاصله را فقط بر حسب  $x, y$  بسنجیم

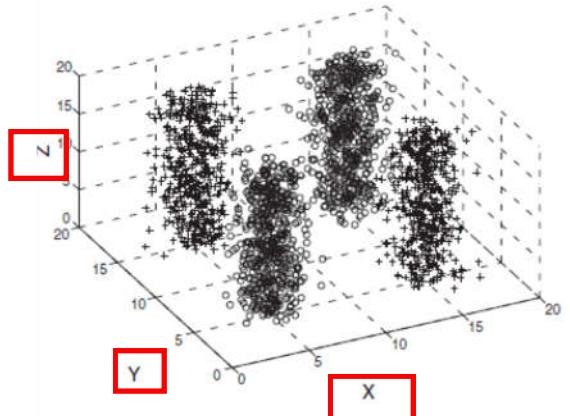
## K-NN Classifiers...

### Handling Irrelevant and Redundant Attributes

- Irrelevant attributes add noise to the proximity measure
- Redundant attributes bias the proximity measure towards certain attributes

زمانی که با ویژگی های  
نامرتبط و ویژگی های  
تکراری سروکار داشته  
باشیم

معیار فاصله حساب کردن را میبریم روی  
مقادیری که missing value ندارند



(a) Three-dimensional data with attributes  $X$ ,  $Y$ , and  $Z$ .

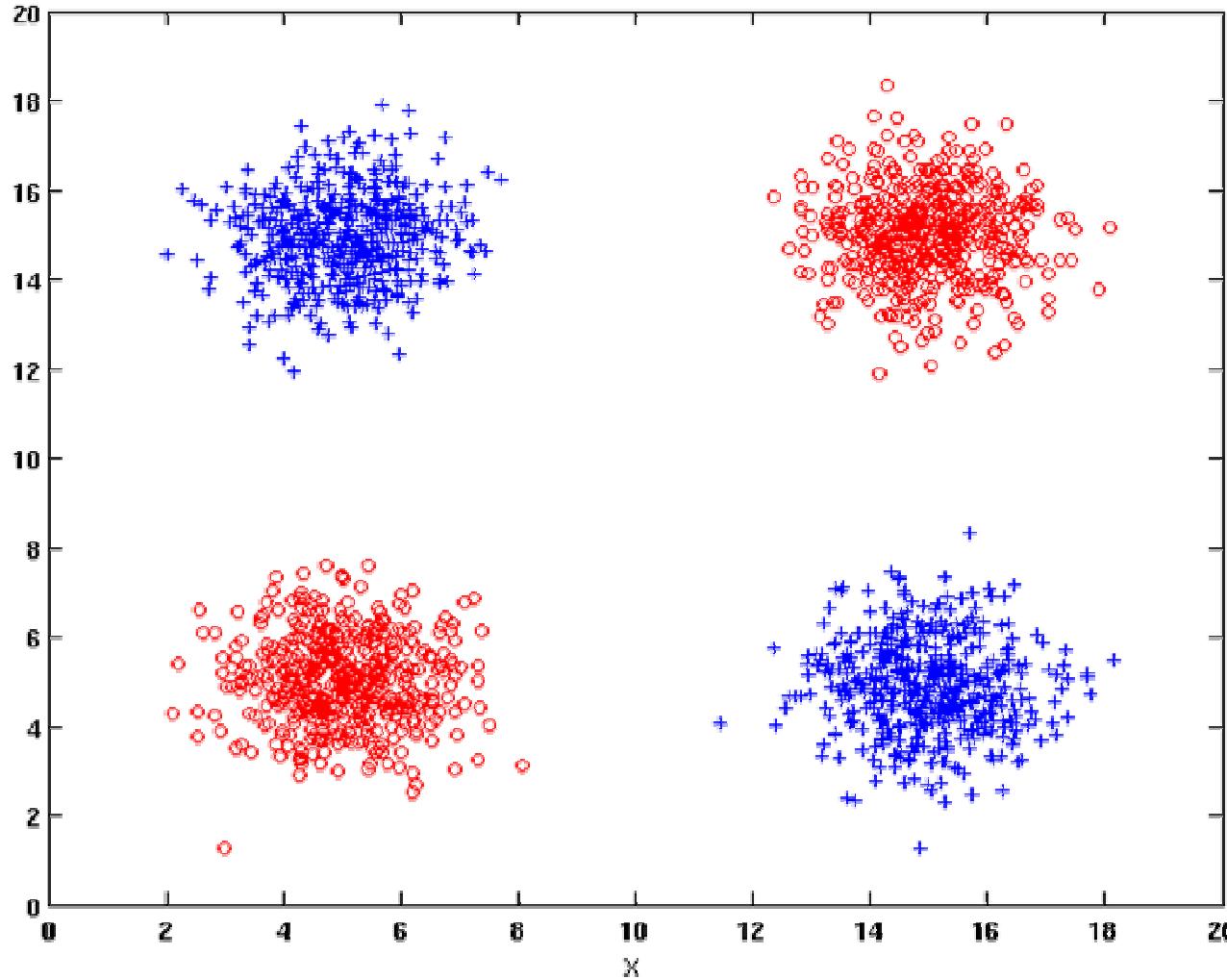
پس زمانی که با ویژگی های نامرتبط و تکراری سروکار داریم اگه آنالیز نکنیم و برآشون کاری نکنیم، مقداری که از فرمول distance بدست میاد خراب میشه و توی فاصله حساب کردن مشکل درست نمیکنه

اگه نویز روی یکی از بعدهای داده ها باشه یا یه بعدی نامرتبط باشه چه اتفاقی میفته؟ باعث میشه نتیجه بایاس شه و به یک مقداری متمایل بشه

محاسبه ای فاصله روی  $n$  بعد

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2}$$

## K-NN Classifiers: Handling attributes that are interacting



اتریبیوت هایی که به هم مرتبط هستند و وابسته هستند به یکدیگر چی؟  
یه تعامل و interaction هست و باید دو تایی بشون نگاه کنیم  
ایا در صورت وجود نویز میشه حذف کرد؟

در حالت دو بعدی فضا وقتی می خاد قسمت بندی بشه، ناحیه ناحیه میشه فضا به صورت شبکه بندی باشه

ما برای تصمیم گیری هم نیاز به ایکس داریم و هم وا پس دو تاشون را میخاهیم مثلا در مقدار ۴ هم یه کلاسی داریم که مقدارش قرمز است و هم کلاسی که مقدارش ابی است در مقدار ۱۴ هم، یه نمونه ابی و یه نمونه قرمز داریم پس کلسفیکیشن اینجا راحت نیست! از بعد لا هم همینطوره پس باید دو بعدی به داده ها نگاه کنیم

اینجا داده ها را باید از دو بعد نگاهشون کرد به یا اینکه فقط از یه بعد نگاه کنیم

# Handling attributes that are interacting

هرچه فیچرها مستقل از هم باشند بهتره  
وقتی فیچرها به هم مرتبط میشوند کار کلسفیکیشن خیلی سخت میشه



داده ها در عمل در کجای فضا قرار میگیرند؟  
اینجا به رابطه‌ی بین خود ویژگی‌ها کاری نداریم  
به محل قرارگیری شون در فضانگاه میکنیم  
نحوه‌ی قرارگیری برچسب‌ها مهم میشه

نحوه‌ی ارتباط فیچرها، روی فضایی که  
میخواهیم فاصله را حساب کنیم اثر میگذاره

k-NN is a distance-based algorithm that uses a similarity measure, typically Euclidean distance, to find the k-nearest neighbors of a query instance in the training set. However, one potential issue with distance-based algorithms like k-NN is that they can be affected by the scale and interactions between attributes.

When features are of different scales, some attributes may dominate the distance metric, leading to biased predictions. One way to handle this issue is to normalize the features to have zero means and unit variances, so that all attributes contribute equally to the distance metric.

Another issue arises when there are interacting attributes, meaning that the relationship between two attributes is not linear or additive. For example, in a dataset of housing prices, the interaction between the number of bedrooms and bathrooms may have a significant effect on the price, whereas each attribute alone may not be as predictive.

To handle interacting attributes, one approach is to perform feature engineering to create new features that capture the interactions between attributes. For example, we could add a feature for the product of the number of bedrooms and bathrooms, which may capture the non-linear relationship between these two attributes.

Alternatively, we could use a more advanced machine learning algorithm that can automatically learn non-linear relationships and interactions between attributes, such as decision trees, random forests, or neural networks. These algorithms can model complex relationships between attributes without explicitly creating new features, thus avoiding the need for manual feature engineering.

here's an example of a dataset with interacting attributes and a scatter plot showing the interaction:  
In this example, we generated a synthetic dataset with 2 input features ( $x_1$  and  $x_2$ ) and a target variable  $y$  that has an interaction term  $4*x_1*x_2$ . We added some random noise to the target variable to make it more realistic.

The scatter plot shows the relationship between  $x_1$  and  $y$  versus the relationship between  $x_1*x_2$  and  $y$ . As we can see, the interaction term captures a non-linear relationship between the two input features and the target variable, which cannot be modeled well by a simple linear model.

By including the interaction term in the model, we can capture this non-linear relationship and improve the accuracy of our predictions.

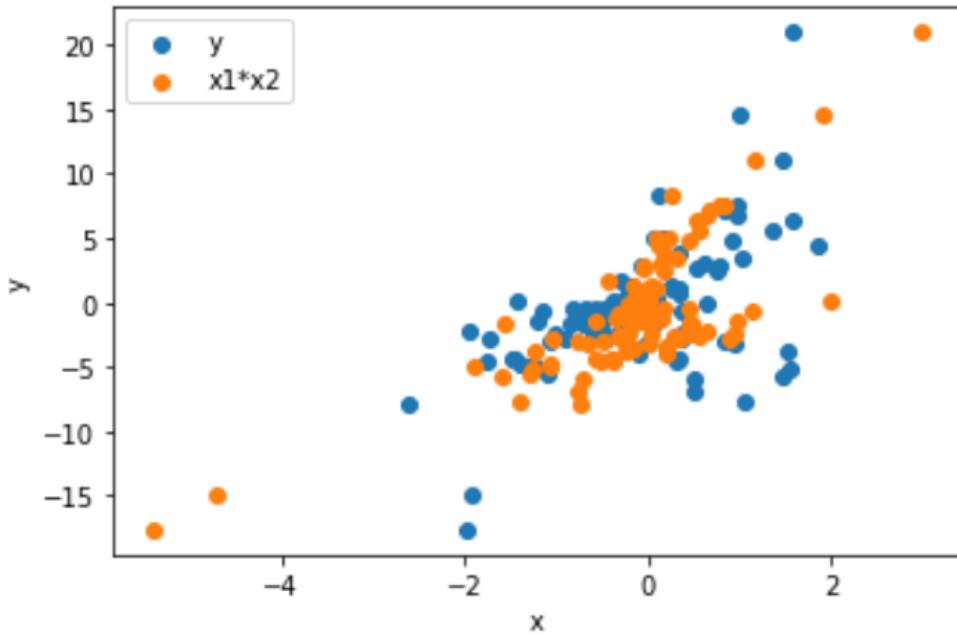
```

import numpy as np
import matplotlib.pyplot as plt

# Generate synthetic data with an interaction between x1 and x2
np.random.seed(42)
n_samples = 100
x1 = np.random.normal(size=n_samples)
x2 = np.random.normal(size=n_samples)
y = 2*x1 + 3*x2 + 4*x1*x2 + np.random.normal(scale=0.5, size=n_samples)

# Plot the interaction between x1 and x2
fig, ax = plt.subplots()
ax.scatter(x1, y, label='y')
ax.scatter(x1*x2, y, label='x1*x2')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.legend()
plt.show()

```



In the scatter plot, we can see that there is a clear non-linear relationship between  $x_1 \cdot x_2$  and  $y$ , which is not captured by the relationship between  $x_1$  and  $y$ . This suggests that including an interaction term between  $x_1$  and  $x_2$  in the model could improve its predictive accuracy.

# Improving KNN Efficiency

معمولاً در مسائل  
بیگ دینا از این  
روش‌ها استفاده می‌شود

- Avoid having to compute distance to all objects in the training set

فاصله و Distance حساب کردن زمان برو و هزینه برای نباید برای همه ابجکت‌هایی که توانی ترینینگ سمت هست بایام حسابشون کنیم

- Multi-dimensional access methods (k-d trees)
- Fast approximate similarity search
- Locality Sensitive Hashing (LSH)

## Condensing

- Determine a smaller set of objects that give the same performance

متراکم شدن  
- مجموعه کوچکتری از اشیاء را که عملکرد یکسانی دارند تعیین کنید

## Editing

ویرایش  
- برای بهبود کارایی اشیاء را حذف کنید

از محاسبه فاصله تا تمام اشیاء در مجموعه آموزشی خودداری کنید  
- روش‌های دسترسی چند بعدی (درخت k-d)  
- جستجوی سریع تشابه تقریبی  
- محل هش حساس (LSH)

- Remove objects to improve efficiency

میشه به جای اینکه تک تک رکوردها را توانی دیتابیس ذخیره کنیم، خوش خوش ذخیره کنیم (برای حل مشکل فاصله حساب کردن های خیلی زیاد و محاسبات خیلی بالا میشه نمونه‌های را فشرده کنیم) بعد وقتی یه نمونه برای مقایسه اومد به جای مقایسه باهمه ی رکوردها، فقط با نماینده‌ی اون نمونه‌ها مقایسه را انجام میدیم و سرعت مقایسه کردن بالا میره

پاید یه سرچی انجام بدیم و فاصله‌ی نمونه فعلی را بانمونه های قبلی بسنجدیم و چون اردر عملیات سرچ بالاست، برای فاصله حساب کردن معمولاً از رویکردهای هشینگ استفاده میکنند

There are several techniques that can be used to improve the efficiency of KNN algorithm:

**Feature Selection:** KNN algorithm is sensitive to irrelevant and redundant features, therefore selecting only the most relevant features can significantly reduce the dimensionality of the problem and subsequently improve the efficiency.

**Distance Metrics:** The choice of distance metric can have a significant impact on the performance of KNN algorithm. For instance, using Euclidean distance may not work well for high-dimensional data, whereas using Cosine similarity or Mahalanobis distance may yield better results.

**KD-Tree:** A KD-tree is a binary tree structure that recursively partitions data into smaller regions based on the distance measure. KD-trees can be used to speed up the search process in KNN algorithm by reducing the number of distance calculations required.

**Approximate Nearest Neighbor:** The approximate nearest neighbor (ANN) algorithm is a technique that trades off accuracy for speed. ANN algorithms can be used to efficiently find an approximate solution to KNN problem.

**Parallelization:** KNN algorithm can benefit from parallel processing techniques such as multi-threading or distributed computing. This can help to reduce the computation time for large datasets.

By employing these techniques, it is possible to significantly improve the efficiency of KNN algorithm without sacrificing its accuracy.