

# **PROCESAREA CUNOȘTINȚELOR ȘI CALCUL INTELIGENT - PCCI**

---

**Introducere în calculul inteligent**

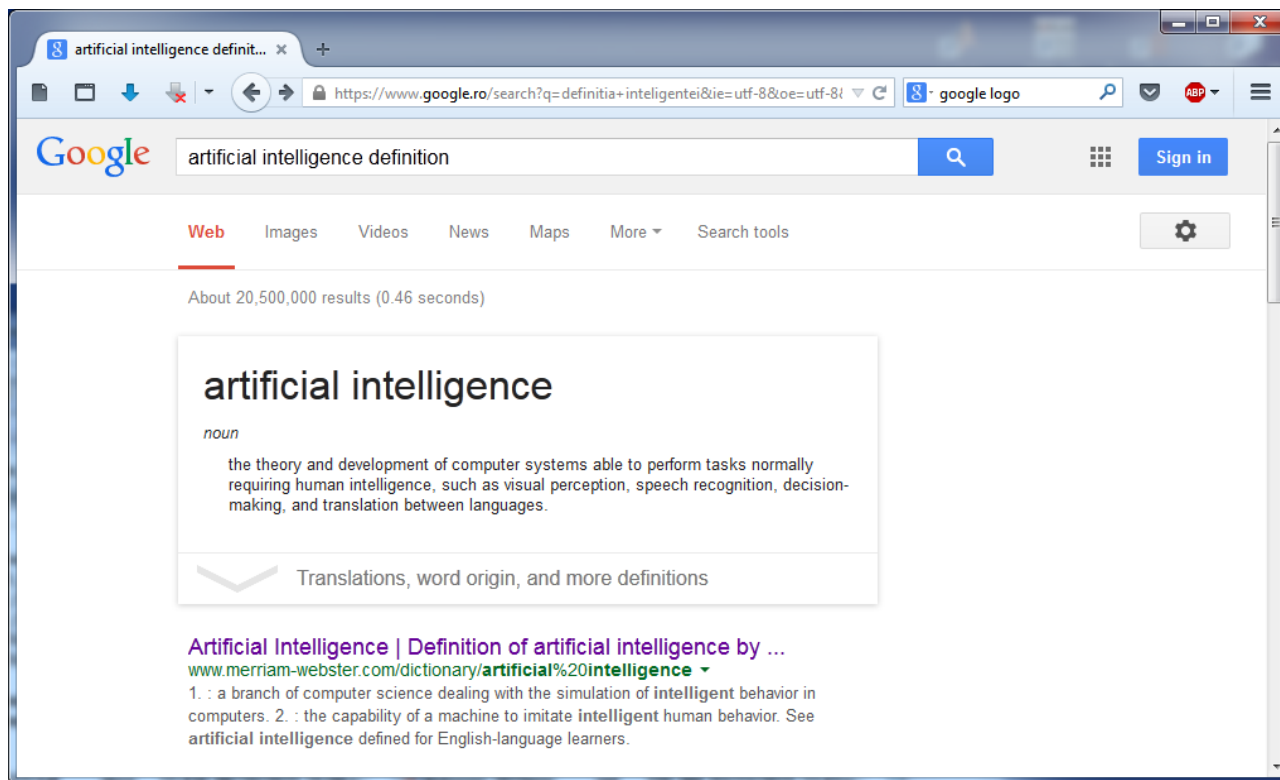
# Obiectivele cursului

- Prezentarea teoretică a conceptelor și algoritmilor de calcul inteligent, cu aplicații în electroenergetică.
- Calculul inteligent este un subdomeniu al inteligenței artificiale.

# Ce este inteligența ?

- **INTELIGENȚĂ** = capacitatea de a înțelege ușor și bine, de a sesiza ceea ce este esențial, de a rezolva situații sau probleme noi pe baza experienței acumulate anterior.
- **INTELIGENȚĂ ARTIFICIALĂ** = domeniu al informaticii care dezvoltă sisteme tehnice capabile să rezolve probleme dificile legate de inteligența umană.

# Inteligența artificială



“teoria și implementarea unor sisteme de calcul capabile să rezolve sarcini care necesită inteligență umană, precum percepția vizuală, recunoașterea vorbirii, luarea deciziilor, traducerea din limbi străine”

# Aplicații ale inteligenței artificiale

- Căutarea pe Google folosește un algoritm de inteligență artificială (în prezent, algoritmul Panda, introdus în 2011)

## What is Artificial Intelligence (AI)? A Webopedia Definition

[www.webopedia.com](http://www.webopedia.com) > TERM > A ▼

Artificial intelligence is the branch of computer science concerned with making computers behave like humans. The term was coined in 1956 by John McCarthy at the Massachusetts Institute of Technology.

## Artificial intelligence | Define Artificial intelligence at ...

[dictionary.reference.com/browse/artificial+intelligence](http://dictionary.reference.com/browse/artificial+intelligence) ▼

the capacity of a computer to perform operations analogous to learning and decision making in humans, as by an expert system, a program for CAD or CAM, or a ...

## What is AI (Artificial Intelligence)? - Definition from WhatIs.com

[searchcio.techtarget.com](http://searchcio.techtarget.com) > BPM ▼

AI (pronounced AYE-EYE) or artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems.

## Artificial intelligence - Wikipedia, the free encyclopedia

[https://en.wikipedia.org/wiki/Artificial\\_intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence) ▼

Major AI researchers and textbooks define this field as "the study and design of intelligent agents", in which an intelligent agent is a system that perceives its ...

## Machine learning - Wikipedia, the free encyclopedia

[https://en.wikipedia.org/wiki/Machine\\_learning](https://en.wikipedia.org/wiki/Machine_learning) ▼

This definition is notable for its defining machine learning in fundamentally ... The field changed its goal from achieving artificial intelligence to tackling solvable ...  
You visited this page.

## Artificial intelligence - definition of artificial intelligence by ...

[www.thefreedictionary.com/artificial+intelligence](http://www.thefreedictionary.com/artificial+intelligence) ▼

The ability of a computer or other machine to perform those activities that are normally thought to require intelligence. 2. The branch of computer science ...

## What is Artificial Intelligence (AI)? - Definition from Techopedia

[www.techopedia.com/definition/190/artificial-intelligence-ai](http://www.techopedia.com/definition/190/artificial-intelligence-ai) ▼

Artificial Intelligence AI Definition - Artificial intelligence (AI) is an area of computer science that emphasizes the creation of intelligent machines...

- ❖ identificarea conținutului util al unei pagini în funcție de cuvinte cheie și sortarea lor în funcție de relevanță
- ❖ învățarea din experiența anterioară - **machine learning** (învățare automată) este o aplicație a inteligenței artificiale

# Aplicații ale inteligenței artificiale

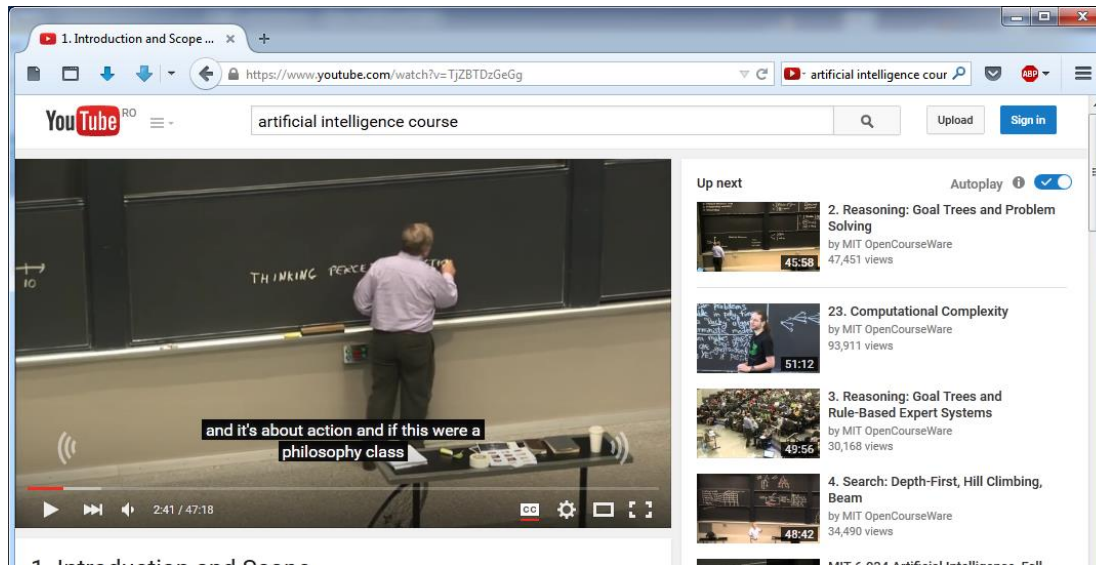


[sursă imagine: Wikipedia]

- ❖ Recunoașterea imaginilor  
(image processing, computer vision)
- ❖ Recunoașterea cuvintelor  
(speech recognition, natural language processing)

- Proces instantaneu pentru creierul uman, dar foarte dificil pentru un calculator

# Aplicații ale inteligenței artificiale



[<https://www.youtube.com/watch?v=TjZBTDzGeGg>]

❖ Traducerea automată  
(YouTube sau Google)

Translate



English Spanish French English - detected ▼



English Spanish Romanian ▼

Translate

the theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages



teoria și dezvoltarea de sisteme informatice capabile să îndeplinească sarcini în mod normal necesită inteligența umană, cum ar fi percepția vizuală, recunoașterea vorbirii, de luare a deciziilor, precum și traducerea între limbi

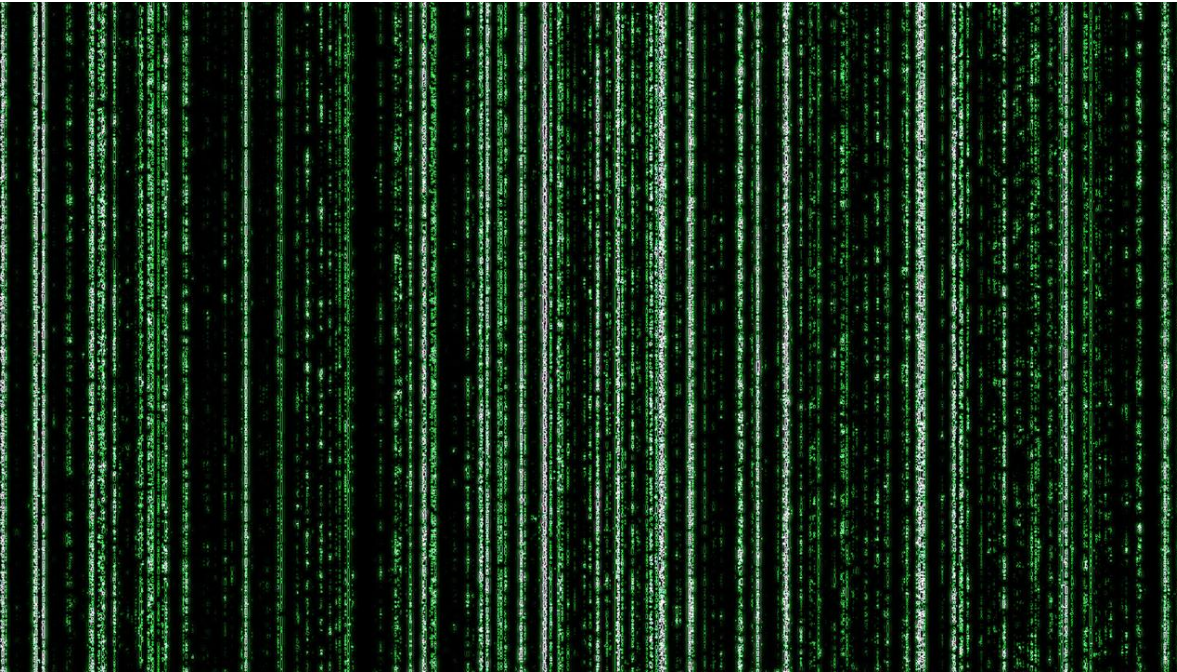
✎ Wrong?

# Aplicații ale inteligenței artificiale

The screenshot shows the Google Play Books interface. At the top, there's a browser address bar with the URL [https://play.google.com/store/books/details?id=IZoscgIMJUC&rdid=book-IZoscgIMJUC&rdot=1&source=gbv\\_vpt\\_read&pcampaignid=books\\_booksearch\\_viewport](https://play.google.com/store/books/details?id=IZoscgIMJUC&rdid=book-IZoscgIMJUC&rdot=1&source=gbv_vpt_read&pcampaignid=books_booksearch_viewport). Below the browser bar is a Google Play header with a search bar and navigation links: Genres, Home, Top Charts, and New Arrivals. The main content area displays the book 'Computational Intelligence: An Introduction, Edition 2' by Andries P. Engelbrecht, published by John Wiley & Sons on October 22, 2007. The book has a 4.5-star rating. A 'Free sample' button is available, along with a price of RON388.44. A 'Read more' button is also present. To the right, a 'Similar' section lists other books, including 'Swarm Intelligence' by Mauro Birattari and 'Computational Intelligence' by Diego Andina. The book description states: 'Computational Intelligence: An Introduction, Second Edition' offers an in-depth exploration into the adaptive mechanisms that enable intelligent behaviour in complex and changing environments. The main focus of this text is centred on the computational modelling of biological and natural intelligent systems, encompassing swarm intelligence, fuzzy systems, artificial neural networks, artificial immune systems and evolutionary computation. Engelbrecht provides readers with a wide knowledge of Computational Intelligence (CI) paradigms.

- ❖ Sisteme de recomandare – fac sugestii pe baza datelor personale ale utilizatorului sau a istoricului de căutare al altor utilizatori

# Aplicații ale inteligenței artificiale



- ❖ Data mining, big data – extragerea informațiilor relevante dintr-un volum mare de date: citiri primite de la senzori, profile de utilizator pe web

[ Bensik Imeri/freeimages.com ]

# Aplicații ale inteligenței artificiale

- ❖ Detectarea anomaliilor
- ❖ Creativitatea artificială
- ❖ Automatizarea deciziilor
- ❖ Calcul inteligent

# Calcul inteligent

- ❖ Algoritme capabile să reproducă mecanisme naturale, să se adapteze la schimbări și să învețe din experiență:
  - ❖ rețele neuronale artificiale (artificial neural networks) - simulează attribute ale creierului uman: învățarea, descoperirea legăturilor într-un noian de informații, generalizarea, recunoașterea și clasificarea;
  - ❖ algoritme evolutive (evolutionary algorithms) - explorarea, adaptarea la condiții schimbătoare, supraviețuirea celui mai bine adaptat;
  - ❖ agenți inteligenți - algoritmul furnicii (the ant algorithm) și optimizarea deplasării roiurilor (Particle Swarm Optimization) - agenți care învață interacționând cu mediul și din experiența predecesorilor
  - ❖ logica fuzzy (fuzzy logic, fuzzy systems) - tratarea informațiilor vagi

# Bibliografie:

- **Mihai Gavrilaș** – Inteligență artificială cu aplicații în energetică, vol. I, II, Editura Gh Asachi, Iași
  - **Andries P. Engelbrecht** – Computational Intelligence – An introduction, John Wiley & Sons, 2007
  - **Resurse online:**
  - **Christopher MacLeod** – An Introduction to Practical Neural Networks and Genetic Algorithms for Engineers and Scientists
- 
- YouTube** • Machine Learning – Stanford University, prof. **Andrew Ng**
  - YouTube** • Neural Networks for Machine Learning - Toronto University, prof. **Geoffrey Hinton**

# Terminologie

- **Set de date** – un grup sau o colecție de informații alcătuite din elemente distincte între care există o relație (exemple: un coș de mere sau un grup de consumatori de energie electrică), procesate de calculator.
  - ❖ Un element dintr-un set de date (un măr sau un consumator fizic) se numește **model**.
    - ❖ Caracteristicile fizice ale unui obiect (parametri, ipoteze de lucru) reprezintă un **model de intrare**.
      - ❖ Ex: măr mare, colorat, moale
      - ❖ Ex: consumator care locuiește într-un apartament la periferia orașului și consumă 90 kWh de energie electrică într-o lună.
    - ❖ Concluziile trase asupra obiectului respectiv (rezultatul) reprezintă un **model de ieșire**.
      - ❖ Ex: măr copt
      - ❖ Ex.: consumator mic.
  - ❖ Ipotezele de lucru ale problemei, (dimensiune, culoare, consistență), (consumul lunar, localizarea față de centrul orașului) se numesc **parametri** sau **caracteristici** (features); pe baza lor, printr-o transformare sau funcție, se obține rezultatul.



- ❖ Valori diferite ale parametrilor de intrare introduse în aceeași funcție pot determina rezultate diferite.

# Procesarea informației în creier

- Creierul stabilește rezultatul prin raționament bazat pe experiența anterioară



- ❖ Determină cât de copt este un măr, astfel:
- ❖ observă dimensiunea, intensitatea culorii și consistența la atingere a mărului;
- ❖ “consultă” memoria pe termen lung, experiența anterioară despre cum trebuie să arate un măr copt, comparând “șabloanele” din creier cu realitatea;
- ❖ Trage concluzia dacă mărul este copt sau nu.



- ❖ Identifică tipul unui consumator de energie electrică, astfel
- ❖ Citește suprafața casei și coordonatele geografice ale casei consumatorului
- ❖ Consultă un normativ sau un regulament
- ❖ Stabilește tipul consumatorului în funcție de limitele de consum prevăzute în normativ sau regulament și localizarea casei.

# Procesarea informației în calculator

- Calculatorul reprezintă altfel informațiile: prin *parametri*, *coeficienți* și *funcții*.

- ❖ Parametrii sunt caracteristicile fizice luate în calcul pentru rezolvarea problemei date

- Pentru un măr: culoarea ( $x_1$ ), mărimea ( $x_2$ ), moliciunea ( $x_3$ )
- Pentru un consumator: consumul lunar ( $x_1$ ) și distanța față de centrul orașului ( $x_2$ )

- ❖ Coeficienții reprezintă ponderile (importanța) asociată parametrilor

- ❖ culoarea mărului poate conta în proporție de 50% ( $a_1$ ), mărimea în proporție de 10% ( $a_2$ ), iar moliciunea, în proporție de 40% ( $a_3$ )
- ❖ Coeficienții pot fi și penalizatori (când au semnul  $-$ , de exemplu)

- ❖ Parametrii trebuie să exprime realitatea fizică sub formă de numere (de ex. o valoare a culorii între 0 – minim și 1 – maxim, respectiv un consum).

- ❖ Pentru mere, putem scrie următoarea funcție:

*grad coacere = pondere culoare · culoare + pondere marime · marime + pondere moliciune · moliciune*

$$y = a_1 \cdot x_1 + a_2 \cdot x_2 + a_3 \cdot x_3 = 0.5 \cdot x_1 + 0.1 \cdot x_2 + 0.4 \cdot x_3$$

$$\text{mar aproape copt} : 0.73 = 0.5 \cdot 0.8 + 0.1 \cdot 0.5 + 0.4 \cdot 0.7$$

$$\text{mar necopt} : 0.21 = 0.5 \cdot 0.2 + 0.1 \cdot 0.3 + 0.4 \cdot 0.2$$

$$\text{mar rascopt} : 1.12 = 0.5 \cdot 1.1 + 0.1 \cdot 0.9 + 0.4 \cdot 1.2$$

# Procesarea informației în calculator

- ❖ Pentru consumatorul de energie electrică, putem scrie următoarea funcție:

$$tip\ consumator = pondere\ consum \cdot consum + pondere\ distanta \cdot \frac{1}{distanta}$$

$$y = a_1 \cdot x_1 + a_2 \cdot \frac{1}{x_2}$$

- ❖ Cu cât  $y$  este mai apropiat de 1, cu atât consumatorul este mai mare;
- ❖ Această funcție exprimă faptul că localizarea departe de centrul orașului ar scădea probabilitatea ca un consumator să fie mare (raționament fals, dar posibil a fi codificat matematic).
- ❖ Uneori, parametrii unei probleme se exprimă printr-o altă mărime, opusă, mai convenabilă
  - ❖ În cazul merelor, am folosit parametrul moliciune, care este inversul consistenței ( $consistenta = 1/moliciune$ )

# Avantajul abstractizării matematice

- Abordările matematice prezentate pentru problema merelor și cea a consumatorilor de energie electrică sunt similare.
- Dacă există un algoritm care o rezolvă pe prima, el este aplicabil și celei de-a doua, deși cele două probleme fac parte din domenii complet diferite de activitate (agricultură, respectiv piața de energie electrică).

# Reprezentarea matematică a informației

- Numere (întregi, reale, complexe)
- Vectori (șiruri de numere):
  - ❖ Un model de intrare care codifică prin coeficienții  $a_1$ ,  $a_2$  și  $a_3$  caracteristicile unui măr (culoare 0.8 – intensă, mărime 0.5 - potrivită, moliciune 0.7 - peste medie) poate fi reprezentat
    - sub forma unui vector-linie  $[0.8 \ 0.5 \ 0.7]$
    - Sub forma unui vector-coloană  $\begin{bmatrix} 0.8 \\ 0.5 \\ 0.7 \end{bmatrix}$
- Matrice:
  - ❖ O bază de date care conține mai multe mere (de exemplu, 5 mere, fiecare linie a matricei reprezentând un măr) poate fi scrisă sub forma unei matrice.
- Șiruri de caractere (string)

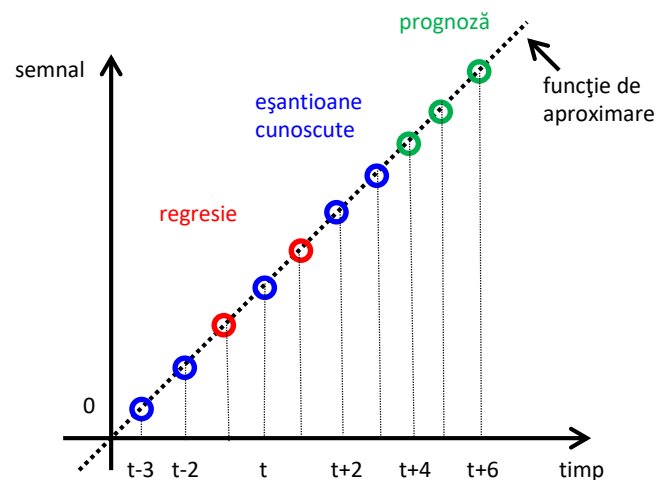
$$\begin{bmatrix} 0.8 & 0.5 & 0.7 \\ 0.3 & 0.2 & 0.2 \\ 1.2 & 0.9 & 1.2 \\ 0.5 & 0.3 & 0.2 \\ 1.0 & 0.8 & 0.8 \end{bmatrix}$$

# Reprezentarea matematică a informației

- Scrierea vectorială este preferată din mai multe motive:
  - ❖ Este compactă și ordonată.
  - ❖ Este, în multe cazuri, mai intuitivă.
  - ❖ Mediile de programare moderne procesează mai eficient vectorii și matricele, comparativ cu utilizarea ciclurilor FOR pentru realizarea aceluiași calcule.

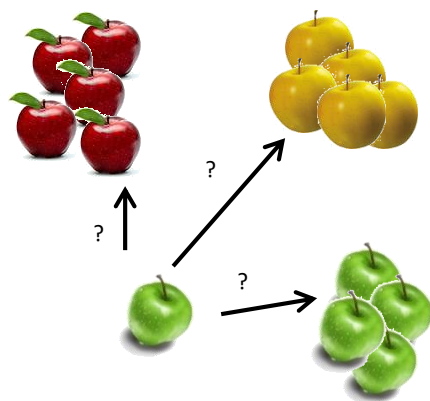
# Probleme frecvente în inginerie

- Aproximarea funcțiilor sau a seriilor de date: identificarea expresiei analitice aproximative pe baza unor eșantioane  $x - f(x)$ 
  - Regresia sau interpolarea: Cunoaștem valoarea unui semnal în intervalele de timp  $(t-3)$ ,  $(t-2)$ ,  $(t)$ ,  $(t+2)$ ,  $(t+3)$ . Determinând expresia funcției care exprimă variația semnalului în timp, pot fi calculate valorile semnalului în intervalele lipsă  $(t-1)$  sau  $(t+1)$ .
  - Prognoza: estimarea valorii semnalului în viitor, în intervalele de timp  $(t+4)$  sau  $(t+6)$ .



# Clasificare

- Clasificare = repartizare a unui obiect (model de intrare) în una dintre mai multe categorii cunoscute.
  - ❖ Într-un set de date se cunosc mai multe categorii (mere rosii, verzi, galbene sau consumatori de energie electrică mici, mijlocii, mari).
  - ❖ Dat un obiect (măr sau consumator) nou, trebuie determinat cărei clase existente aparține (sau dacă trebuie creată o clasă nouă).



Cărei clase îi aparține noul măr ?

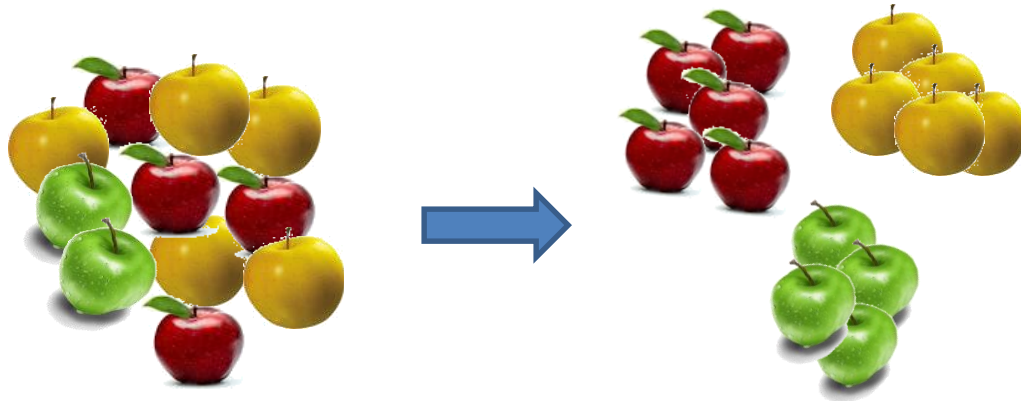
- ❖ Alte probleme de clasificare: recunoașterea formelor, OCR, identificarea mesajelor e-mail spam

# Clustering

- Clustering (grupare) = identificarea categoriilor existente într-un grup de date (modele de intrare) despre care inițial nu se cunoaște nimic.
  - ❖ Dintr-un set de date (mere, consumatori), se identifică categoriile de mere (roșii, galbene și/sau verzi) sau consumatori (mici, mijlocii și/sau mari) și numărul de modele



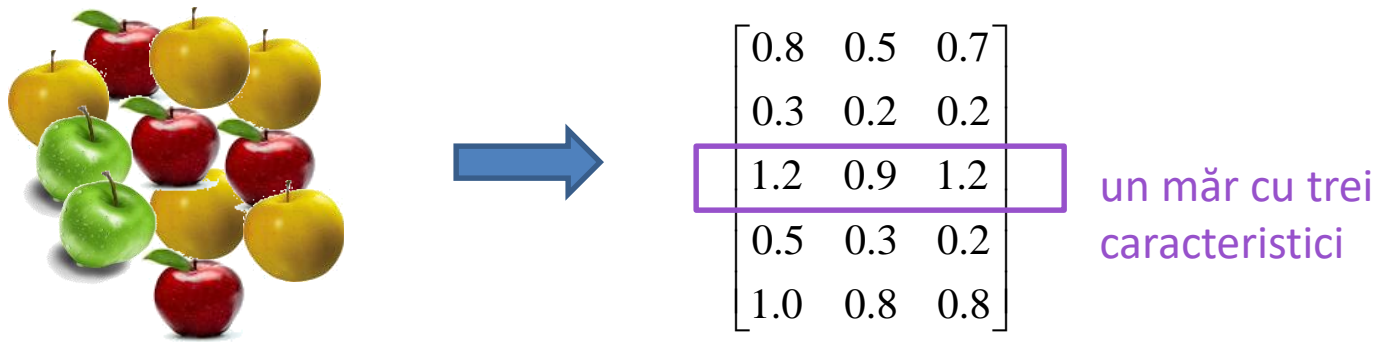
Câte feluri de mere sunt în grămadă ?  
Cum arată fiecare fel de măr ?  
Care măr aparține cărui fel ?



**Algoritme de clasificare  
și clustering**

# Generalități

- Cu elementele supuse clasificării/grupării, este formată o bază de date.
- Caracteristicile fizice ale elementelor sunt codificate în numere.



*Un măr este reprezentat de un vector de numere.*

- Pentru a realiza clasificarea sau gruparea, trebuie găsită o metodă de comparare a elementelor care compun baza de date.

# Distanțe

- Metoda utilizată cel mai frecvent pentru compararea vectorilor este **distanța**.
- Există mai multe moduri de a calcula distanța (selectiv):
  - ❖ Distanța euclidiană (norma 2)
  - ❖ Distanța Manhattan (norma 1)
  - ❖ Distanța după norma  $\infty$
  - ❖ Distanța Hamming

# Norme

- Pentru doi vectori oarecare  $X = [x_1, \dots, x_i, \dots, x_n]$  , se definesc:

$$Y = [y_1, \dots, y_i, \dots, y_n]$$

- ❖ Distanța după norma 1

$$\|X - Y\|_1 = |x_1 - y_1| + \dots + |x_i - y_i| + \dots + |x_n - y_n| = \sum_{i=1}^n |x_i - y_i|$$

- ❖ Distanța după norma 2

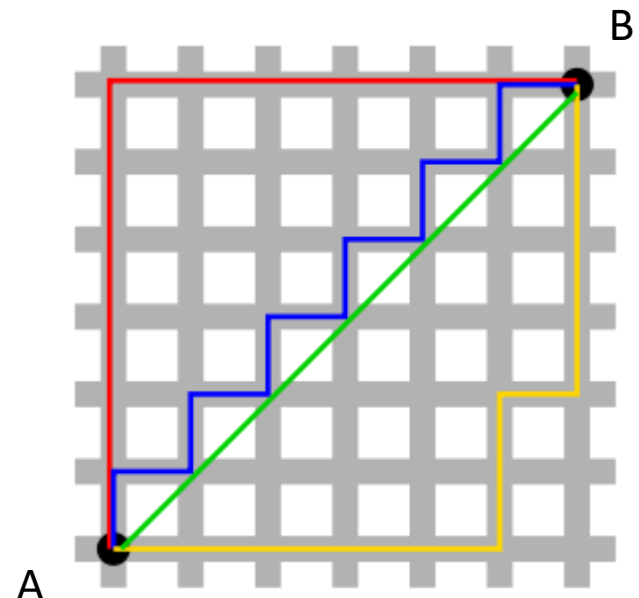
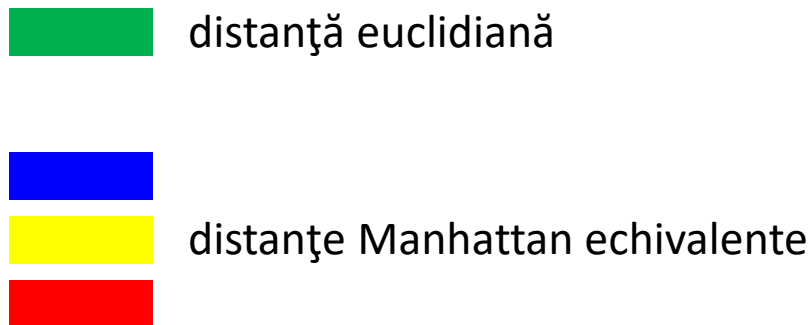
$$\|X - Y\|_2 = \sqrt{(x_1 - y_1)^2 + \dots + (x_i - y_i)^2 + \dots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

- ❖ Distanța după norma  $\infty$

$$\|X - Y\|_\infty = \max_{i=1..n} |x_i - y_i|$$

# Interpretări geometrice

- Distanța Manhattan și distanța euclidiană:
  - ❖ Euclidiană: teorema lui Pitagora
  - ❖ Manhattan (taxicab): suma proiecțiilor pe cele două axe, orizontală și verticală



# Norme

## ❖ Distanța după norma Hamming

- Se aplică pentru vectorii binari, alcătuiți doar din elemente 0 și 1
- Calculează diferența totală dintre cei doi vectori

$$\|X - Y\|_H = |x_1 - y_1| + \dots + |x_i - y_i| + \dots + |x_n - y_n| = \sum_{i=1}^n |x_i - y_i|$$

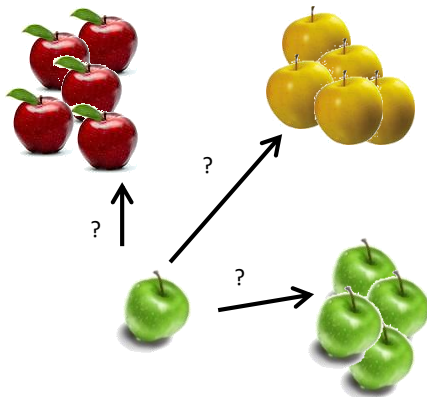
vector 1	1	0	0	1	1	0	1
vector 2	1	1	0	1	1	1	0
diferența	0	1	0	0	0	1	1
Distanța Hamming	Suma diferențelor: 3						

# Algoritme de clasificare și clustering

- Algoritm de clasificare: *algoritmul celor mai apropiați k vecini* (k-nearest neighbors, acronim kNN)
- Algoritm de clusterizare: *k-medii* (k-means)

# Algoritmul kNN

- Cel mai simplu algoritm de clasificare: *algoritmul celor mai apropiați  $k$  vecini*
- Urmărește să determine cărei clase dintre cele cunoscute aparține un model de intrare nou.



Cărei clase îi aparține noul măr ?

# Algoritmul kNN

- Algoritmul parcurge următorii pași:
- Definirea bazei de date: se dă un număr de elemente pentru care se cunoaște clasa căreia aparține fiecare dintre ele.
- Definirea numărului de vecini  $k$
- Prezentarea noului element, care se dorește a fi clasificat
- Calculul distanță (de obicei euclidiană) dintre noul element și elementele cunoscute din baza de date
- Sortarea în ordine crescătoare a distanțelor și reținerea primelor  $k$  elemente cu cele mai mici distanțe calculate
- Identificarea claselor cărora aparțin cele  $k$  elemente reținute
- Clasa căreia îi va fi asociat noul element va fi clasa dominantă (care apare de cele mai multe ori)

# Algoritmul kNN

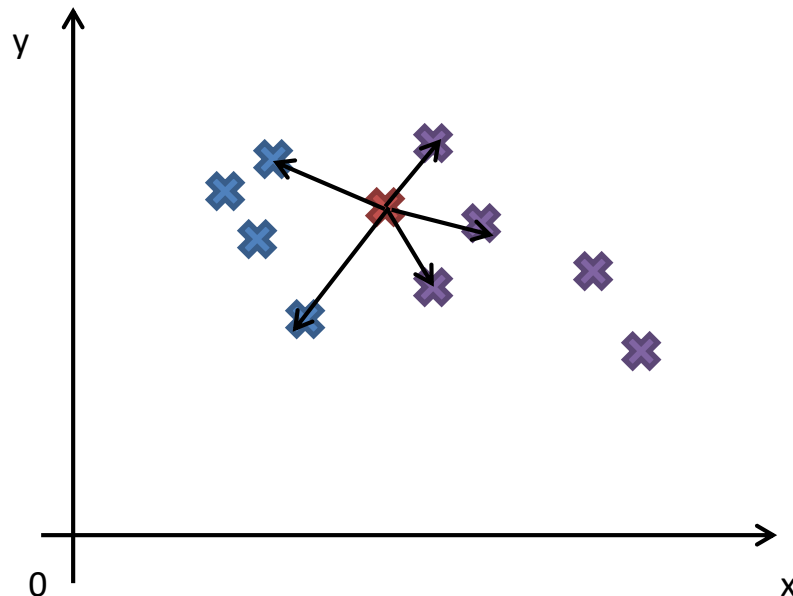
- Este un algoritm “leneș” (lazy), deoarece nu are o etapă de antrenare, se trece direct la recunoaștere.
- Este un algoritm non-parametric, adică nu consideră nicio distribuție specială a datelor de intrare, luându-le ca atare.
- Reprezentare geometrică a principiului de funcționare:

✕ Clasa 1

✕ Clasa 2

Dacă  $k=5$ , noul element ar  
aparține clasei 2

Dacă  $k=7$ , noul element ar  
aparține clasei 1

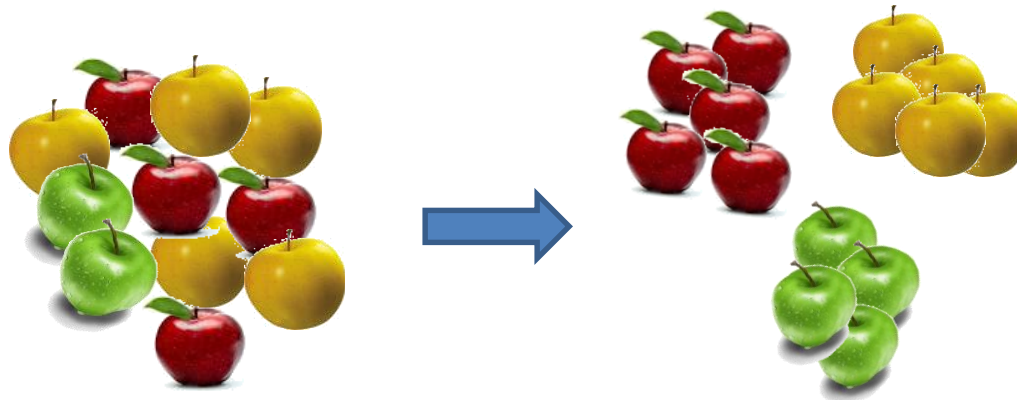


# Algoritmul kNN - avantaje și dezavantaje

- Avantaje:
  - ❖ Este ușor de implementat software, necesită cod puțin și simplu.
  - ❖ Dă rezultate bune dacă setul inițial de date cunoscute este mare și reprezentativ.
- Dezavantaje:
  - ❖ Rezultatele depind de alegerea corectă a numărului  $k$ .
  - ❖ Lucrează cu un număr fix de clase, cel inițial. Nu poate descoperi clase noi.
  - ❖ Nu permite definirea unui “reprezentant-tip” pentru clase.
  - ❖ Poate da rezultate greșite dacă setul inițial de date este foarte mic, nereprezentativ.
- Când  $k=1$  - algoritmul celui mai apropiat vecin (1NN)

# Algoritmul k-means

- Algoritm de clusterizare – *k-medii* (k-means)
- Urmărește să identifice clasele de elemente care există într-o bază de date nesortată.

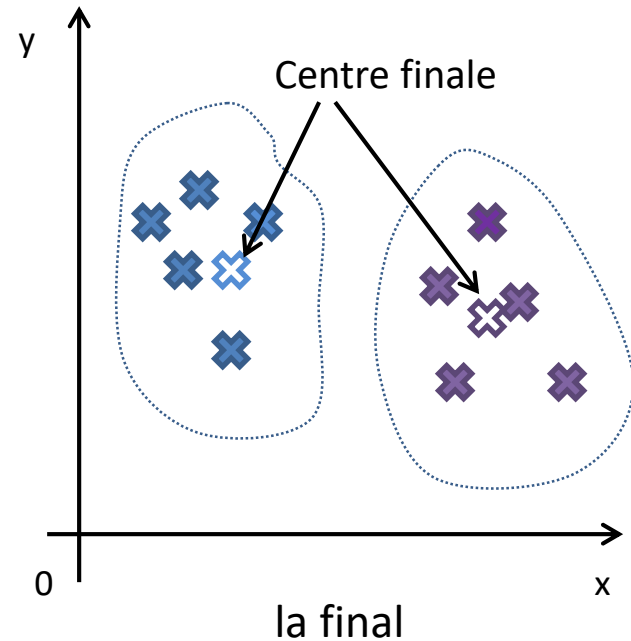
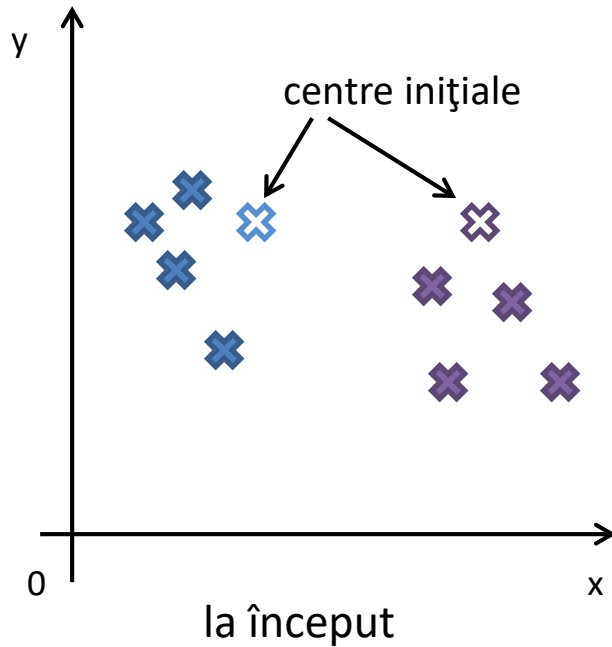


# Algoritmul k-means

- Algoritmul parcurge următorii pași:
  - ❖ Definirea bazei de date: un număr de elemente pentru care nu se cunoaște clasa de apartenență
  - ❖ Definirea unui număr fix de  $k$  centre, dintre punctele existente în baza de date
  - ❖ Repetă până la îndeplinirea criteriului de oprire
    - Calculul distanței între fiecare centru și toate punctele din baza de date
    - Alocarea pentru centrul  $k$  a punctelor celor mai apropiate ca distanță, până la un prag limită
    - Recalcularea fiecărui centru ca medie a elementelor asociate centrului respectiv în iterația curentă
    - Verifică condiția de oprire: nici un punct nu a schimbat centrul căruia i-a fost alocat.

# Algoritmul k-means

- Interpretare geometrică



- Centrul final nu coincide neapărat cu unul dintre punctele inițiale.

# Algoritmul k-means - avantaje și dezavantaje

- Avantaje:
  - ❖ Este ușor de implementat software, necesită cod puțin și simplu.
  - ❖ Permite definirea unor “reprezentanți-tip” pentru clasele existente în baza de date (media vectorilor asociați centrului respectiv).
- Dezavantaje:
  - ❖ Rezultatele depind de alegerea corectă a numărului și poziției inițiale a centrelor  $k$ .
  - ❖ Necesită definirea unui număr fix de clase, egal cu numărul de clase cunoscute în setul inițial. Nu poate descoperi clase noi.
  - ❖ Pentru a obține rezultate optime, datele din baza de date trebuie să formeze clase bine definite și separabile.

**va urma...**

Algoritme evolutive.