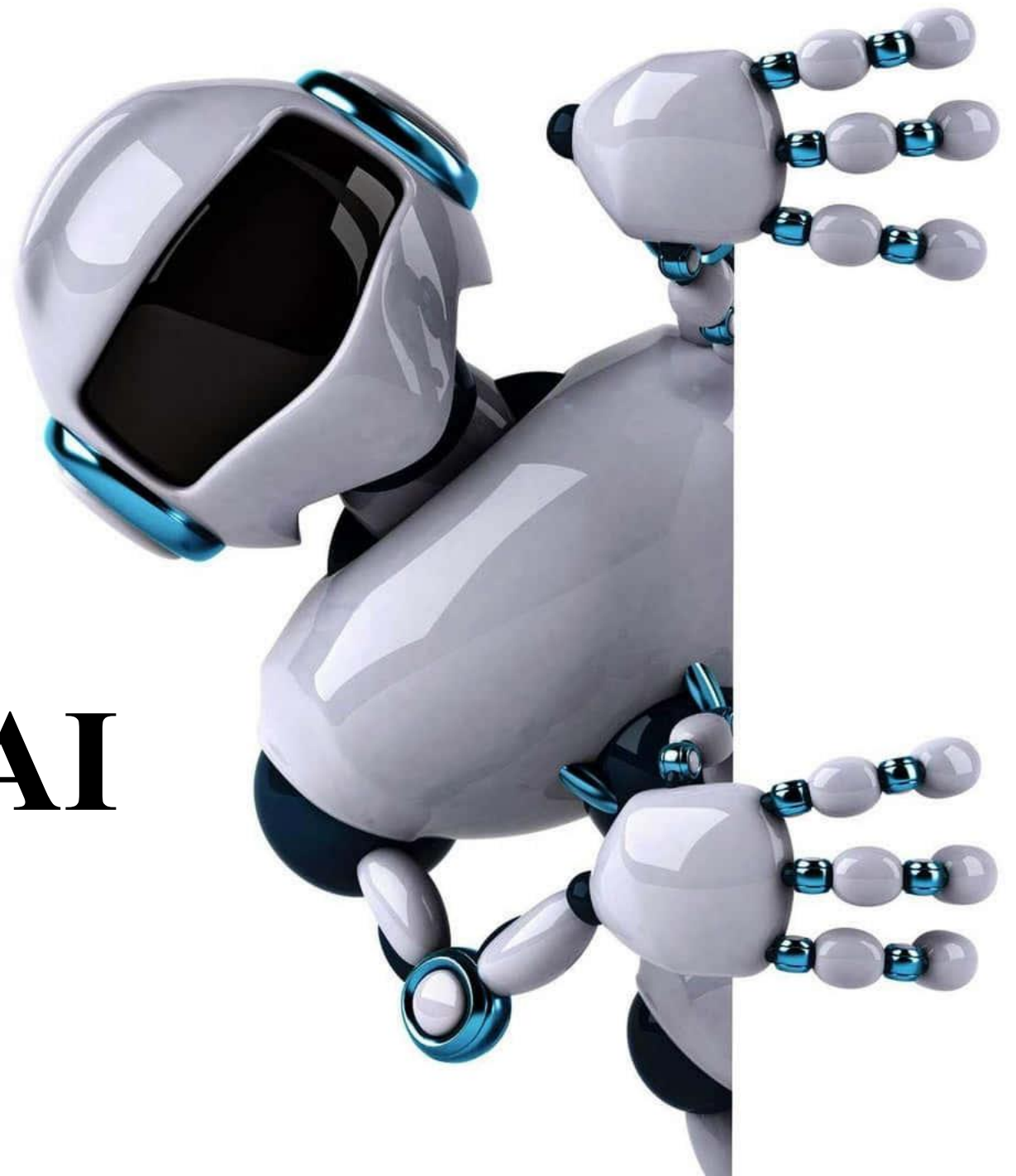
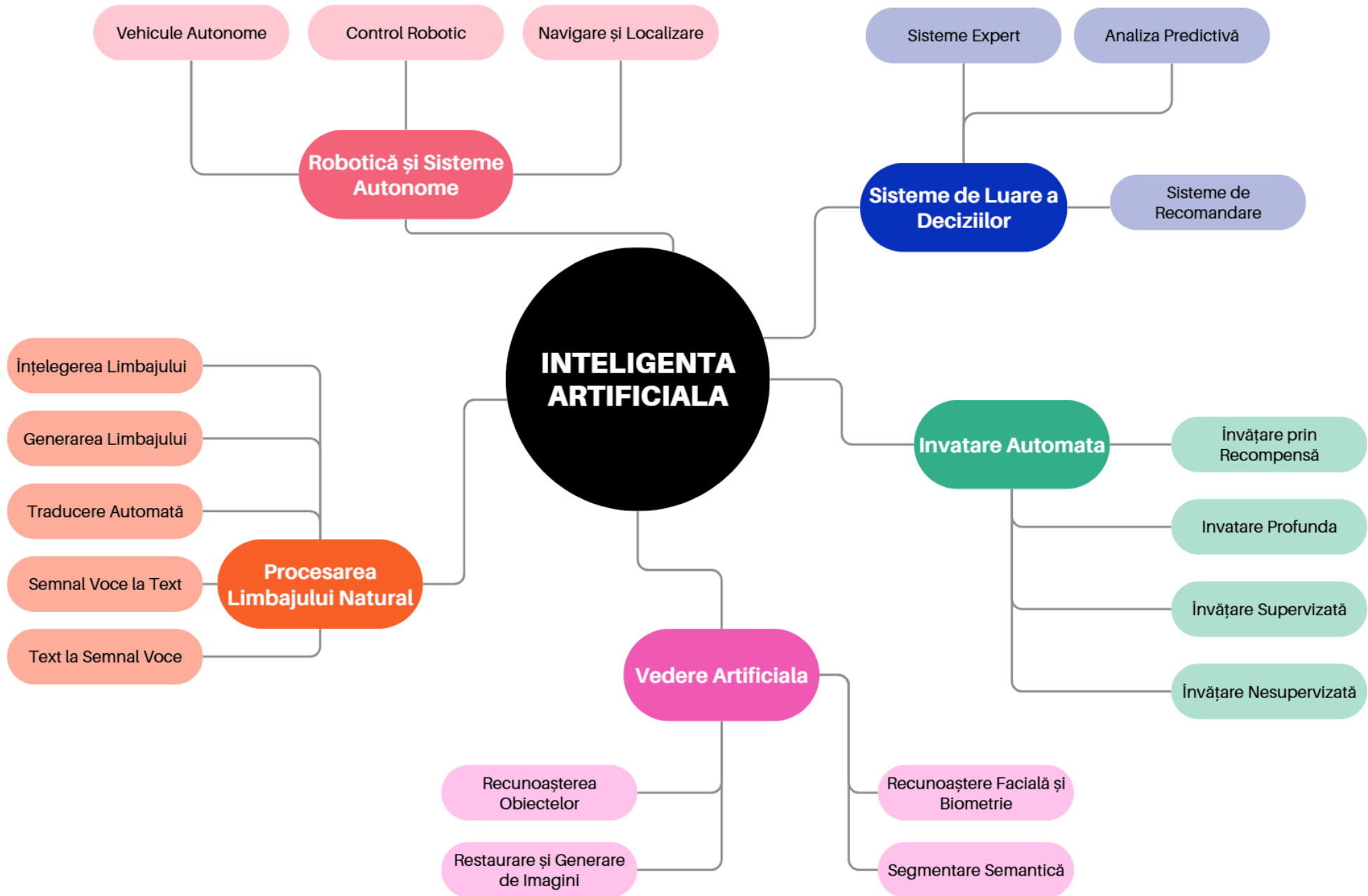
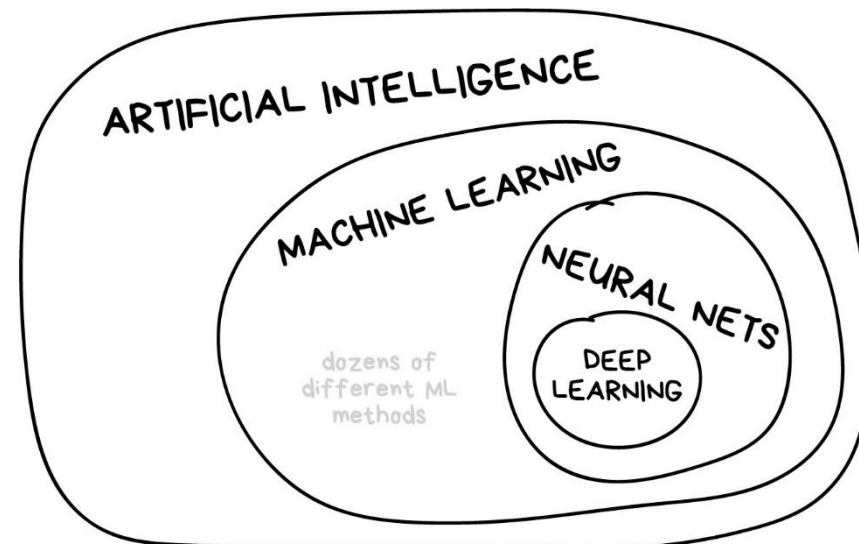


CURS **Introducere în AI**





- **Primele metode de învățare automată (Machine Learning -ML) → bazate pe statistică (anii '50).**
- **ML clasic** – pentru **date simple**, cu caracteristici clare.
- **Rețele neuronale & Deep Learning** – pentru **date complexe și nestructurate** (imagini, text, video).
- **Învățare prin recompensă (Reinforcement Learning)** – când nu avem date istorice, dar avem un **mediu de explorat**. Se folosește când un agent învață prin încercare într-un mediu. Agentul primește recompense sau penalizări.
- **Regula de baza: $DL \subset ML \subset AI$.**



Classical Machine Learning

Task Driven

Data Driven

Supervised Learning

(Pre Categorized Data)

Unsupervised Learning

(Unlabelled Data)

Classification

(Divide the socks by Color)

Eg. Identity
Fraud Detection

Regression

(Divide the Ties by Length)

Eg. Market
Forecasting

Clustering

(Divide by Similarity)

Eg. Targeted
Marketing

Association

(Identify Sequences)

Eg. Customer
Recommendation

Dimensionality Reduction

(Wider Dependencies)

Eg. Big Data
Visualization

Obj: Predications & Predictive Models

Pattern/ Structure Recognition

De ce avem nevoie de Machine Learning?

Volumul mare de date face imposibilă scrierea manuală a regulilor.

Problemele reale sunt complexe și **greu de modelat explicit.**

Permite adaptarea automată la date noi.

Oferă performanțe **mai bune** decât metodele bazate pe reguli fixe.

Automatizează sarcini **dificile** (clasificare, predicții, recunoaștere).

Descoperă tipare **ascunse** în date care nu sunt evidente pentru oameni.

Nu există soluție universală → **mai multe variante posibile.**

- ❑ **Machine Learning** = Un sistem care nu este programat explicit cu reguli, ci învață automat din date cum să ia decizii sau să facă predicții (ex. Identificarea automată a leziunilor în imagini dermatologice).
- ❑ **Date** = Informații brute colectate din lumea reală (ex. ECG).
- ❑ **Date etichetate (Ground Truth / Labeled Data)** = Date pentru care știm răspunsul corect, de obicei stabilit de specialiști (ex. diagnostice confirmate clinic).
- ❑ **Caracteristici (Features)** = Aspectele relevante extrase din date, pe baza cărora se iau decizii (ex. dimensiunea unei tumori dintr-o imagine).
- ❑ **Etichetă (Label)** = Rezultatul corect asociat unui exemplu (ex. diagnostic, scor de risc)
- ❑ **Clasă (Class)** = Eticheta sau categoria în care încadrăm un exemplu (ex. „boală prezentă” vs „absentă”)
- ❑ **Model** = „Regula” sau funcția pe care o învață sistemul pentru a face predicții (ex. un model care primește o radiografie și spune dacă există pneumonie)
- ❑ **Set de antrenare / testare**
 - Antrenare / Train: date pe care învață modelul
 - Testare / Test: date noi, folosite pentru **evaluare**Exemplu: antrenezi pe pacienți existenți, testezi pe pacienți noi.
- ❑ **Set de validare (validation Set)** = Folosit pentru **reglarea hiperparametrilor** modelului (între train și test).

- ❑ **Antrenare (Training)** = Procesul prin care **modelul învață din exemple** (ex. sistemul analizează mii de radiografii etichetate pentru a învăța tipare)
- ❑ **Funcție de pierdere (Loss Function)** = Măsoară **cât de greșite sunt predicțiile** modelului (ex. cât de departe este predicția riscului de infarct față de realitate).
- ❑ **Optimizare (Optimization)** = Procesul de ajustare a modelului pentru a **minimiza eroarea**.
Practic: „cum învață modelul din greșeli”.
- ❑ **Predicție** = Ieșirea modelului pentru o intrare data (ex. „probabilitate de cancer: 0.82”)
- ❑ **Inferență** = Aplicarea modelului **pe date noi**, nevăzute anterior (ex. analiza unei radiografii noi a unui pacient)
- ❑ **Forme de predicție**
 - Clasificare → categorii (boală / nu)
 - Regresie → valori numerice (nivel risc, timp de supraviețuire)
- ❑ **Generalizare** = Capacitatea modelului de a funcționa bine **pe date nevăzute**.

Cum știe modelul dacă greșește?

Eroarea (loss) = diferența dintre predicție și valoarea reală

Cum măsurăm eroarea:

- **MAE (Mean Absolute Error)** → cât greșim în medie
- **MSE (Mean Squared Error)** → penalizează mai mult greșelile mari

Exemple:

- Predicție: 100 | Real: 102 → eroare mică
- Predicție: 100 | Real: 180 → eroare mare

Cum învață modelul:

- face o predicție
- calculează eroarea
- se ajustează **parametrii**
- repetă

Când un model de AI este cu adevărat bun?

Probleme posibile:

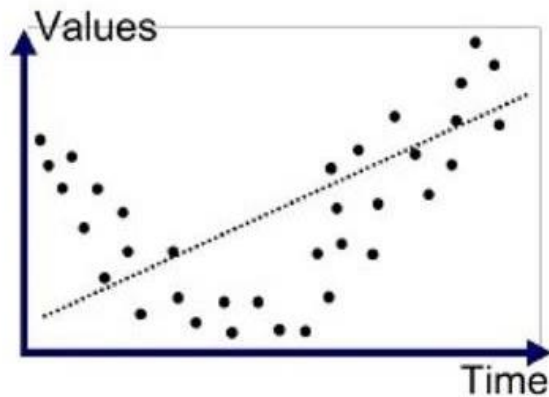
Underfitting (prea simplu)

→ nu învață relația din date

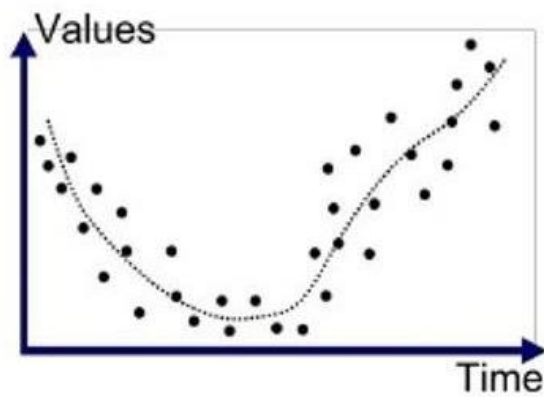
Overfitting (prea complex)

→ memorează datele

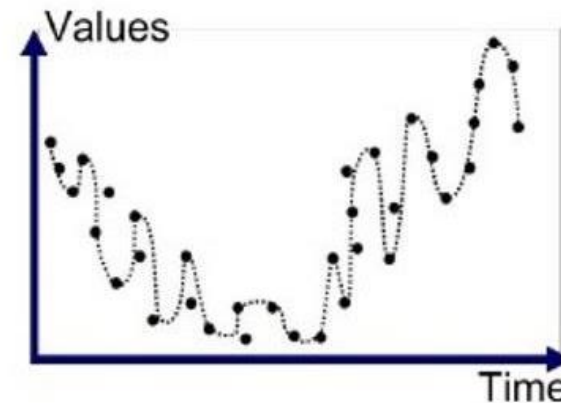
→ nu funcționează pe date noi



Underfitted



Good Fit/Robust



Overfitted

ÎNVĂȚAREA NESUPERVIZATĂ

Invățarea nesupervizată

clustering = **partiționarea unui set de date neetichetate în grupuri (clustere)** astfel încât obiectele din același grup să fie **cât mai similare între ele**, iar obiectele din grupuri diferite să fie **cât mai diferite**.

Nu se pornește de la etichete sau informații a priori. Totul depinde de modul în care **definim similaritatea** (distanță euclidiană, distanța cosinus etc.) dintre date.

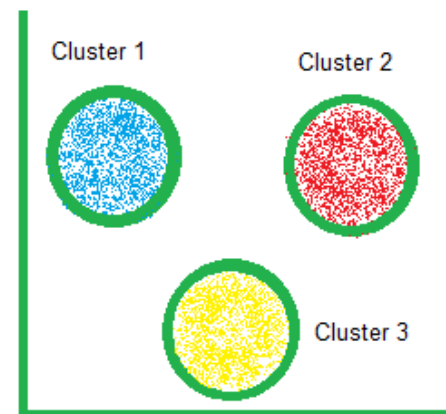
Util pentru situații unde **nu știm dinainte câte grupuri există**.

Algoritmi populari: **K-Means, DBSCAN, Mean-Shift**.

Exemple: segmentarea clienților (loialitate, comportamente), organizarea automată a fotografiilor după fețe, detectarea comportamentelor anormale.



Red, Blue and Yellow data points before Clustering



Red, Blue and Yellow data points after Clustering

Tehnici de clasificare nesupervizată

Tipuri de similaritate în clustering

1. Similaritate structurala = asemănarea dintre obiecte din punct de vedere al structurii sau formei lor.

Exemplu:

- documente comparate după structura textului (lungime propoziții, organizare în paragrafe)
- imagini sau video comparate după aranjare spațială a pixelilor sau cadrelor

2. Similaritate semantică = asemănarea în sens sau semnificație, nu neapărat în formă.

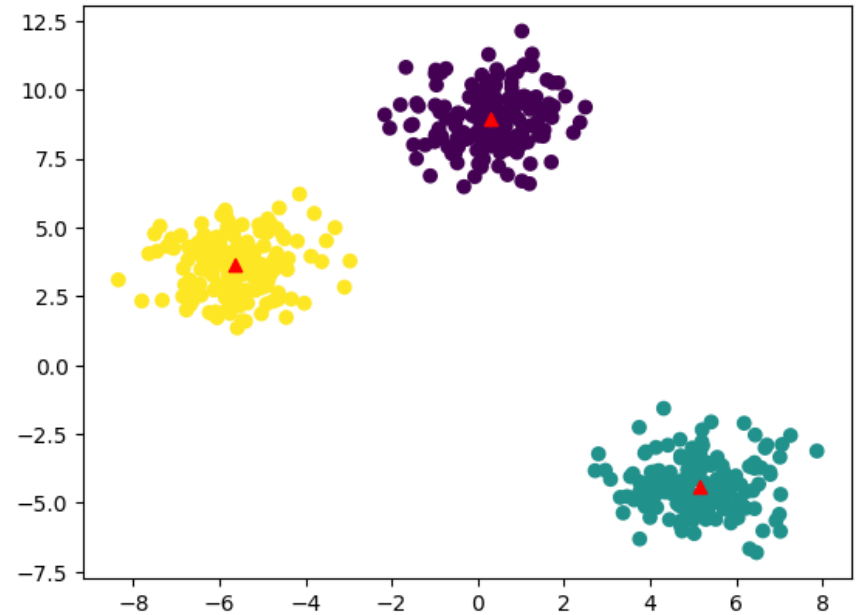
Exemplu:

- două texte care vorbesc despre același subiect, chiar dacă folosesc cuvinte diferite
- „mașină” și „automobil” sunt semantic similar

Se folosește frecvent în: NLP, motoare de căutare, sisteme de recomandare

K-Means

- Alegem mai întâi **numărul de clustere** K (câte grupuri vrem să obținem).
- Apoi alegem K **puncte inițiale ca centre** (centroizi), de obicei aleator.
- **Fiecare punct din date este atribuit celui mai apropiat centru**, adică merge în clusterul al cărui centru este cel mai „aproape” de el.
- După ce toate punctele au fost împărțite, pentru fiecare cluster **se recalculează centrul** astfel: → se face „media” tuturor punctelor din acel grup (centrul se mută în mijlocul lor).
- Din nou, fiecare punct este reassignat la cel mai apropiat nou centru.
- Pașii se repetă până când centrele nu se mai mișcă semnificativ și grupurile rămân stabile.



Limitările învățării nesupervizate

- ❑ **Interpretarea rezultatelor este mai dificilă** și mai puțin intuitivă.
- ❑ **Rezultatele sunt orientative, nu există un „răspuns corect” cu care sa verificam datele.**
- ❑ Există riscul de a identifica modele irelevante sau simple fluctuații aleatorii din date.
- ❑ Este utilizat în principal pentru explorarea datelor, analiza inițială și organizarea informațiilor.
- ❑ **Precizia este mai greu de evaluat** comparativ cu învățarea supervizată.
- ❑ De multe ori este **necesară intervenția unui expert** pentru a decide dacă rezultatele sunt relevante.

ÎNVĂȚAREA SUPERVIZATĂ

Invățarea supervizată

Două tipuri majore de probleme:

- **Clasificare** – predicția unei **categorii discrete**.
- **Regresie** – predicția unei **valori numerice continue**.

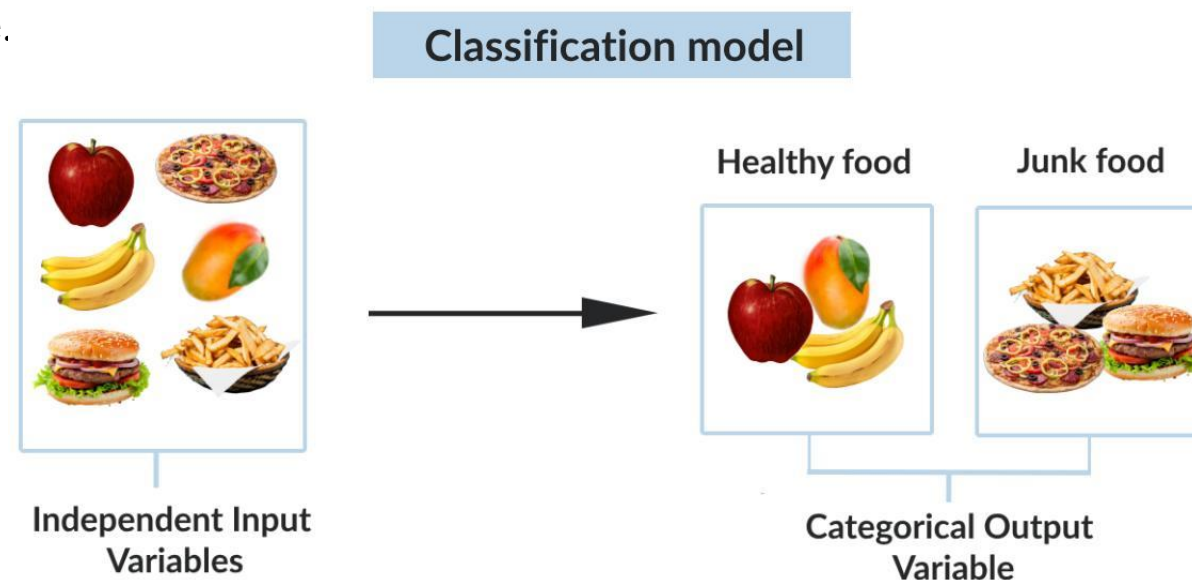
Se bazează pe tipare descoperite din date etichetate.

Alegerea algoritmului depinde de:

- **tipul și dimensiunea datelor**
- **precizia și viteza dorite**
- **nevoia de interpretabilitate**

Exemple de clasificare:

1. email → spam / non-spam
2. client → risc scăzut / risc ridicat
3. imagine medicală → normală / suspect



Regresia

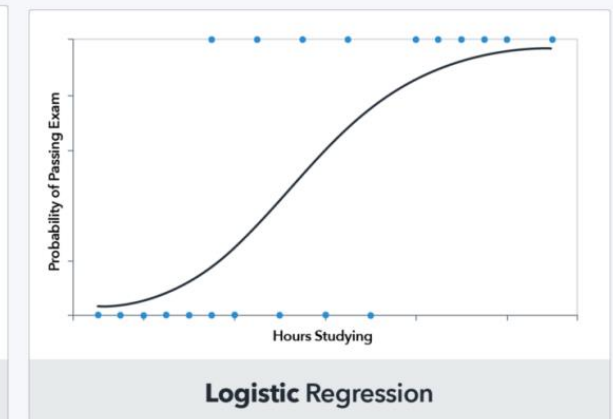
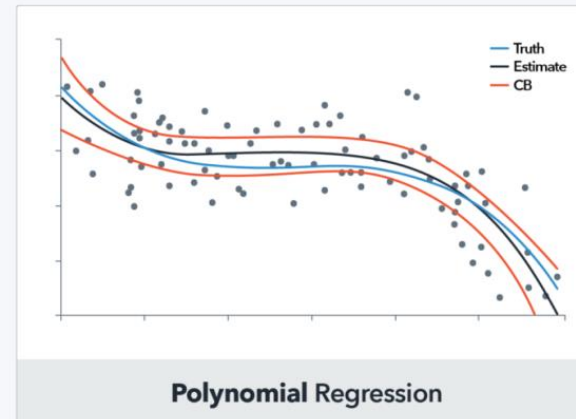
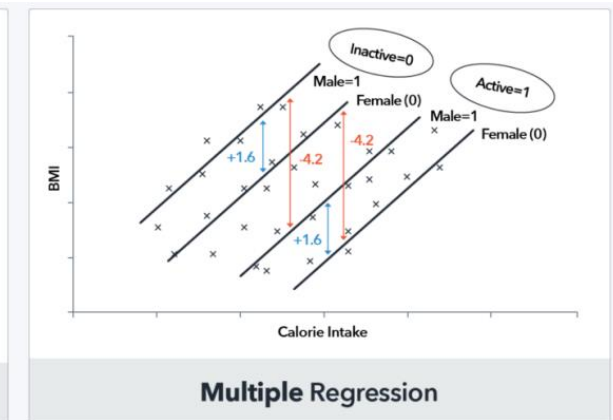
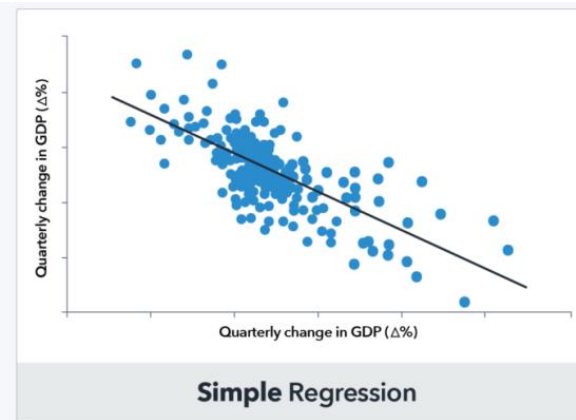
Regresia = predicția unei valori numerice continue

- Spre deosebire de clasificare → prezice **valori**, nu categorii
- Modelul învață **relații matematice** între variabile
- Folosește date istorice pentru a face **estimări viitoare**

Exemple:

- estimarea prețurilor
- calculul timpilor de livrare
- predicția indicatorilor medicali
- estimarea consumului de resurse

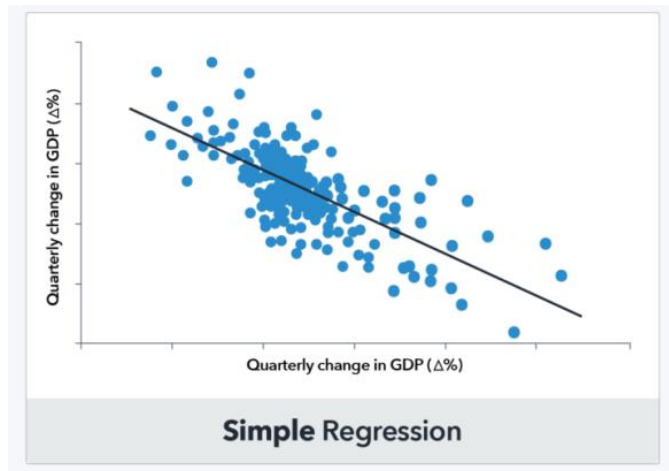
Regresia răspunde la întrebarea: „*Cât?*”



Modele liniare

Regresia Liniară

- prezice valori numerice
- model: $y = w_1x_1 + w_2x_2 + \dots + b$
- fiecare feature x_n (date de intrare) are o pondere w_n
- ajustăm ponderile pentru a minimiza eroarea
- **ponderea arată cât de mult contează acel feature în predicție**



Regresie Logistică

- folosită pentru **clasificare binară** (ex: 0 / 1)
- calculează: $z = w \cdot x + b$
- probabilitatea de apartenență la o clasă = $\text{sigmoid}(z)$
- dacă probabilitatea $> 0.5 \rightarrow$ clasa 1



Modele neliniare

k-Nearest Neighbors (k-NN)

Pentru un punct nou:

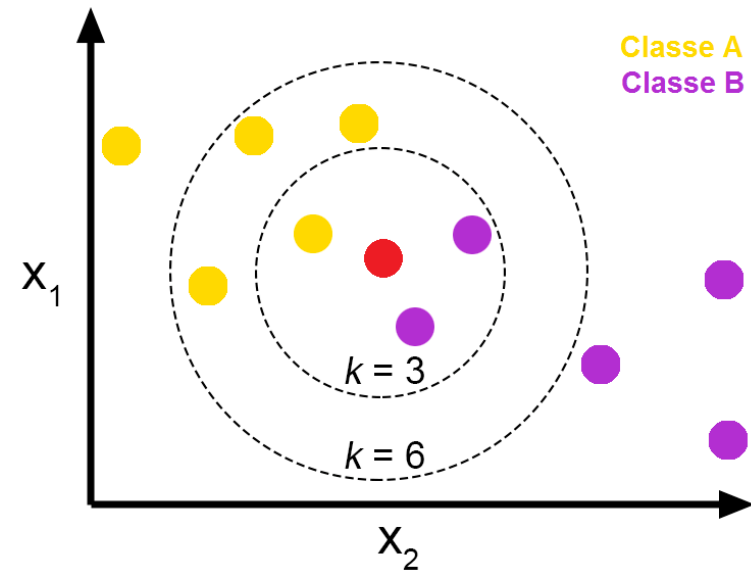
- calculăm distanța față de **toate punctele din train set**
- alegem **cei mai apropiați k vecini** (ex: $k = 3, 5$)

clasificare → vot majoritar

regresie → media celor k vecini

De obicei alegem k impar ca să evităm egalitățile

Exemplu: recomandarea tratamentelor pe baza pacienților similari



Modele neliniare

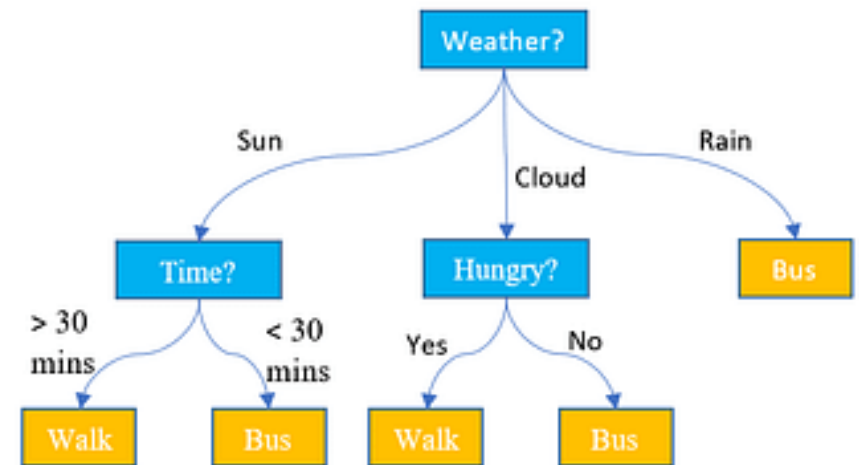
Decision Trees (Arbori de decizie)

- Pentru fiecare feature, algoritmul sortează valorile și **testează mai multe praguri posibile**, împărțind datele în două grupuri (\leq prag și $>$ prag).
- Pentru fiecare împărțire **calculează calitatea separării** și o alege pe cea care produce grupurile cele mai omogene.
- Procesul se **repetă recursiv pe fiecare nod nou**, până când se îndeplinesc condițiile de oprire (ex: adâncime maximă).

Frunzele sunt noduri finale care nu mai fac împărțiri și **conțin eticheta de clasă**:

- clasificare → clasa finală (majoritară)
- regresie → fiecare frunză conține mai multe exemple din setul de antrenare, iar predicția este **media valorilor** lor reale

Exemplu: diagnosticarea unei boli pe baza simptomelor și analizelor.



Modele neliniare

Random Forest (Păduri aleatoare)

Construiește mai mulți **arbori de decizie independenți**.

Fiecare arbore:

- primește un subset aleator de date
- folosește un subset aleator de features
- construiește un arbore complet pe acele date

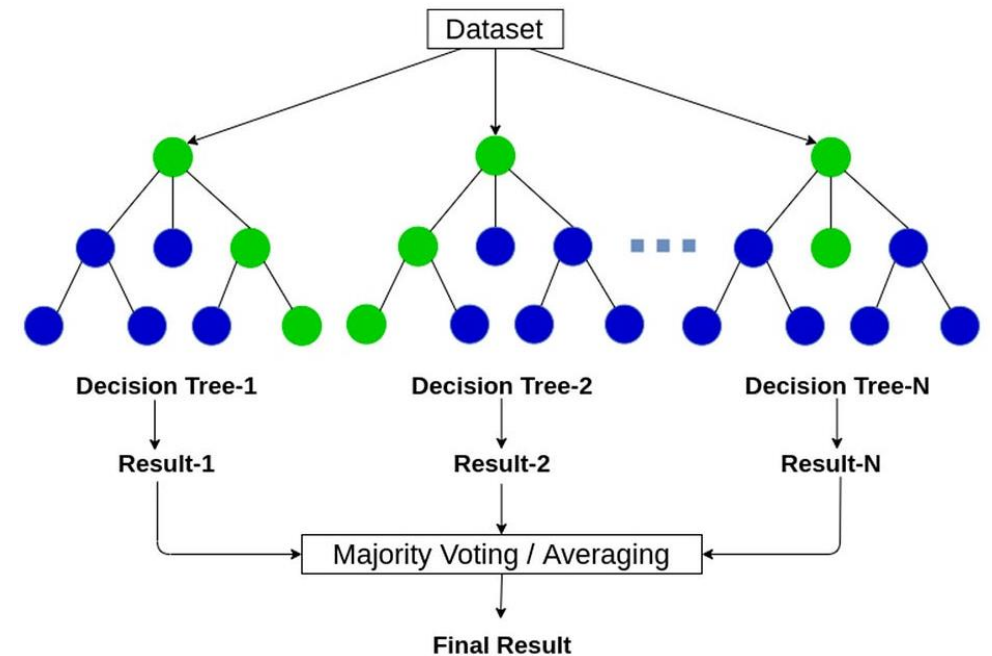
Rezultat:

- clasificare → vot majoritar
- regresie → se face **media predicțiilor tuturor arborilor**

Exemplu: detectarea diabetului pe baza mai multor variabile medicale, nu doar glicemie

Decision Tree = un singur „doctor”

Random Forest = mai mulți doctori care votează



Support Vector Machine (SVM)

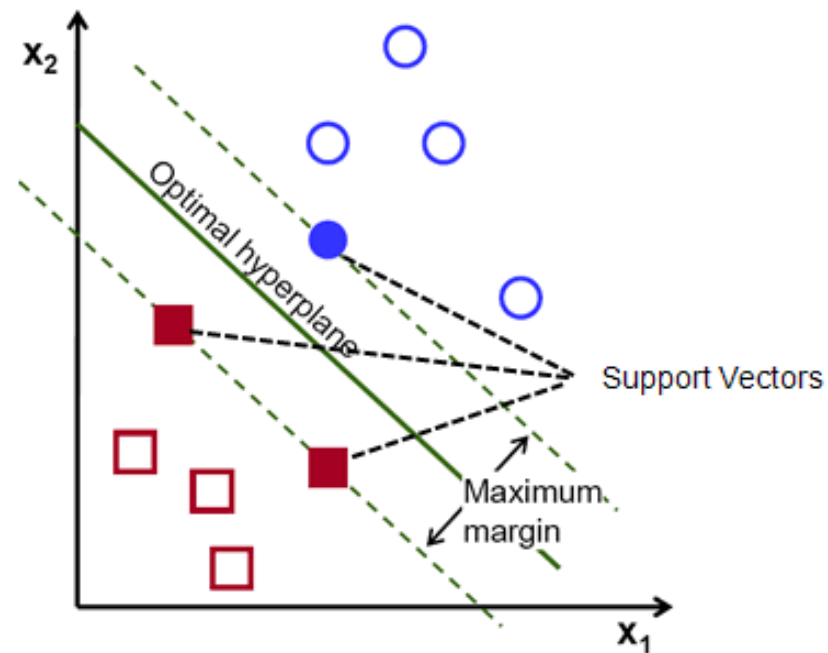
SVM liniar

- separă datele cu un **hiperplan** (linie în 2D, plan în 3D)
- caută separarea cu **margină maximă între clase**
- funcționează bine când datele sunt aproximativ separabile liniar
- separarea este determinată de câteva puncte apropiate de frontieră numite **support vectors (vectori de suport)**

SVM neliniar

- transformă datele într-un spațiu unde devin separabile liniar
- în spațiul original, separarea este o **frontieră curbată / neliniară**
- separarea este determinată de **support vectors**

Exemplu: diagnosticarea hipertensiunii arteriale

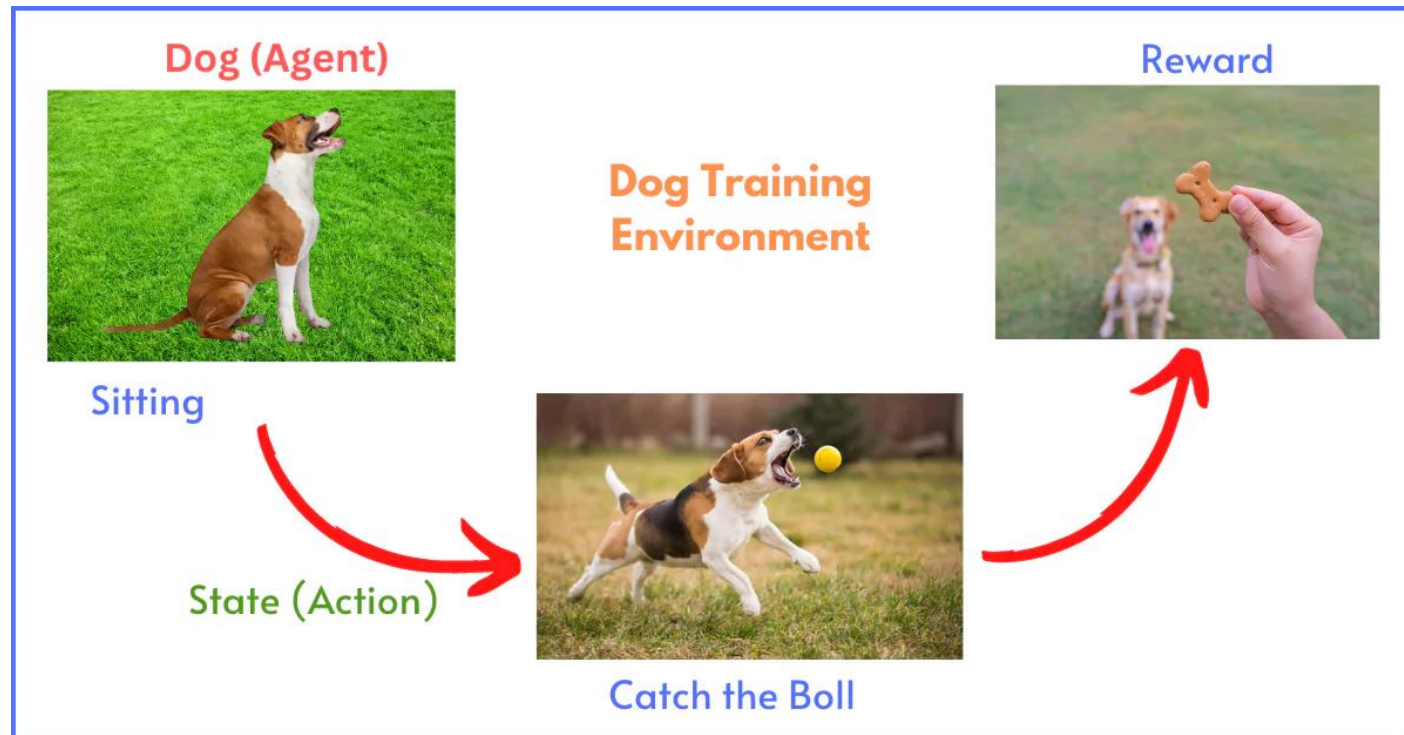


REINFORCEMENT LEARNING

În Reinforcement Learning (RL) există câteva componente esențiale care formează mecanismul de învățare:

- **Agentul** → entitatea care ia decizii (ex.: robot, mașină autonomă, software).
- **Mediul** → contextul în care acționează agentul (ex.: oraș, joc video, piață financiară).
- **Acțiuni** → alegerile posibile la fiecare pas (ex.: accelerează, virează, cumpără/vinde).
- **Recompense** → feedback-ul primit după fiecare acțiune (ex.: puncte pozitive pentru succes, penalizări pentru greșeli).

Obiectivul final → agentul caută să maximizeze recompensa totală în timp, nu doar câștigurile imediate.



- **RL = învățare prin încercare și eroare, cu recompense și penalizări.**
- Folosit când există un mediu de interacțiune, nu doar date statice.
- Două mari abordări: **Model-Based** (agentul își construiește o „hartă” a mediului) și **Model-Free** (agentul nu are hartă completă, ci învață direct din experiențele trecute).
- **Algoritmi populari: Q-Learning, SARSA, Deep Q-Networks**

Exemple: jocuri, roboți, mașini autonome, optimizare resurse.

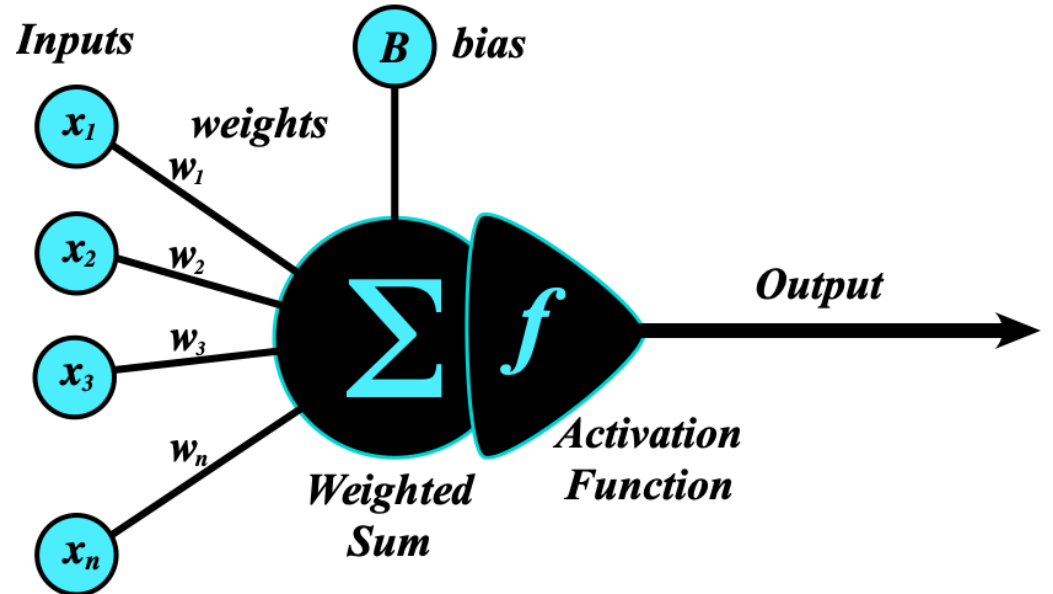
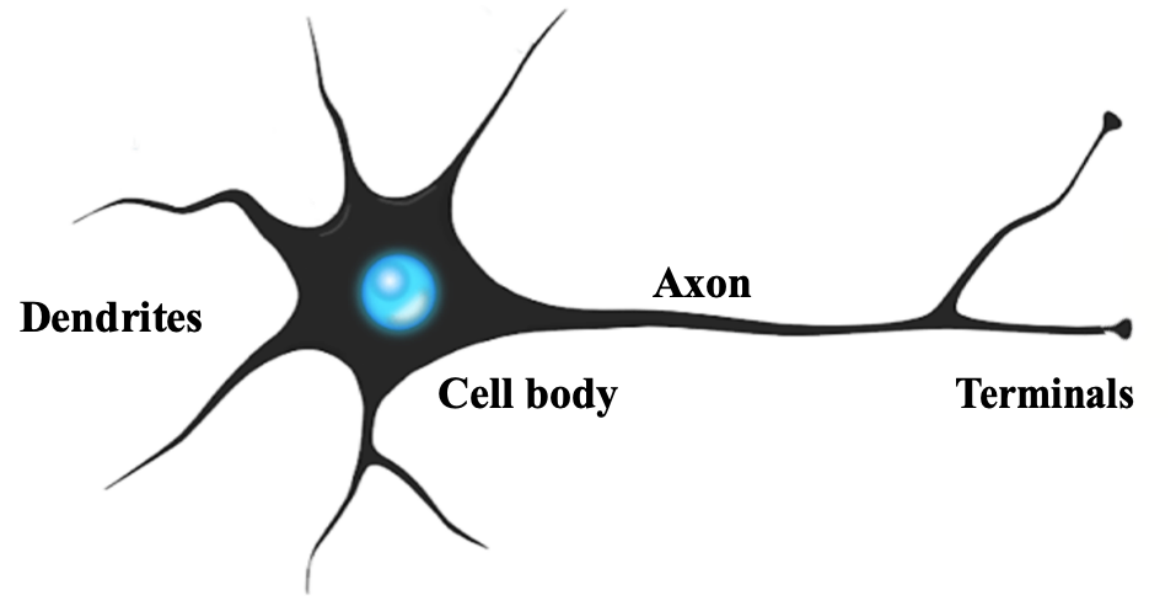
Limitări

- Necesită **foarte multe date și interacțiuni**
- Antrenarea poate fi **lentă și instabilă**
- Definierea corectă a **recompensei** este dificilă
- Poate învăța **strategii greșite** dacă reward-ul e prost ales
- Greu de aplicat în medii reale (costuri, risc)

NEURAL NETWORKS Şİ DEEP LEARNING

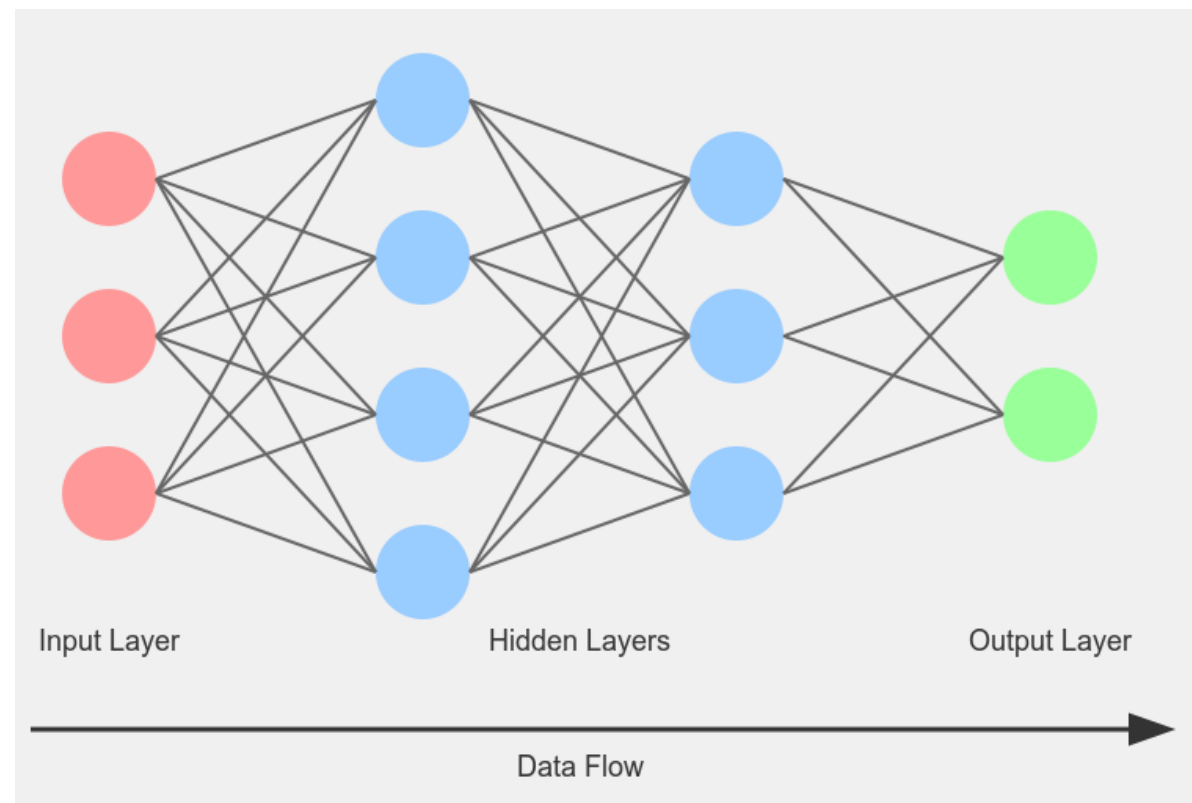
Rețelele neuronale:

- pot reproduce și depăși metode clasice de ML
- au devenit **populare datorită creșterii puterii de calcul (GPU)**
- sunt folosite pentru **imagini, text, video, voce** și multe alte tipuri de date
- inspirate de modul de funcționare al creierului uman
- formate din „neuroni” artificiali conectați prin **ponderi (weights)**
- învață prin ajustarea acestor ponderi pe baza datelor



Cum funcționează o rețea neuronală

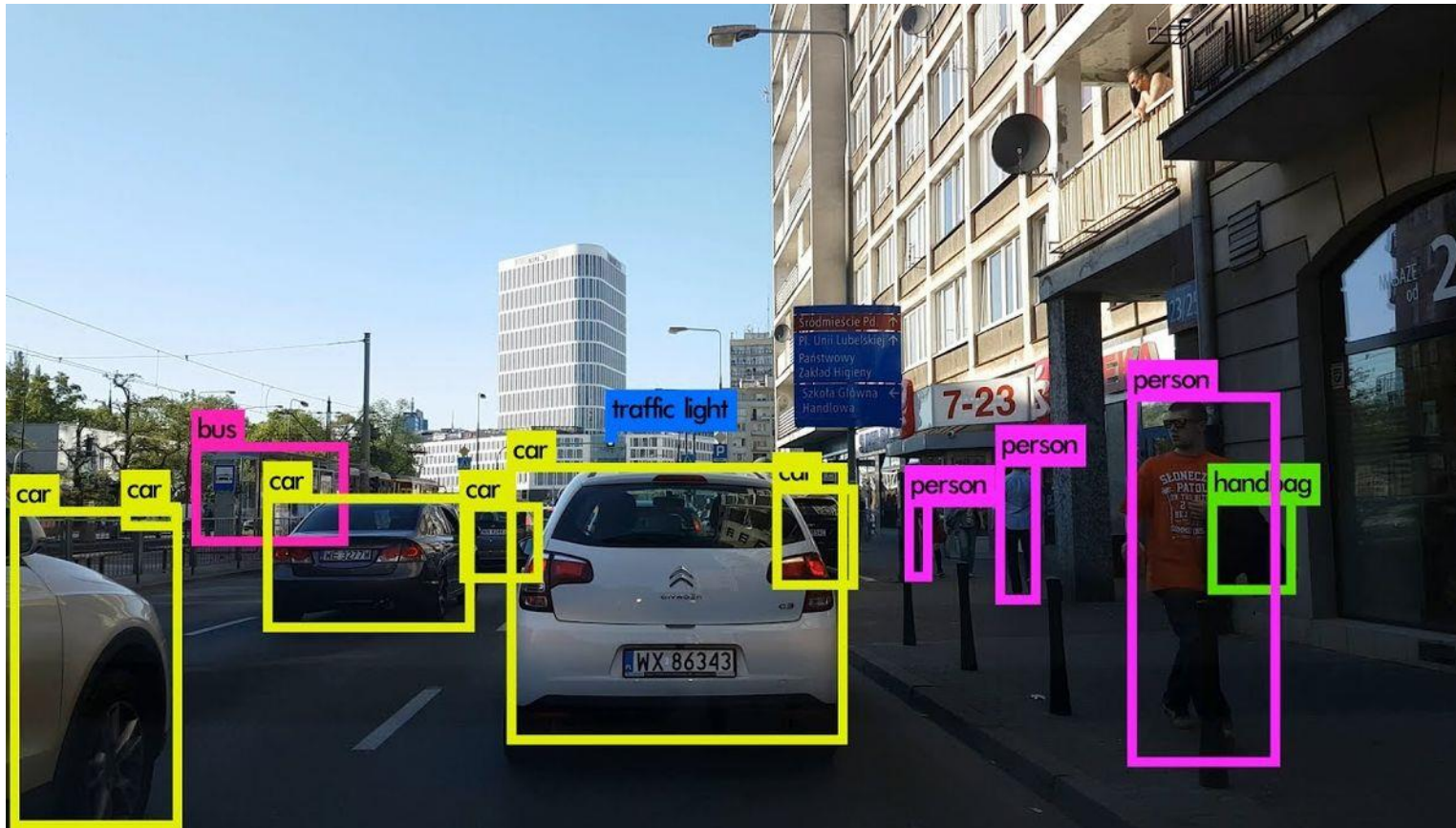
- **Structura rețelei:**
 - input (date brute)
 - straturi ascunse (descoperă tipare)
 - output (rezultat final)
- **Neuronul:** primește intrări → aplică o funcție → generează o ieșire
- **Ponderi (weights):** stabilesc **importanța fiecărei intrări** în calculul final
- **Învățarea** se face prin ajustarea ponderilor pentru a reduce eroarea și a îmbunătăți predicțiile la fiecare pas (**backpropagation**)



Deep Learning

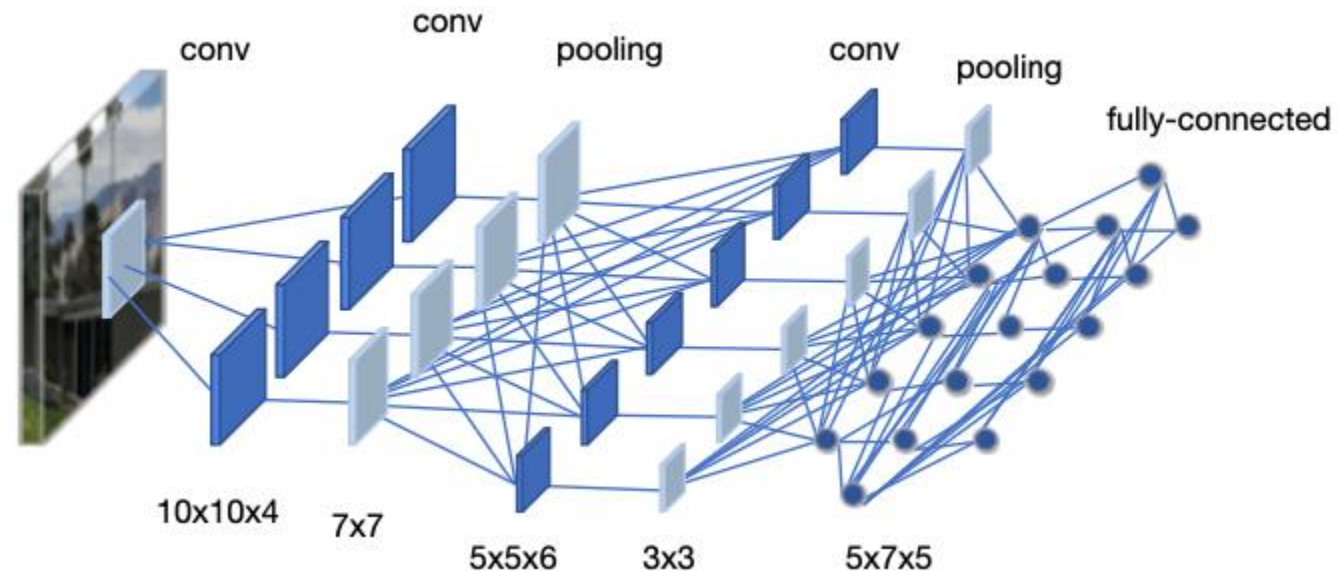
Rețele neuronale clasice: doar câteva straturi → putere limitată de analiza → pot rezolva probleme simple.

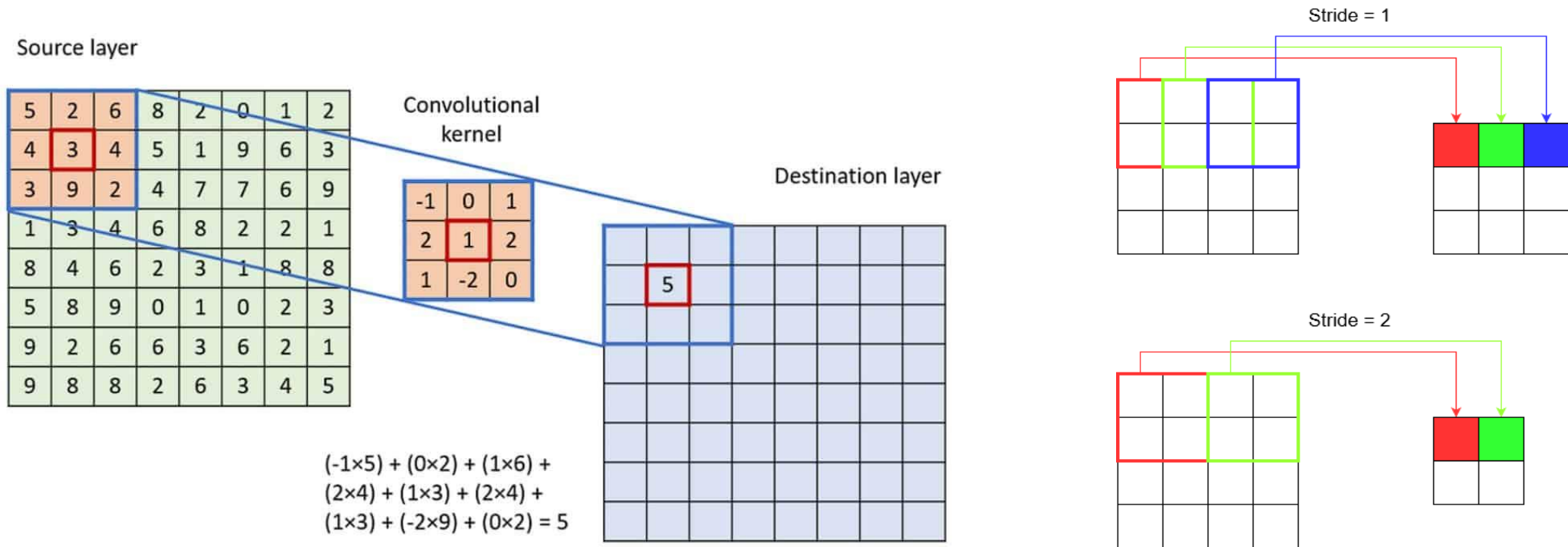
Deep Learning: rețele cu multe straturi, capabile să învețe tipare complexe.



CNN (Convolutional Neural Networks)

- Folosite în special pentru **imagini**
- Aplică **filtre (convoluții)** pentru a **extrage caracteristici locale**
- Fiecare strat învață tipare din ce în ce mai complexe (muchii → forme → obiecte)
- **Pooling reduce dimensiunea** și păstrează informația importantă
- Exemple de modele CNN cunoscute: ResNet, VGG16, VGG19, AlexNet





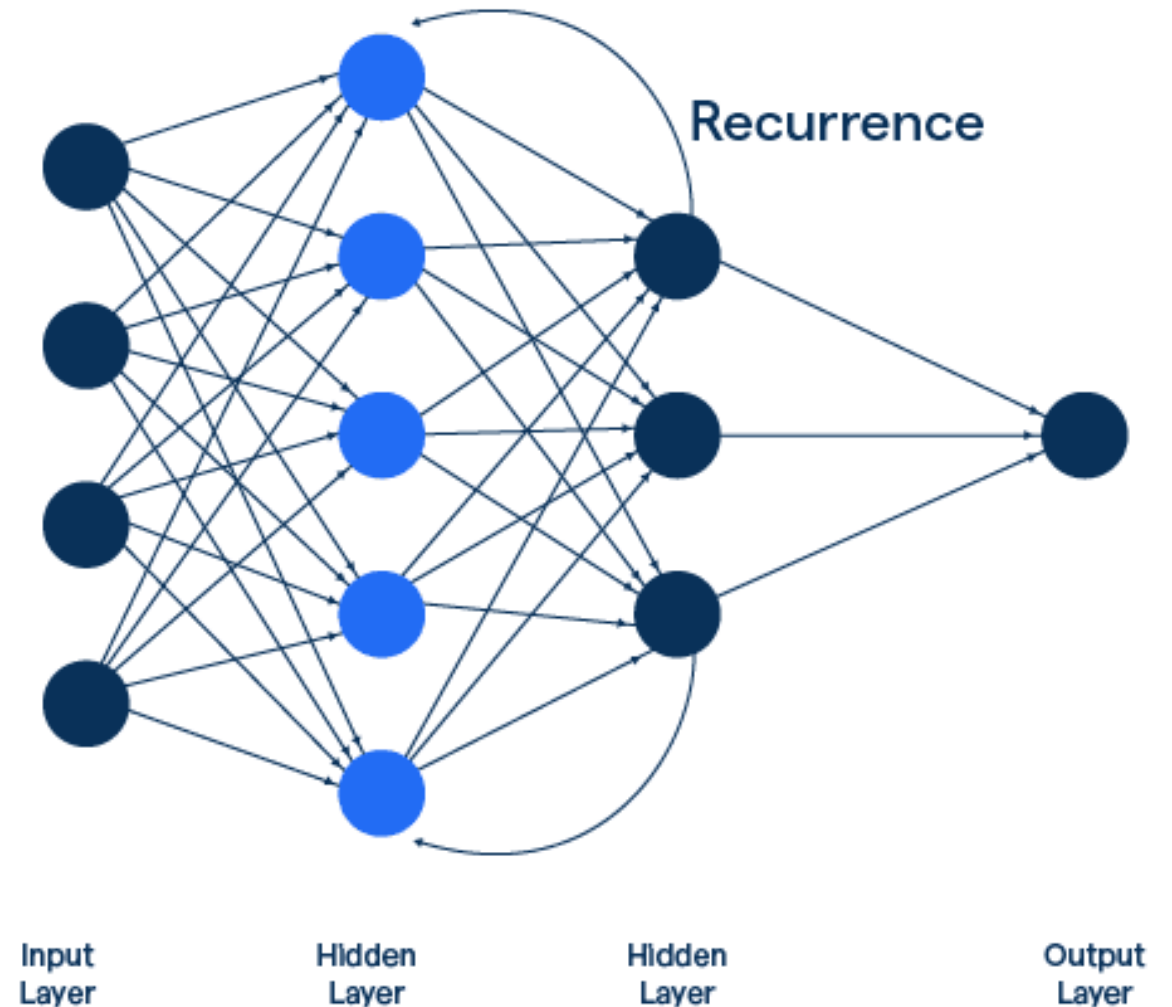
- **Stride:** distanța (în pixeli) cu care se deplasează filtrul pe imagine la fiecare pas, **atât pe orizontală cât și pe verticală**

Exemplu: stride = 2 înseamnă că filtrul sare câte 2 pixeli la dreapta și 2 pixeli în jos când schimbă poziția.

- **Padding:** adăugarea unei margini (de obicei cu zerouri) în jurul imaginii înainte de convoluție (controlează modul în care sunt tratate marginile imaginii).

RNN (Recurrent Neural Networks)

- Proiectate pentru **date secvențiale**
- Au conexiuni recurente → păstrează o **stare internă (memorie)**
- Ieșirea depinde atât de inputul curent, cât și de cele anterioare
- Probleme: **dificultăți în memorarea pe termen lung**
- Soluții: variante îmbunătățite (LSTM, GRU)
- Utilizare: procesare text, traducere, serii temporale (ex. detectarea aritmiilor din ECG)



Transformers

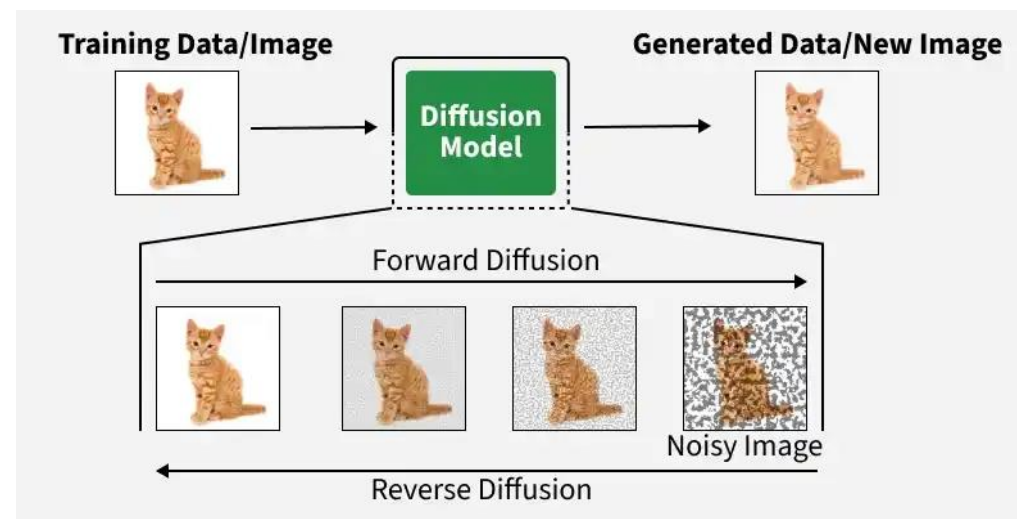
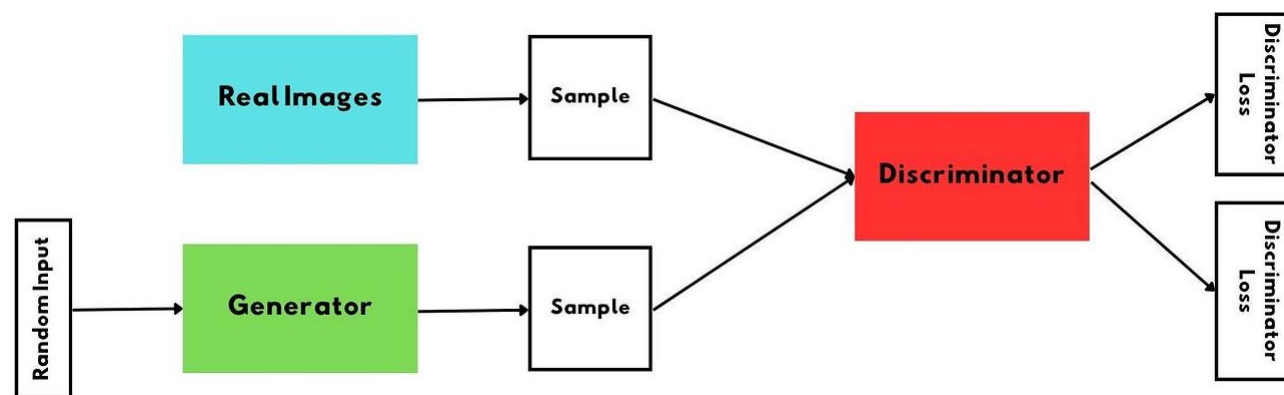
- Au înlocuit multe modele RNN în NLP
- Model de rețele neuronale folosit în special pentru text, dar și imagini, audio și video
- Bazat pe mecanismul de self-attention → „se concentrează” pe părțile importante din date
- Nu procesează secvențial (ca RNN), ci **analizează toate elementele în paralel**
- Foarte eficient pentru **înțelegerea contextului pe distanțe mari**



Modele generative: GAN și Diffusion Models

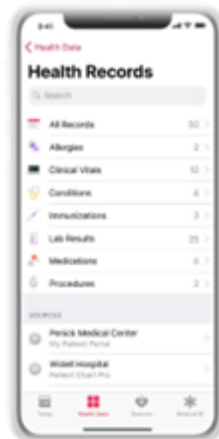
- **GAN (2014):** două rețele „în competiție”
 - Generator (creează) vs. Discriminator (evaluează)
 - Rezultat: imagini, video sau sunete realiste
- **Diffusion Models (2020+):**
 - pornesc de la zgomot aleator
 - îl transformă pas cu pas într-o imagine coerentă.

Aplicații: artă, design, gaming, multimedia, medicină (imagini sintetice pentru antrenare)





TIPURI DE DATE ÎN DOMENIUL SĂNĂTĂȚII



Calitatea datelor și impactul asupra modelului

- Calitatea și diversitatea datelor **influențează direct acuratețea modelului**.
- Fiecare **rând** din setul de date reprezintă **un eșantion** (sample / data point), iar fiecare **coloană** reprezintă o **caracteristică** (feature).
- **Caracteristicile** sunt variabilele de intrare care **descriu datele** și sunt folosite de model pentru învățare.
- **Performanța** unui model depinde adesea mai mult de calitatea caracteristicilor decât de algoritmul folosit.
- Alegerea datelor și a caracteristicilor poate introduce **subiectivitate și erori în rezultate**.

PatientID	Gender	Age	Zip code	Test
55998	M	19	15723	Negative
88557	F	35	15674	Positive
55868	F	35	15674	Positive
44551	M	45	15623	Negative
58524	M	45	15623	Negative
25584	F	61	15633	Negative
58744	F	61	15643	Positive
87524	M	19	15762	Positive
87384	M	19	15762	Negative
17583	F	19	15762	Positive

Date clinice structurate (EHR - Electronic Health Records)

Exemple:

- vârstă, sex
- tensiune arterială, puls
- rezultate analize (glicemie, colesterol)
- diagnostice codificate (ex: ICD)
- tratamente prescrise

→ IA le folosește pentru predicții (ex: risc de infarct, evoluția bolii)

Key Companies Working in the EHR Market



And Others

Date medicale nestructurate (text)

Informații scrise liber de medici:

- scrisori medicale
- observații clinice
- rapoarte de externare

→ IA (NLP) extrage informații utile din text

DD FORM 1289
1 NOV 71
DOD PRESCRIPTION

FOR (Full name, address, & phone number) (if under 12, give age)
John R. Doe, HM3, USN

U.S.S. Neverforgotten (DD 178)

MEDICAL FACILITY U.S.S. Neverforgotten (DD 178) DATE 23 Jan 99

B (Superscription) gm or ml

(Inscription)
Tr Belladonna 15 ml
Amphogel good 120 ml

(Subscription)
m & FI solution

(Signa)
Sig: 5ml tid a.c.

MFGR: Wxeth EXP DATE: 12/02

DD FORM 1289
1 NOV 71
DOD PRESCRIPTION

FOR (Full name, address, & phone number) (if under 12, give age)
John R Doe, HM3, USN

U.S.S. Neverforgotten (DD 178)

MEDICAL FACILITY U.S.S. Never forgotten (DD 178) DATE 23 JAN 99

R (Superscription) gm or ml

(Inscription)
Tr Belladonna 15ml
Amphogel good 120 ml

(Subscription)
m & FI solution

(Signa)
Seg: 5ml tid a.c.

Name Armande Cognia
Address Went Rimbo, Makati City
Age 29 Sex M Date 12-03-90

Rx

(Thiniox)
Amoxicillin 500mg Cap#21

Sig: 1 cap 3x a day for seven days

Physician's Sig: Idela Gny
Lic. No. 122457
PTR No 1234567
S2 No.

Name: armande Cognia
Address Went Rimbo , makati City
Age: 29 Sex : M Date: 12-03-90

Rx

(Thiniox)

Amoxicillin 500mg Cap#21

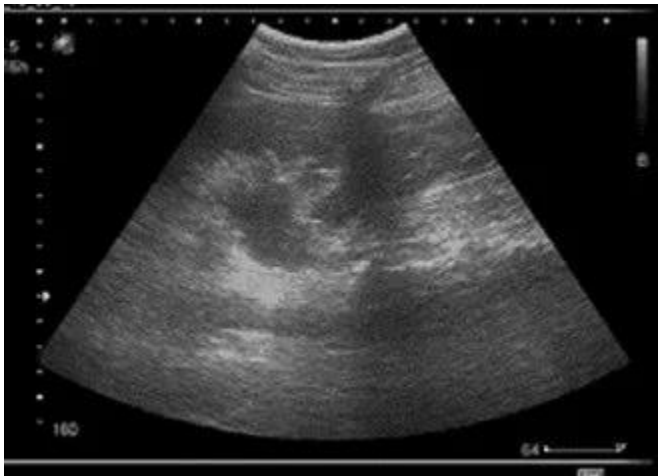
Sig: 1 cap 3X a day for seven days.

Physicians Sig. Idelating
Lic. No. 122457
PTR No 1234567
S2 No.

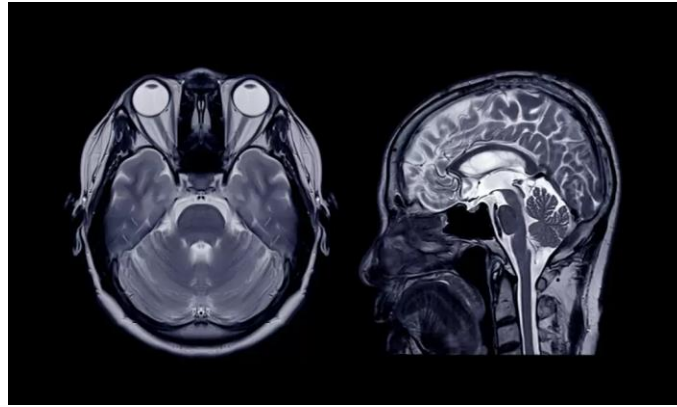
Imagini medicale

IA detectează:

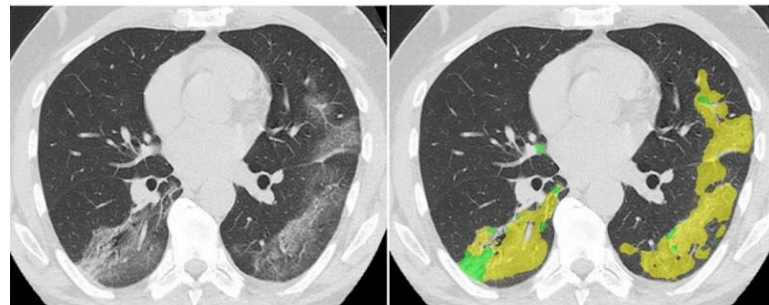
- tumori
- fracturi
- pneumonie



Ecografii



RMN



CT (tomografii)



Radiografii

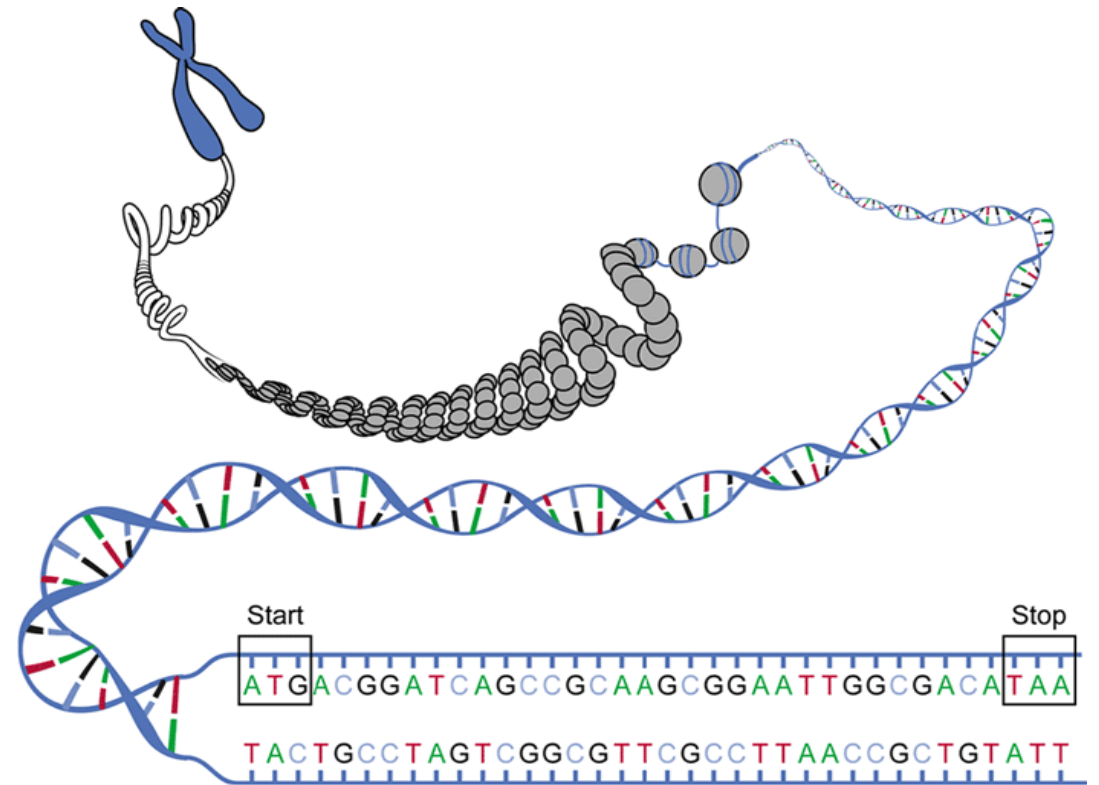
Date genomice

Exemple:

- secvențe ADN
- variații genetice
- mutații genetice (ex: BRCA1)
- date de expresie genică (nivelul de activitate al genelor)

IA contribuie la:

- medicină personalizată (tratament adaptat profilului genetic)
- identificarea riscului de boli (ex: cancer)
- diagnostic mai precis



Date de la dispozitive wearable & senzori

Exemple:

- semnale fiziologice
- activitate fizica
- somn
- nivel oxigen

IA monitorizează sănătatea și detectează anomalii.

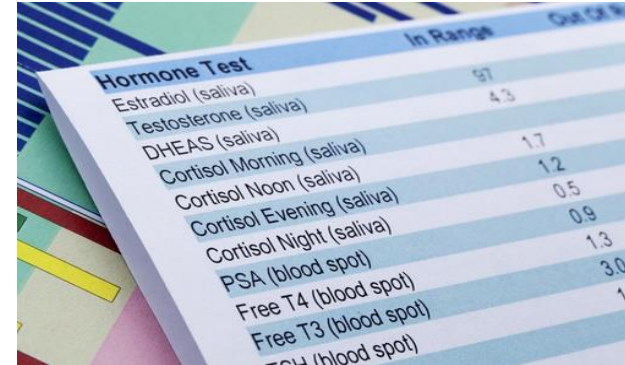


Date de laborator

Exemple:

- hemoleucogramă
- analize biochimice
- teste hormonale

IA detectează anomalii și tipare pentru diagnostic precoce.



Hormone Test	In Range	Out of Range
Estradiol (saliva)	97	
Testosterone (saliva)	4.3	
DHEAS (saliva)		1.7
Cortisol Morning (saliva)	1.2	
Cortisol Noon (saliva)	0.5	
Cortisol Evening (saliva)	0.9	
Cortisol Night (saliva)		1.3
PSA (blood spot)		3.0
Free T4 (blood spot)		1
Free T3 (blood spot)		
TSH (blood spot)		

Date farmaceutice și tratamente

Exemple:

- istoricul medicamentelor
- reacții adverse
- doze

IA optimizează tratamentele și evită interacțiuni periculoase.

Date audio medicale

Exemple:

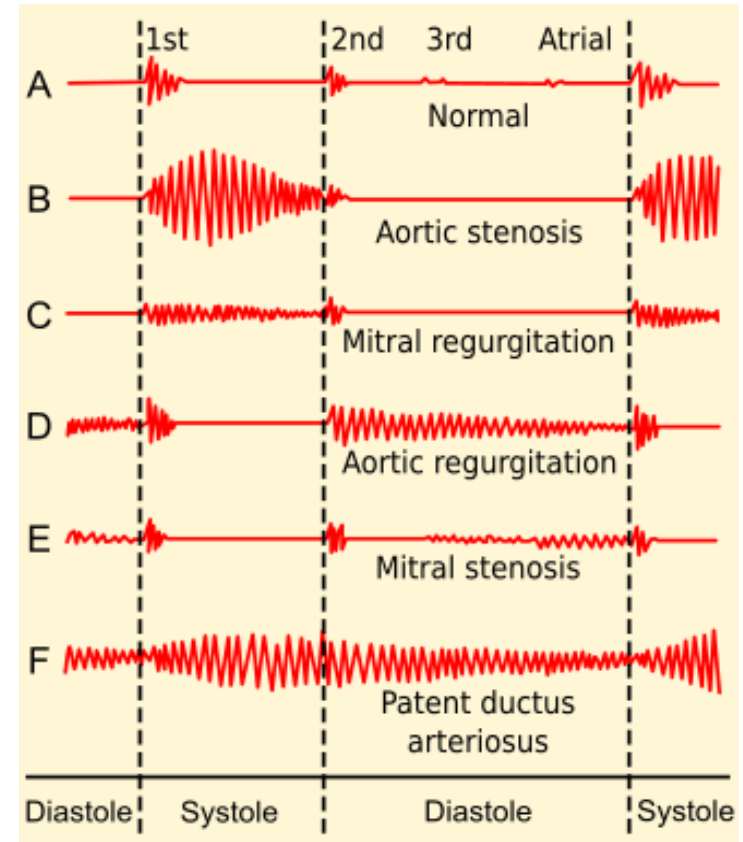
- sunete cardiace
- tuse
- înregistrări audio ale pacienților

IA detectează boli respiratorii sau neurologice

COUGH AI



COVID-19 test
using **cough sounds**

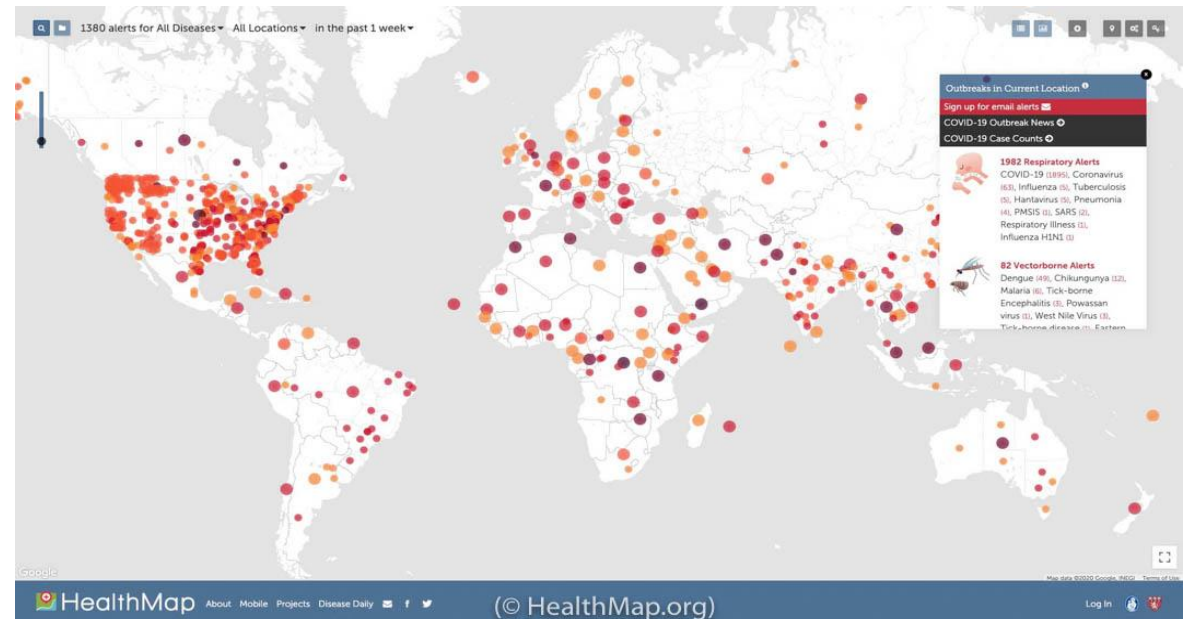


Date epidemiologice și de populație

Exemple:

- răspândirea bolilor
- statistici pe regiuni
- date despre pandemii

IA modelează evoluția epidemiilor și impactul politicilor de sănătate



Date administrative

Exemple:

- programări
- durată spitalizare
- costuri

IA optimizează managementul spitalelor

Preprocesarea datelor

- **Gestionarea datelor lipsă** → completarea valorilor lipsă sau eliminarea rândurilor incomplete
- **Tratarea valorilor extreme (outliers)** → identificarea și corectarea/eliminarea valorilor anormale
- **Normalizare / scalare** → aducerea valorilor numerice la aceeași scară (ex: 0–1)
- **Codificarea variabilelor categorice** → transformarea textului în numere (ex: „masculin/feminin” → 0/1)

Preprocesarea datelor

- Pentru anumite date complexe (ex: imagini, semnale audio) **caracteristicile nu sunt vizibile direct si trebuie extrase automat sau manual.**
- **Datele tind să fie corelate** între ele ceea ce înseamnă că anumite valori pot fi deduse unele din altele -> reprezentare ineficientă.
- Pentru a simplifica datele pentru algoritmi și pentru a face modelele mai rapide, se folosesc **tehnici de reducere a dimensionalității**, care reduc numărul de caracteristici **fără a pierde informația esențială.**
 - PCA (Principal Component Analysis) – date numerice și tabelare
 - Autoencodere – imagini, audio, text
 - Word embeddings (Word2Vec, GloVe) – text
 - Feature Selection – pentru seturi de date tabelare mari, selectarea variabilelor relevante

PCA (Principal Component Analysis)

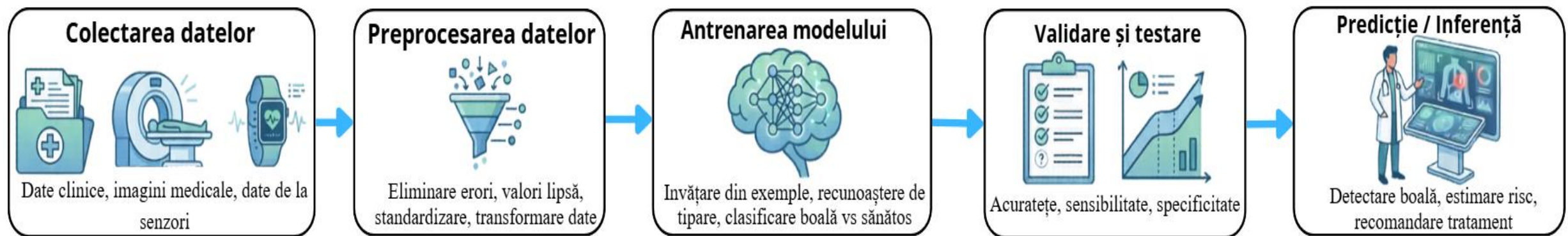
Cum funcționează PCA:

- Pornește de la toate caracteristicile din setul de date și **analizează relațiile dintre ele** (matrice de covarianță)
- **Identifică variabilele corelate** (care conțin informație similară)
- Identifică direcțiile cu variație maximă în date (direcții în care datele sunt cel mai „împrăștiate”, adică varianță maximă)
- Construiește componente principale – variabile noi obținute prin **combinări liniare ale variabilelor inițiale** (vectori proprii).
- **Ordonează componentele în funcție de cantitatea de informație** pe care o păstrează (varianță explicată / valori proprii).
- **Păstrează doar primele componente** și elimină restul, reducând dimensiunea datelor.

Avantaje practice:

- Reduce **timpul de antrenare** pentru algoritmi
- Reduce **riscul de overfitting**
- Reduce numărul de variabile (features) fără să piarda informație important

CUM FUNCTIONEAZĂ SISTEMELE DE IA ÎN SĂNĂTATE



- Sistemele pot fi **actualizate cu date noi**, devenind mai precise în timp.
- În practică, modelul oferă un rezultat sau o recomandare, însă **medicul interpretează rezultatul**, iar **decizia finală aparține medicului**
- În multe aplicații de IA, nu este esențial să știm exact cum a fost luată decizia.
În medicină, acest lucru este diferit:
 - medicii trebuie să înțeleagă decizia modelului
 - deciziile trebuie să fie **explicabile și justificabile** (din punct de vedere legal și etic)

Particularitățile domeniului sănătății



1. **Modelele trebuie să fie mult mai precise și sigure.** O eroare poate duce la diagnostic greșit și poate afecta viața pacientului.



2. **Tipurile de date sunt foarte complexe și variate** - date clinice (tabele), imagini (CT, RMN), text (rapoarte medicale), date de la senzori.



3. În multe aplicații, nu contează „de ce” a decis modelul. În medicină, **medicii trebuie să înțeleagă decizia care trebuie justificată** (legal și etic).

Particularitățile domeniului sănătății



4. **Validarea este foarte strictă**, iar modelele nu pot fi folosite direct după antrenare. Sunt necesare testări clinice, validare pe pacienți reali, comparație cu medici.



5. Confidențialitatea datelor este importantă. **Se impun anonimizarea datelor, securitatea ridicată și respectarea regulilor (ex.: GDPR).**



6. **Apare dezechilibrul în date** (unele boli sunt rare și datele colectate sunt insuficiente), modelul vede foarte puține exemple și poate învăța greșit.



7. **IA nu înlocuiește medicul.** IA face parte dintr-un sistem de suport decizional, iar medicul ia decizia finală.

Evaluarea performantei unui sistem de IA

- Modelul trebuie să **explice deciziile**, să fie de **încredere pentru medici** și să **respecte standardele etice și reglementările medicale**.
- Se verifică **dacă modelul funcționează corect pentru toți pacienții**. Pot apărea probleme precum:
 - performanță mai scăzută pe anumite grupuri
 - bias (sex, vârstă, etnie)
- Modelul trebuie:
 - **comparat cu rezultatul medicilor**
 - **testat în situații reale**
 - **validat clinic**
- Pentru a verifica dacă modelul generalizează bine, se folosesc:
 - **seturi de test separate**
 - **validare pe date din alte spitale / populații**

Nu testăm pe datele de antrenare

Evaluarea performanței unui sistem de IA

- Există două tipuri de greșeli:
 - Fals pozitiv (FP)** → spune că pacientul e bolnav, dar nu este
 - Fals negativ (FN)** → spune că e sănătos, dar este bolnav
- În sănătate: **fals negativ** = de obicei mai grav

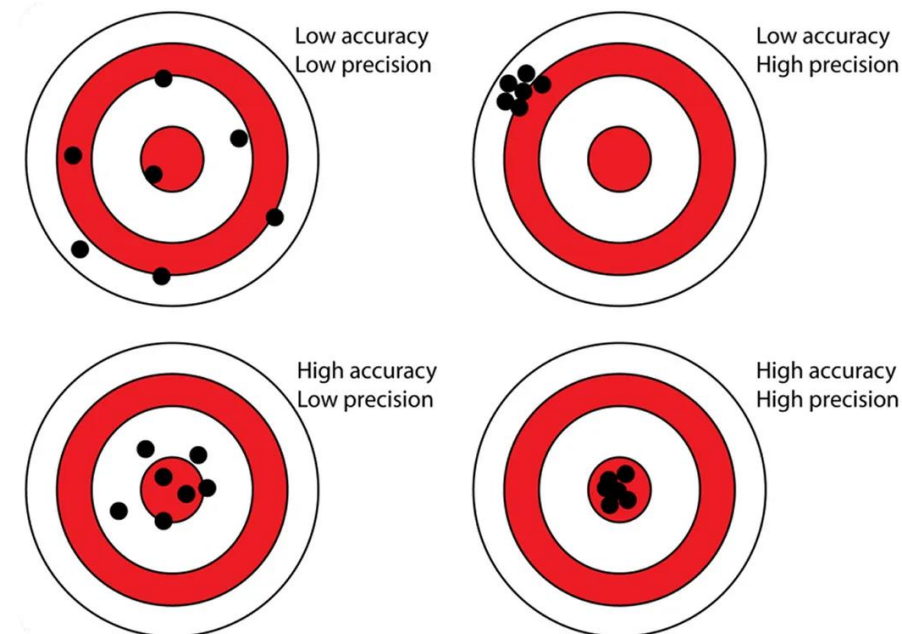
$$\text{Acuratețe} = \frac{TP + TN}{TP + TN + FP + FN} \rightarrow \text{cât de des are dreptate}$$

$$\text{Sensibilitate (recall)} = \frac{TP}{TP + FN} \rightarrow \text{cât de bine detectează cazurile cu boală}$$

$$\text{Specificitate} = \frac{TN}{TN + FP} \rightarrow \text{cât de bine evită alarmele false}$$

$$\text{Precizie} = \frac{TP}{TP + FP} \rightarrow \text{cât de corecte sunt predicțiile pozitive}$$

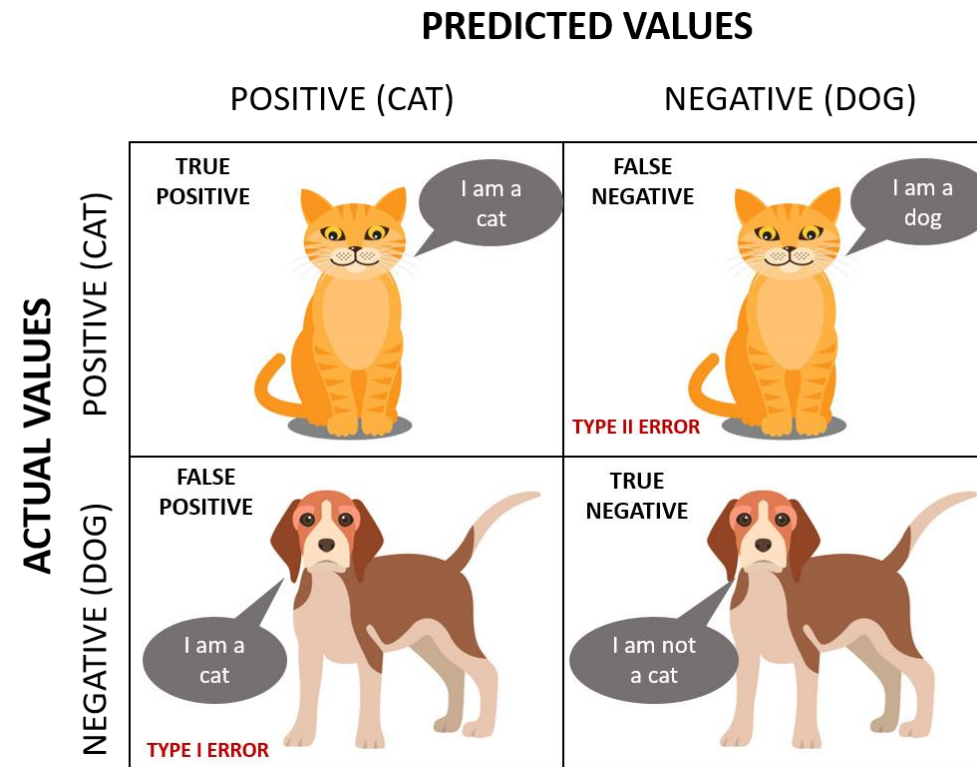
$$\text{Scor F1} = \frac{2 \cdot \text{Precizie} \cdot \text{Sensibilitate}}{\text{Precizie} + \text{Sensibilitate}} \rightarrow \text{echilibru între precizie și sensibilitate}$$



Evaluarea performantei unui sistem de IA

Matricea de confuzie compară valorile reale cu cele prezise de model:

	Predicție Pozitivă	Predicție Negativă
Real Pozitiv	TP (True Positive)	FN (False Negative)
Real Negativ	FP (False Positive)	TN (True Negative)



Multumesc!

ana.vasilevschi@ici.ro

