

# Inteligentă Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

Secția Tehnologia Informației, anul III, 2022-2023  
Cursul 2

# Recapitulare – cursul trecut

1. Aspecte organizatorice legate de cursul de IA
  1. Prezentarea cursului de IA

# Ce vom studia la acest curs de IA?

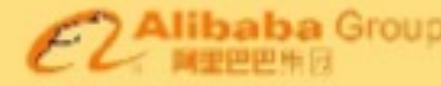
Testul Turing → Subdomenii ale Inteligenței Artificiale:

1. procesarea limbajului natural (comunicare)
2. reprezentarea cunoștințelor (stocare informații)
3. deducție automată (a faptelor pe baza cunoștiințelor)
4. învățare automată (detectare de pattern-uri)
5. vedere artificială (perceperea obiectelor, a scenei)
6. robotică (manipularea obiectelor, mișcare)

Vom studia:

- Învățare automată - primele 7 săptămâni
- Rezolvarea de probleme prin strategii de căutare (informată și neinformată) - următoarele 7 săptămâni

# **Aplicații de succes în Inteligenta Artificială**

	Q1 2009	Q1 2014	Q1 2019
#1	<b>ExxonMobil</b>		 Microsoft
#2	<b>PetroChina</b>		
#3	<b>Walmart</b> 	 Microsoft	 amazon
#4	<b>ICBC</b> 		
#5	 中国移动 China Mobile	<b>BERKSHIRE HATHAWAY INC.</b>	<b>BERKSHIRE HATHAWAY INC.</b>
#6	 Microsoft		
#7	 AT&T		
#8		 	
#9			
#10			
USS	1,900,000,000,000	3,000,000,000,000	6,000,000,000,000

Top 10 Publicly Traded Companies by Market Cap (FT Global 500)

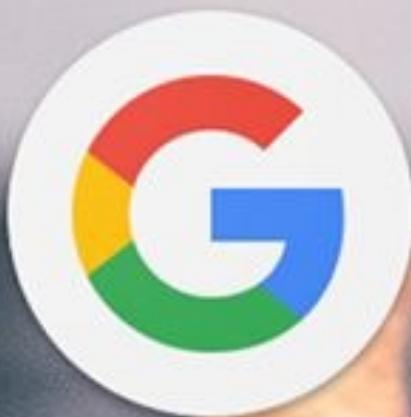


amazon  
alexa

Alexa



Cortana

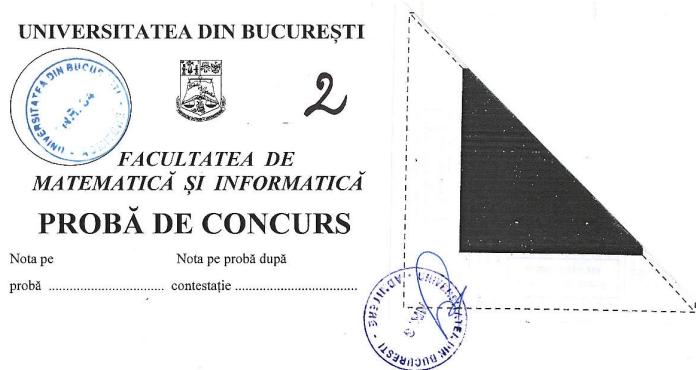


Google



Siri

# Corectarea automată a testelor grilă

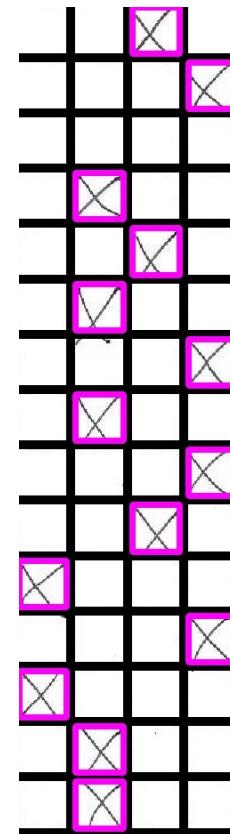


**TEST GRILĂ**

**INFORMATICĂ**  1      **FIZICĂ**  \_\_\_\_\_  
Număr varianță

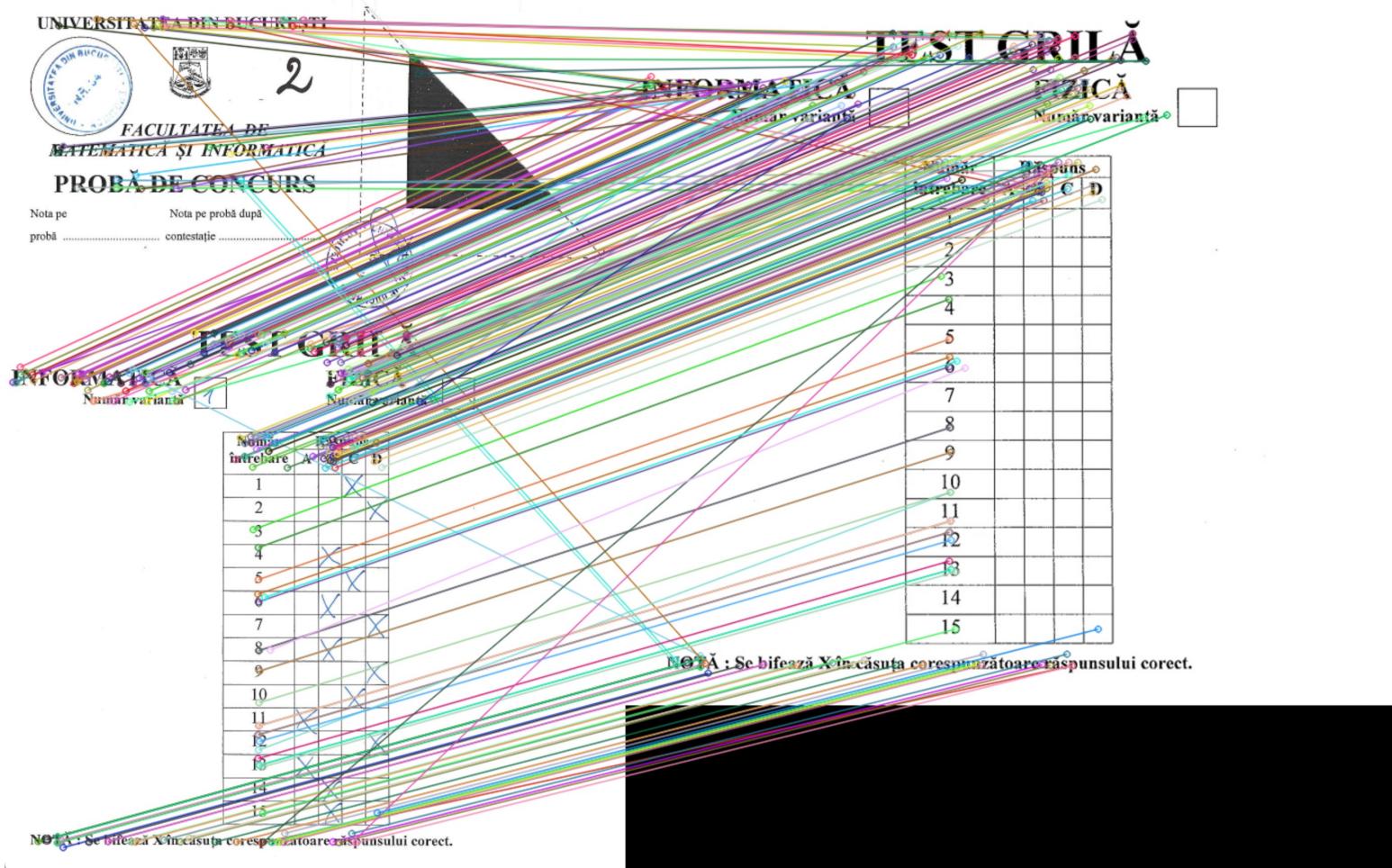
Număr întrebare	A	B	C	D
1			X	
2			X	
3				
4	X			
5		X		
6	X			
7		X		
8	X			
9			X	
10		X		
11	X			
12			X	
13	X			
14		X		
15		X		

NOTĂ : Se bifează X în căsuța corespunzătoare răspunsului corect.



Numar Lucrare	Varianta	0X	1X	mX	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Numar grile corecte	Nota
2	Informatica_1	1	14	0	C	D	B	C	B	D	B	D	C	A	D	A	B	B		2	2.2

# Corectarea automată a testelor grilă



# IA pentru jocuri

- antrenate pentru a găsi cele mai bune strategii de joc
- paradigma învățării pe bază de recompensă (reinforcement learning)
  - recompensă +/- pentru câștigarea/pierderea unui joc

# Şah: Deep Blue vs Kasparov

- Deep Blue (proiectat de IBM) îl învinge în 1997 pe Kasparov după ce în prealabil pierduse în 1996

- **1996: Kasparov câştigă**

“I could feel – I could smell  
kind of intelligence across 1

The 1996 match				
Game #	White	Black	Result	Comment
1	Deep Blue	Kasparov	1–0	
2	Kasparov	Deep Blue	1–0	
3	Deep Blue	Kasparov	½–½	Draw by mutual agreement
4	Kasparov	Deep Blue	½–½	Draw by mutual agreement
5	Deep Blue	Kasparov	0–1	Kasparov offered a draw after the 23rd move.
6	Kasparov	Deep Blue	1–0	

*Result: Kasparov–Deep Blue: 4–2*

- **1997: Deep Blue câştigă**

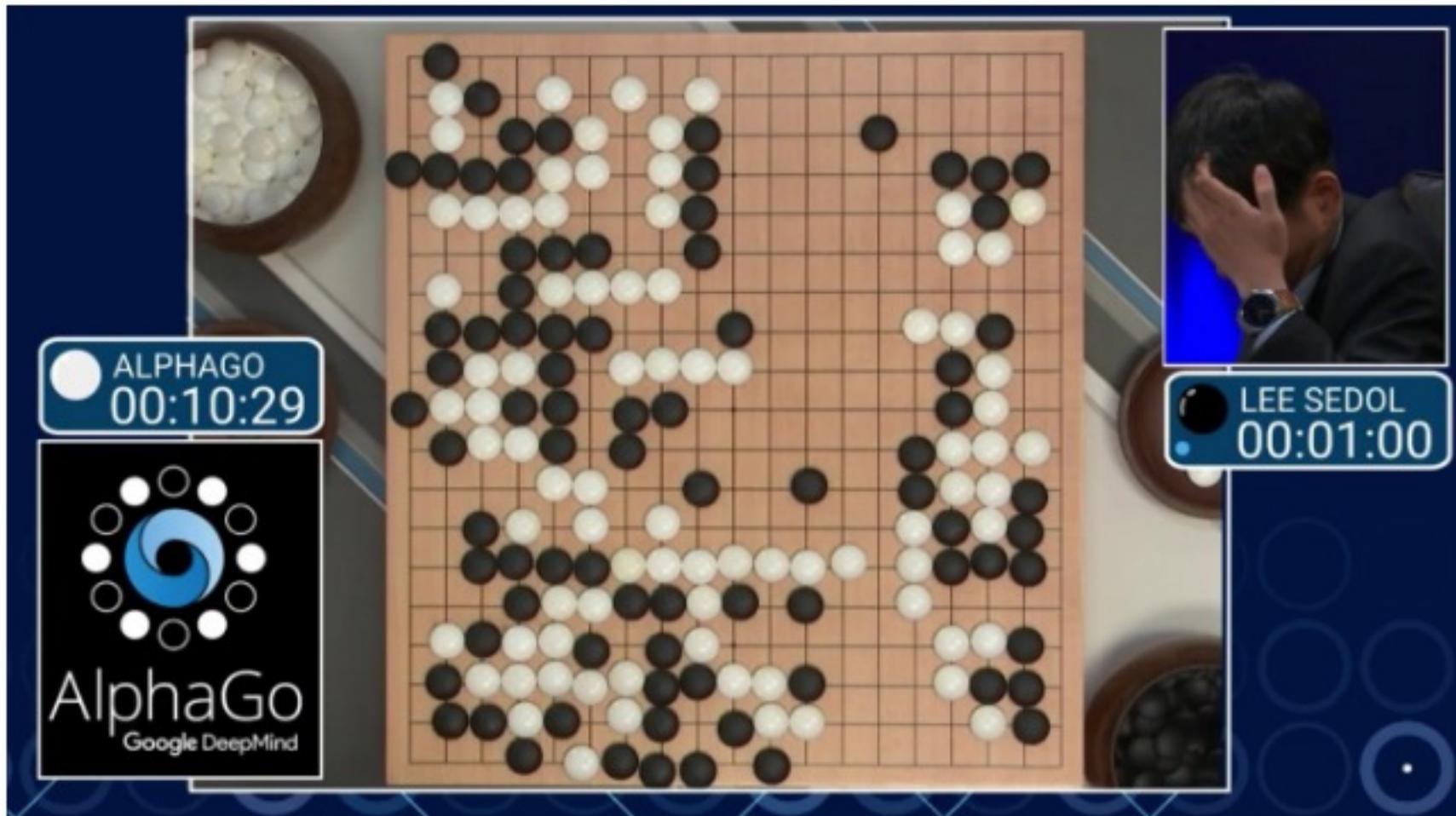
“Deep Blue hasn't proven anythi

The 1997 rematch				
Game #	White	Black	Result	Comment
1	Kasparov	Deep Blue	1–0	
2	Deep Blue	Kasparov	1–0	
3	Kasparov	Deep Blue	½–½	Draw by mutual agreement
4	Deep Blue	Kasparov	½–½	Draw by mutual agreement
5	Kasparov	Deep Blue	½–½	Draw by mutual agreement
6	Deep Blue	Kasparov	1–0	

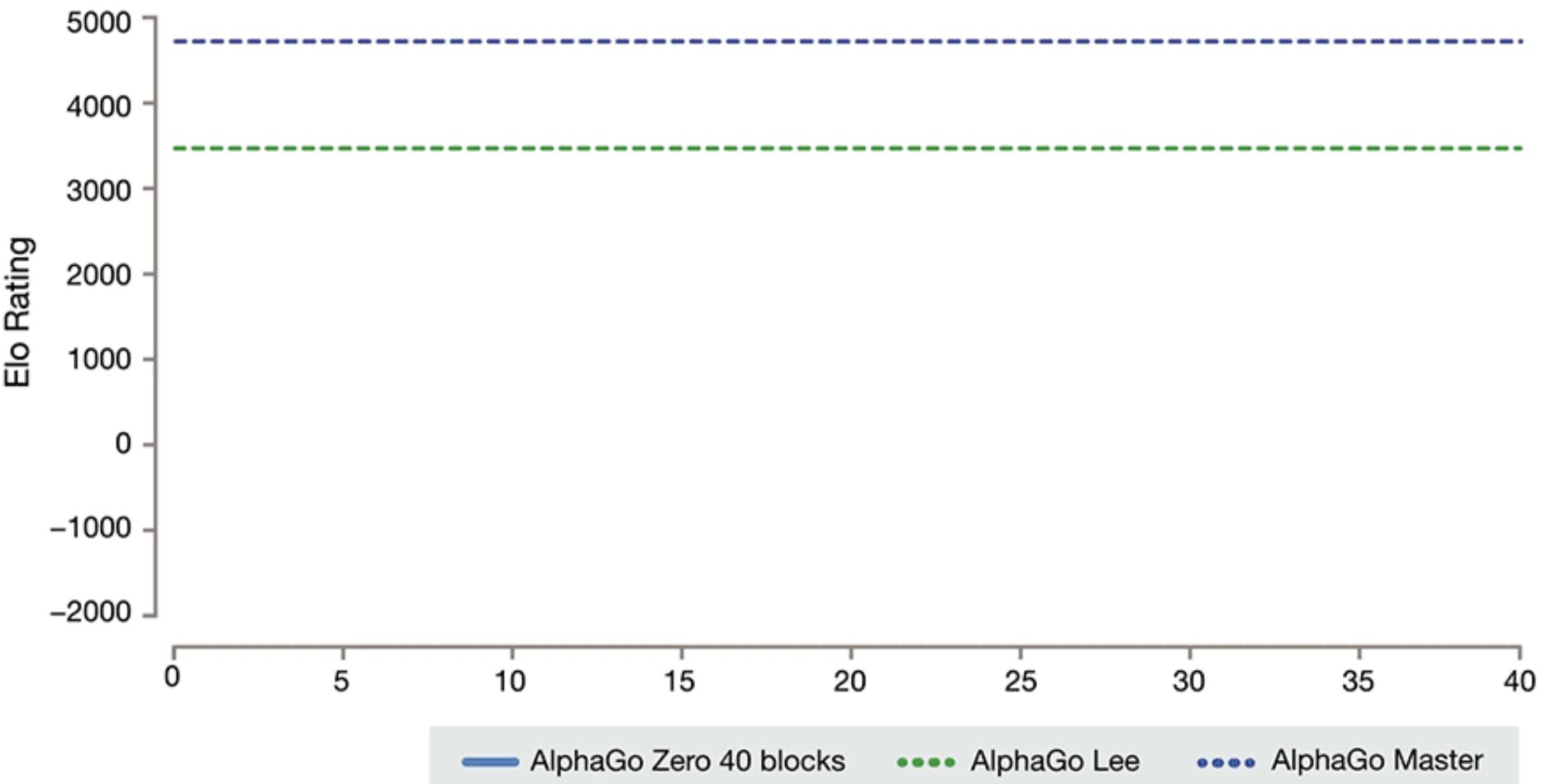
*Result: Deep Blue–Kasparov: 3½–2½*

# Go: AlphaGo vs Lee Sedol

- AlphaGo proiectat de Google DeepMind îl învinge în 2016 campionul mondial la GO



# Go: AlphaGo vs AlphaGo Zero



# Alte aplicații

## Spam filtering



## Machine translation



## Speech recognition



## Advertising and ad placement



## Recommendation systems

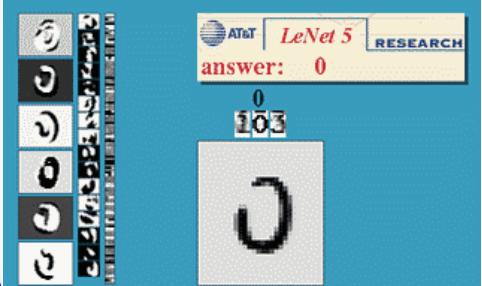


## Driving assistance systems

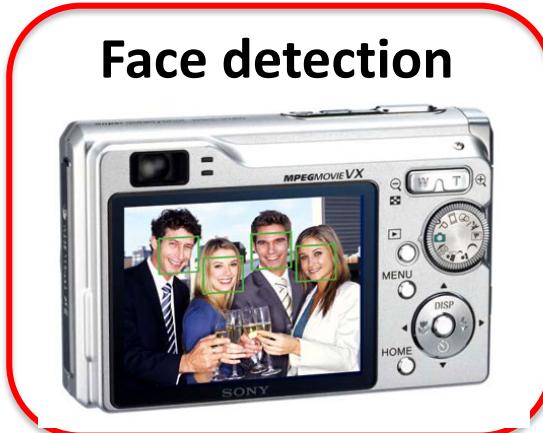


# Alte aplicații

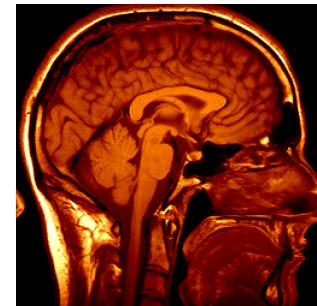
## OCR



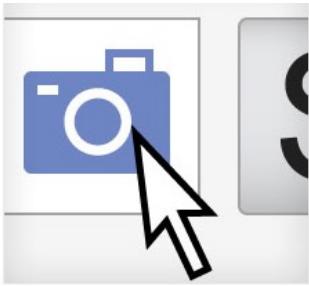
## Face detection



## Medical



## Visual search



## XBOX Kinect



## Video referee



# Cuprinsul cursului de azi

1. Învățarea Automată: definiție și terminologie
2. Paradigme de învățare
3. Metrici de măsurare a performanței pentru clasificare
4. Modelul celor mai apropiati k vecini (k-nearest neighbors)

# Învățarea Automată: definiție și terminologie

# Ce este învățarea automată?

**Învățarea automată** – domeniu care studiază modul în care mașinile (calculatoarele) pot fi înzestrate cu abilitatea de a învăța, fără ca aceasta să fie programată în mod explicit.

Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed.

Arthur Samuel, 1959

Un program este înzestrat cu abilitatea de a învăța dacă performanța sa  $P$  pe un task (problemă)  $T$  se îmbunătăște pe măsură ce acumulează mai multă experiență (date)  $E$ .

A computer program is said to learn from experience  $E$  with respect to some task  $T$  and some performance measure  $P$ , if its performance on  $T$ , as measured by  $P$ , improves with experience  $E$ .

Tom Mitchell, 1997

**Învățarea** se referă la recunoașterea unor tipare / structuri (patterns) complexe și la luarea deciziilor inteligente bazate pe observațiile din **date**

# Abilitatea de a învăța

- Arthur Samuel (1959) a scris un program pentru a juca dame (probabil primul program bazat pe conceptul de învățare)
- Programul a jucat împotriva lui însuși 10.000 de jocuri
- Programul a fost conceput să găsească ce poziții ale tablei de joc erau bune sau rele în funcție de probabilitatea de a câștiga sau pierde
- În acest caz:

$E = 10000$  de jocuri

$T =$  joacă dame

$P =$  câștigă sau nu



# Când se folosește învățarea automată?

- Se aplică în situații în care este foarte greu (imposibil) să definim un set de reguli de mână / să scriem un program
- Cum ați scrie un program pe calculator care vrea să găsească mașinile într-o imagine?



# Când se folosește învățarea automată?

- Se aplică în situații în care este foarte greu (imposibil) să definim un set de reguli de mână / să scriem un program
- Cum ați scrie un program pe calculator care să filtreze mailurile spam?

The screenshot shows a list of flagged messages from a spam filter interface. The messages are listed in two sections: 'This month' and above it.

Category	Message Content
! The CHOICE Home Warranty	Keep Your Home Cool with \$50 Off! – Keep Your Home Co...
! Final, Final (2)	Request To Terminate Your Google™ Account Has Bee...
This month	
! Mr. Tony Adams (2)	Congratulation You Have Won – You have Won \$2.5Million ...
! ZENITH BANK PLC	COMPENSATION UNIT, IN AFFILIATION WITH THE UNITE...
! John Mill	UNITED NATION AND EUROPEAN UNION OFFICIAL WIN...

At the bottom left, there is a green circular icon containing a white letter 'J' and the email address 'urytopk4@crocus.ocn.ne.jp'.

# Când se folosește învățarea automată?

- Se aplică în situații în care este foarte greu (imposibil) să definim un set de reguli de mână / să scriem un program
- probleme pentru care soluțiile tradiționale necesită multă muncă manuală și liste interminabile de reguli, greu de menținut actualizate la schimbările repetitive
  - detectarea de spam, traducere automată
- probleme aproape “imposibil” de programat în abordarea de programare tradițională
  - detectarea de obiecte, recunoașterea vorbirii
- găsirea de corelații în seturi mari de date

# Terminologie în învățarea automată

- vom folosi un **set de date (dataset)** cu informații despre două tipuri de fructe

Greutate (g)	Culoare	Textură	pH	Etichetă
84	Verde	Moale	3.5	Măr
121	Portocaliu	Aspră	3.9	Portocală
85	Roșu	Moale	3.3	Măr
101	Portocaliu	Moale	3.7	Portocală
111	Verde	Aspră	3.5	Măr
...	...	...	...	...
117	Roșu	Aspră	3.4	Portocală

# Terminologie în învățarea automată

- o **etichetă (label, target)** este ceea ce încercăm să prezicem

Greutate (g)	Culoare	Textură	pH	Etichetă
84	Verde	Moale	3.5	Măr
121	Portocaliu	Aspră	3.9	Portocală
85	Roșu	Moale	3.3	Măr
101	Portocaliu	Moale	3.7	Portocală
111	Verde	Aspră	3.5	Măr
...	...	...	...	...
117	Roșu	Aspră	3.4	Portocală

# Terminologie în învățarea automată

- o **etichetă (label, target)** este ceea ce încercăm să prezicem
  - dacă este discretă, se mai numește **clasă**, problemă de *clasificare*
  - dacă este continuă, problemă de *regresie*
  - diferența dintre clasificare și regresie: măsurarea erorii

Greutate (g)	Culoare	Textură	pH	Etichetă
84	Verde	Moale	3.5	Măr
121	Portocaliu	Aspră	3.9	Portocală
85	Roșu	Moale	3.3	Măr
101	Portocaliu	Moale	3.7	Portocală
111	Verde	Aspră	3.5	Măr
...	...	...	...	...
117	Roșu	Aspră	3.4	Portocală

# Terminologie în învățarea automată

- un **exemplu** (**example**, **sample point**) este o instanță particulară a datelor (**data point**)
- poate include sau nu o etichetă (**label**) – date *etichetate* vs. date *neetichetate*

Greutate (g)	Culoare	Textură	pH	Etichetă
84	Verde	Moale	3.5	Măr
121	Portocaliu	Aspră	3.9	Portocală
85	Roșu	Moale	3.3	Măr
101	Portocaliu	Moale	3.7	Portocală
111	Verde	Aspră	3.5	Măr
...	...	...	...	...
117	Roșu	Aspră	3.4	Portocală

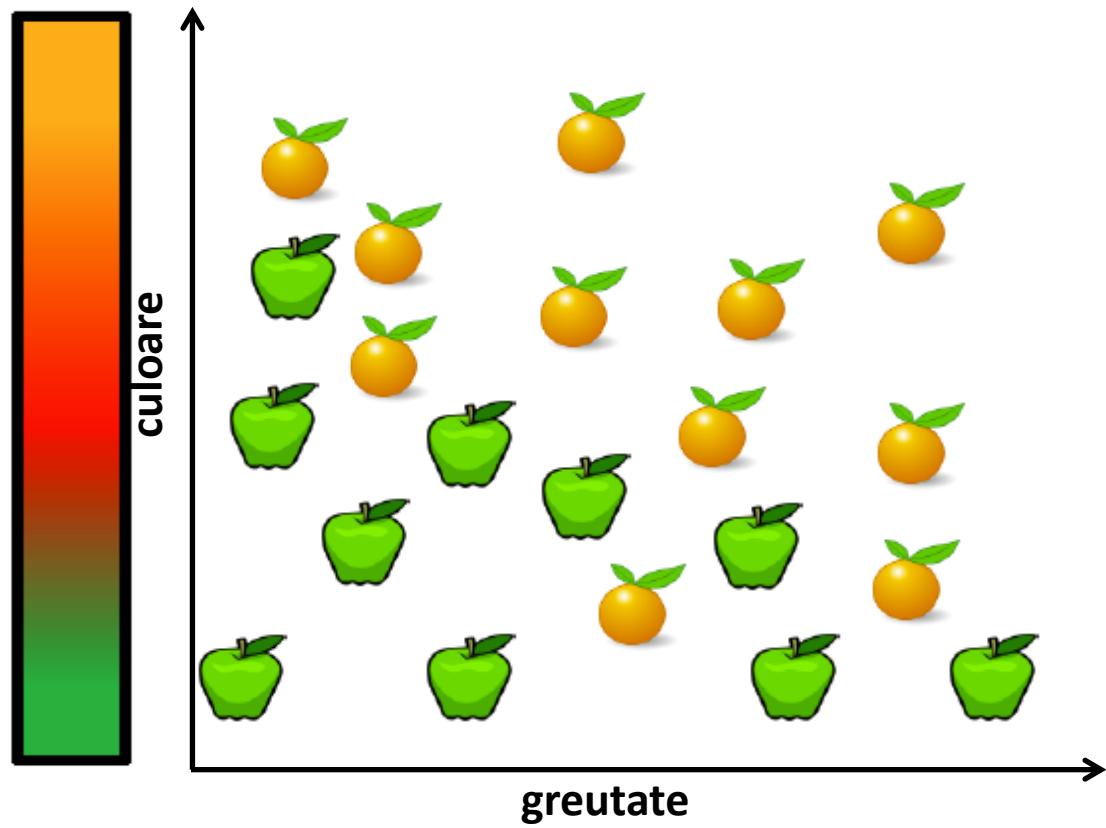
# Terminologie în învățarea automată

- o **caracteristică** (**feature, attribute**) este o proprietate măsurabilă pe baza observațiilor
- toate caracteristicile unui exemplu (direct măsurabile sau obținute) formează un **vector de caracteristici** (**feature vector**) – de obicei au valori numerice

Greutate (g)	Culoare	Textură	pH	Etichetă
84	Verde	Moale	3.5	Măr
121	Portocaliu	Aspră	3.9	Portocală
85	Roșu	Moale	3.3	Măr
101	Portocaliu	Moale	3.7	Portocală
111	Verde	Aspră	3.5	Măr
...	...	...	...	...
117	Roșu	Aspră	3.4	Portocală

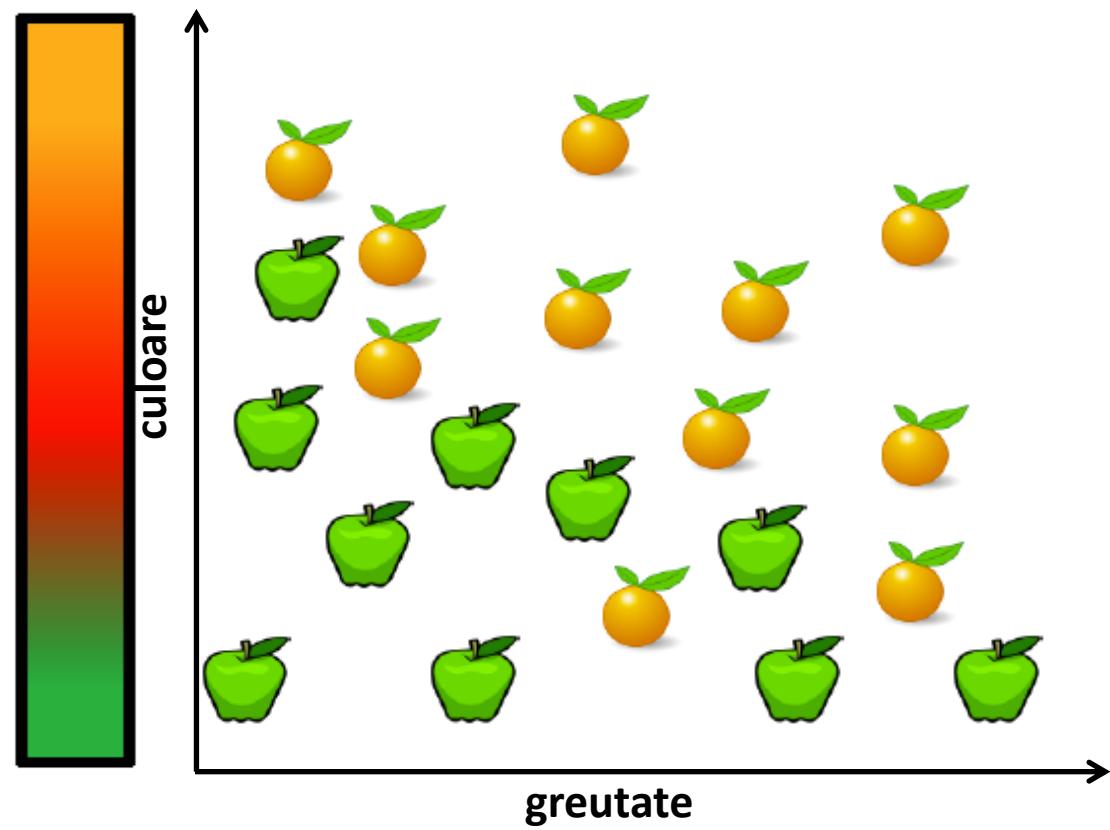
# Terminologie în învățarea automată

- un **model** sau o **ipoteză** este o relație *determinată* dintre caracteristici și etichete



# Terminologie în învățarea automată

- un **model** sau o **ipoteză** este o relație *determinată* dintre caracteristici și etichete
- determinarea relației pe baza unor exemple se numește **antrenarea modelului (training, fitting)**



# Terminologie în învățarea automată

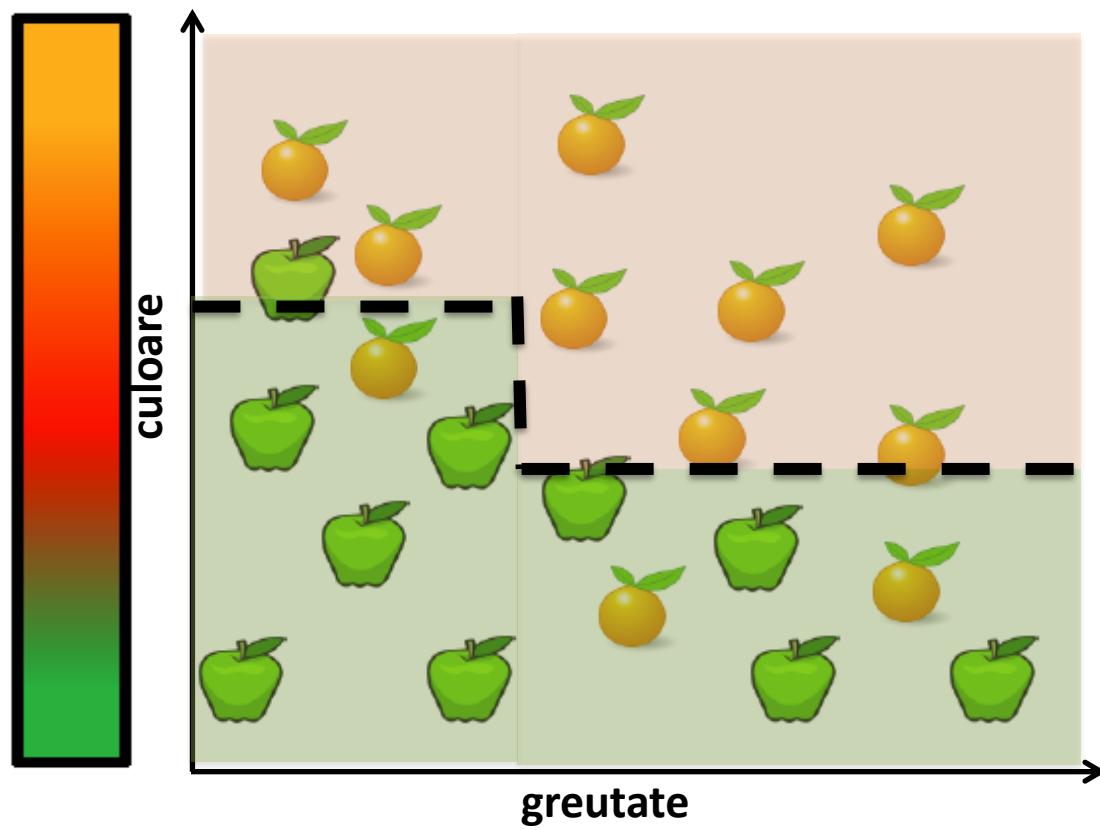
- un **model** sau o **ipoteză** este o relație *determinată* dintre caracteristici și etichete
- determinarea relației pe baza unor exemple se numește **antrenarea modelului (training, fitting)**

Pentru clasificare, fiecărui model îi corespunde o **graniță de separare (decision boundary)**

Etichete prezise

Etichete prezise		
Etichete reale	Green Apple	Orange
Green Apple	9	1
Orange	3	8

Matrice de confuzie



# Terminologie în învățarea automată

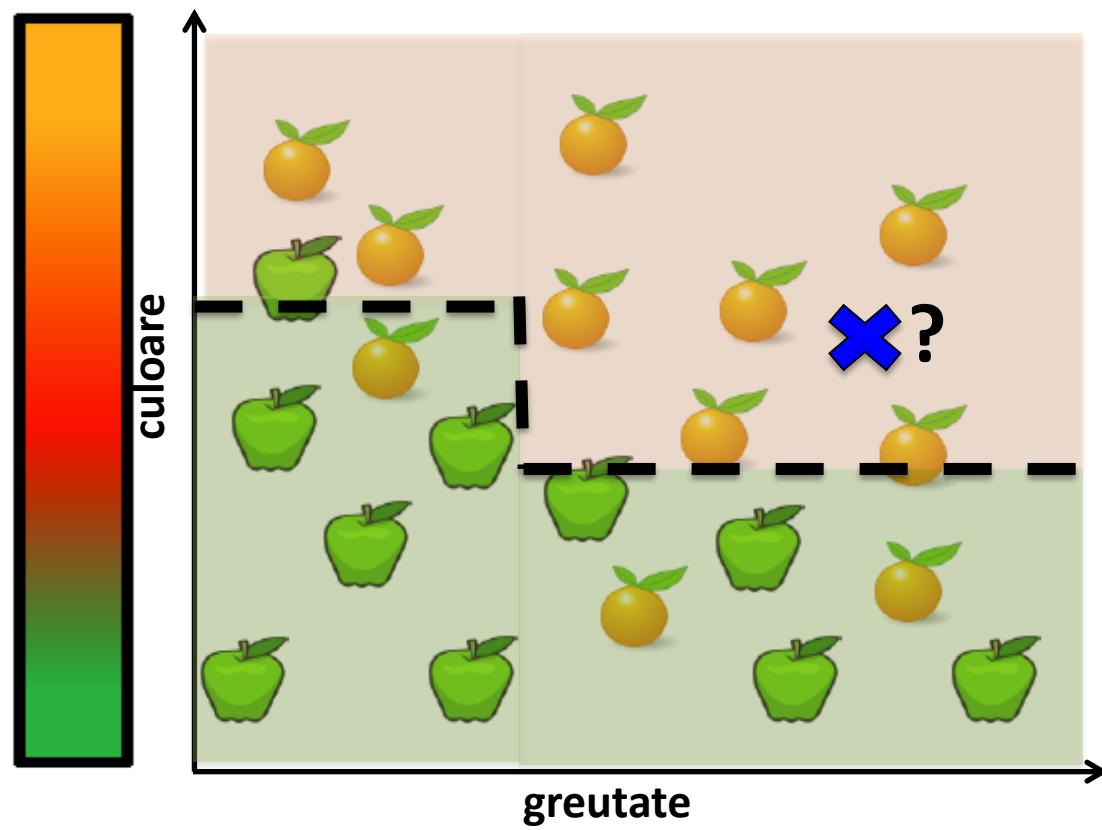
- în timpul **inferenței**, modelul antrenat este folosit la realizare de predicții pentru exemple test, necunoscute

Pentru clasificare, fiecărui model îi corespunde o **graniță de separare** (decision boundary)

Etichete prezise

Etichete prezise		
Etichete reale	green apple	orange
green apple	9	1
orange	3	8

Matrice de confuzie



# Terminologie în învățarea automată

- în timpul **inferenței**, modelul antrenat este folosit la realizare de predicții pentru exemple test, necunoscute
- algoritmii de învățare pot determina ipoteze diferite, cu granițe de separare de forme diferite

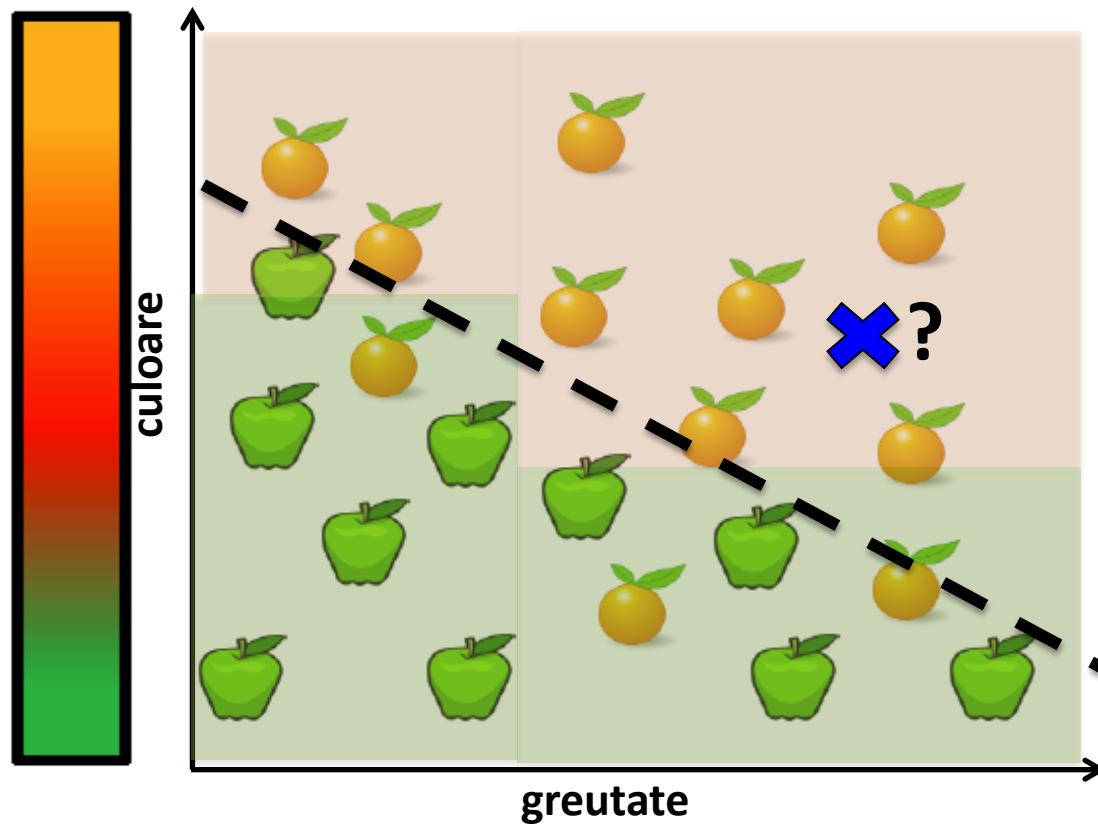
Pentru clasificare, fiecărui model îi corespunde o **graniță de separare** (decision boundary)

Etichete prezise



Etichete reale	apple	orange
apple	10	0
orange	3	8

Matrice de confuzie



# Terminologie în învățarea automată

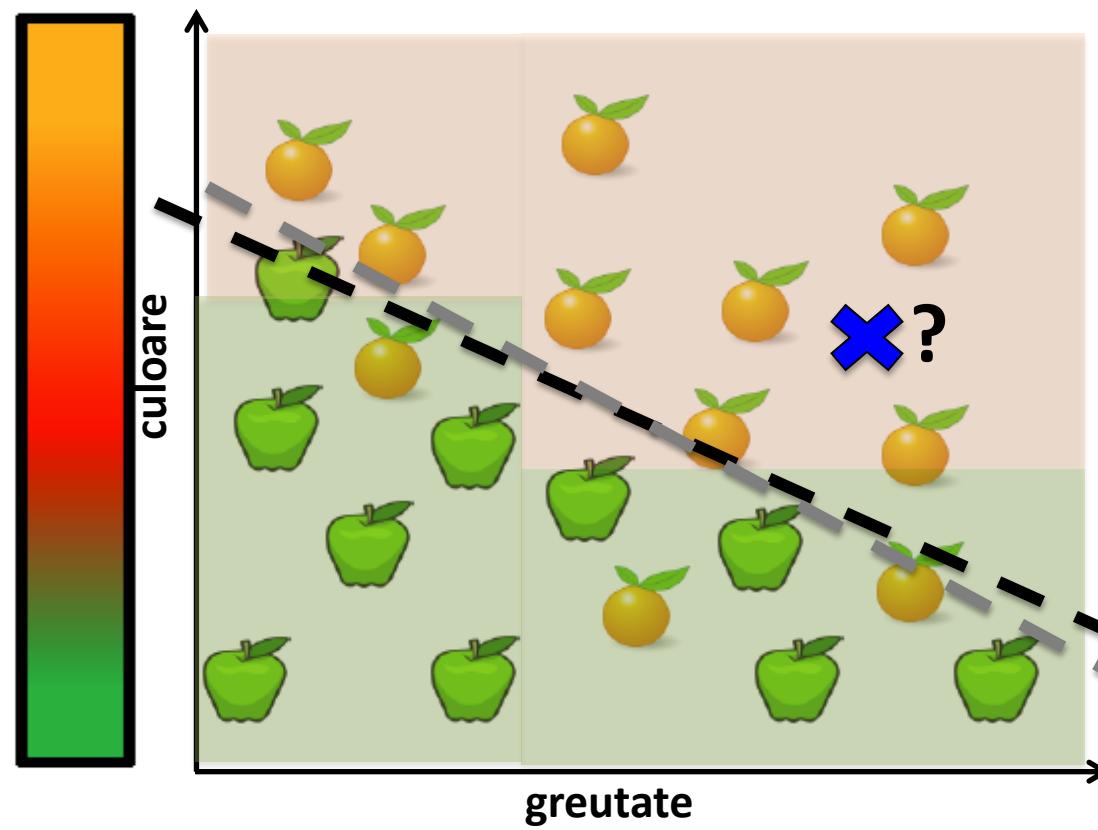
- algoritmii de învățare pot determina ipoteze diferite, cu granițe de separare de forme diferite
- algoritmii pot avea hiperparametri care controlează felul în care decurge învățarea și se reflectă asupra modelului învățat

Pentru clasificare, fiecărui model îi corespunde o **graniță de separare (decision boundary)**

Etichete prezise

Etichete prezise		
Etichete reale	green apple	orange
green apple	10	0
orange	3	8

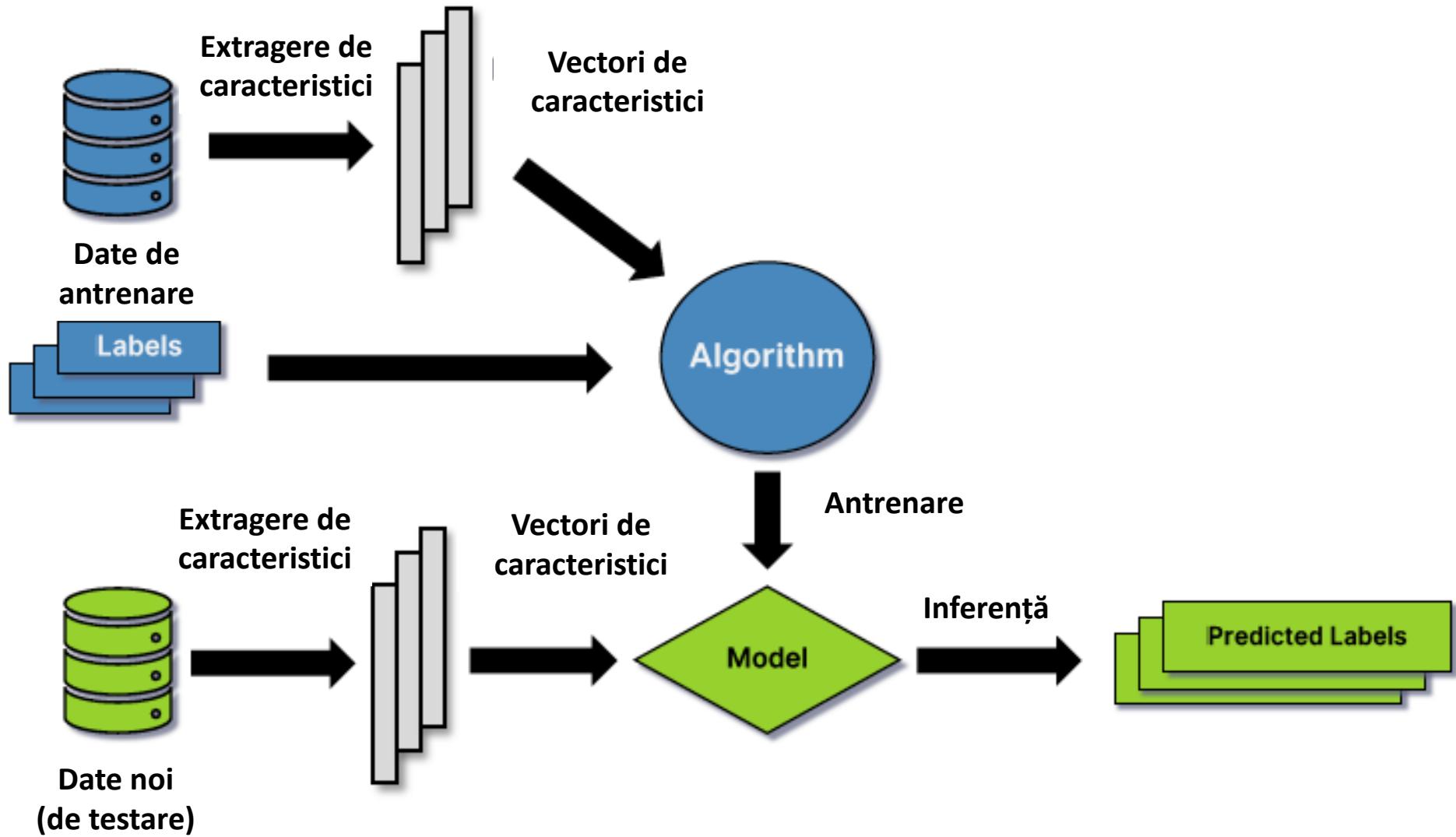
Matrice de confuzie



# Terminologie în învățarea automată

- **etichetă** (label) → ce încercăm să prezicem
- **caracteristică** (feature) → proprietate măsurabilă a unui punct din exemplele din date
- **model** (ipoteză) → relația dintre caracteristici și etichete
- **antrenare** → stabilirea relației pe baza unei multimi de puncte
- **inferență** → realizarea de predicții pe date necunoscute
- **algoritm** → definește cum se face învățarea, poate fi restricționat la o anumită mulțime de ipoteze, fie prin alegerea unui anumit spațiu, fie pe baza hiperparametrilor

# Fluxul tipic în învățarea automată



# Paradigme de învățare

# Învățarea supervizată

- vrem să prezicem o etichetă
  - avem nevoie de date etichetate

# Învățarea supervizată

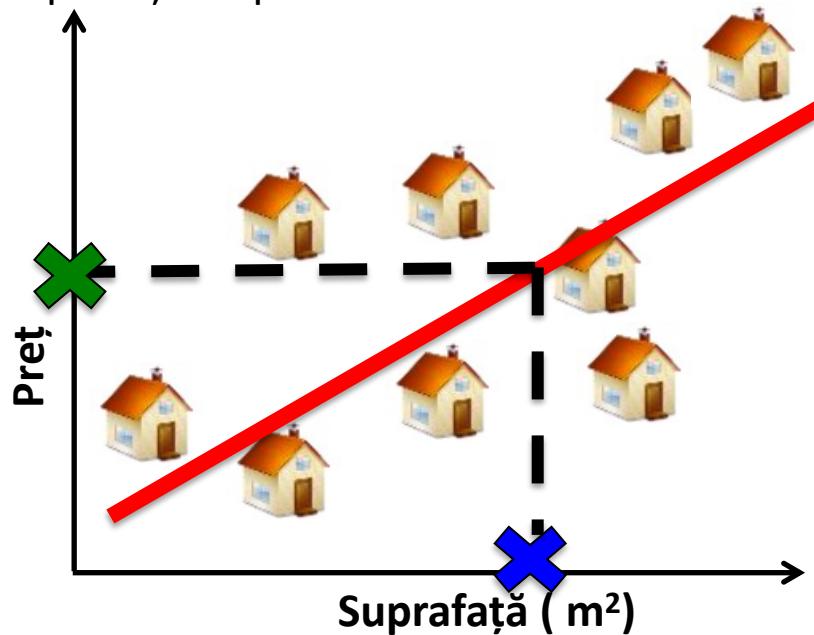
- vrem să prezicem o etichetă
  - avem nevoie de date etichetate

**Regresie:** eticheta este o valoare continuă

Exemplu: prezicerea prețului unei case pe baza suprafeței exprimată în  $m^2$

**Clasificare:** eticheta este o valoare discretă

Exemplu: prezicerea tipului de fruct pe baza culorii și a greutății



Regresie liniară

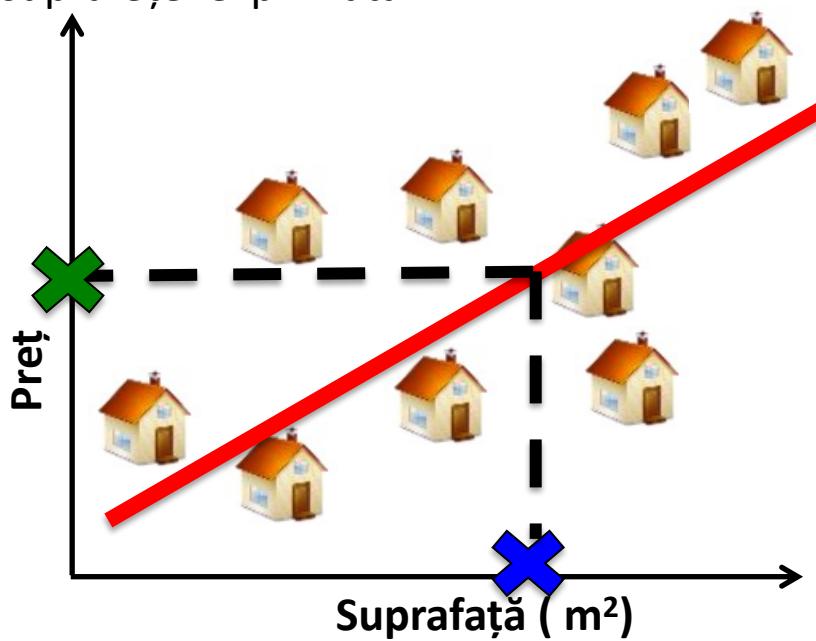
Regresie pe baza kNN

# Învățarea supervizată

- vrem să prezicem o etichetă
  - avem nevoie de date etichetate

**Regresie:** eticheta este o valoare continuă

Exemplu: prezicerea prețului unei case pe baza suprafeței exprimată în  $m^2$

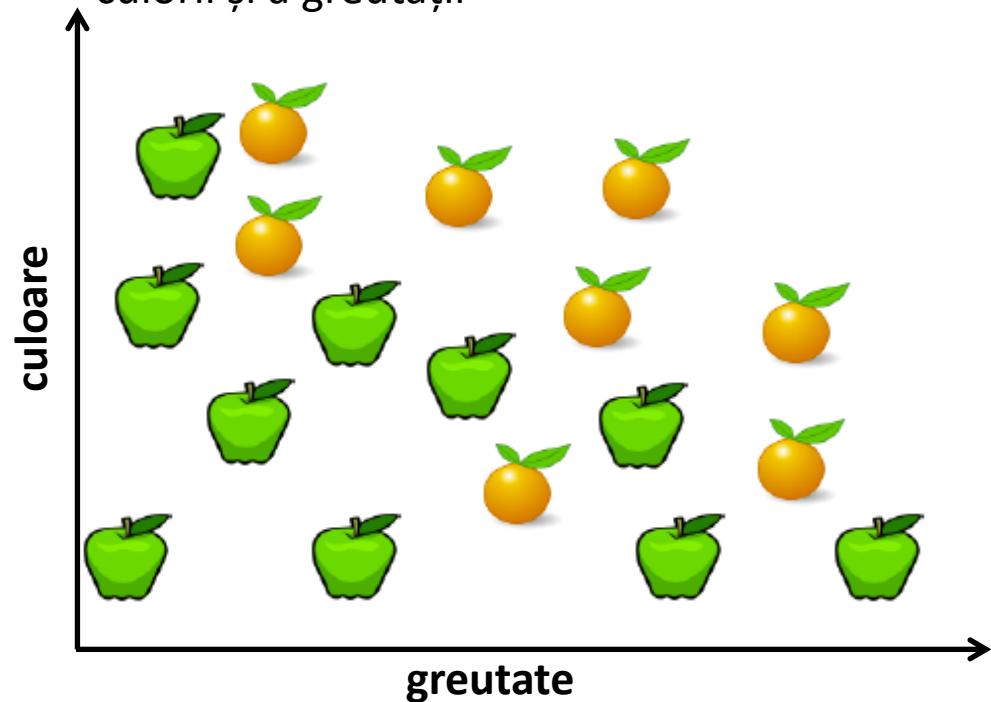


**Regresie liniară**

**Regresie pe baza kNN**

**Clasificare:** eticheta este o valoare discretă

Exemplu: prezicerea tipului de fruct pe baza culorii și a greutății

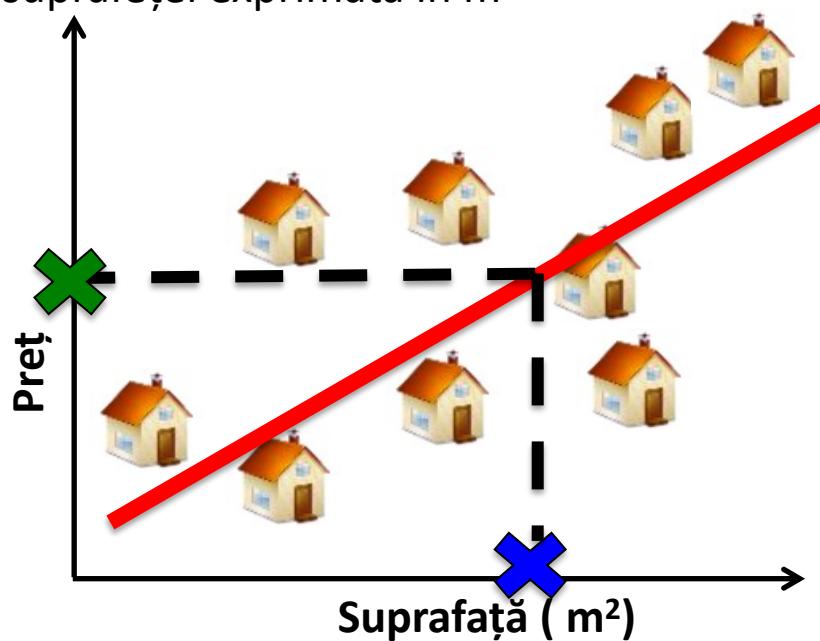


# Învățarea supervizată

- vrem să prezicem o etichetă
  - avem nevoie de date etichetate

**Regresie:** eticheta este o valoare continuă

Exemplu: prezicerea prețului unei case pe baza suprafeței exprimată în  $m^2$

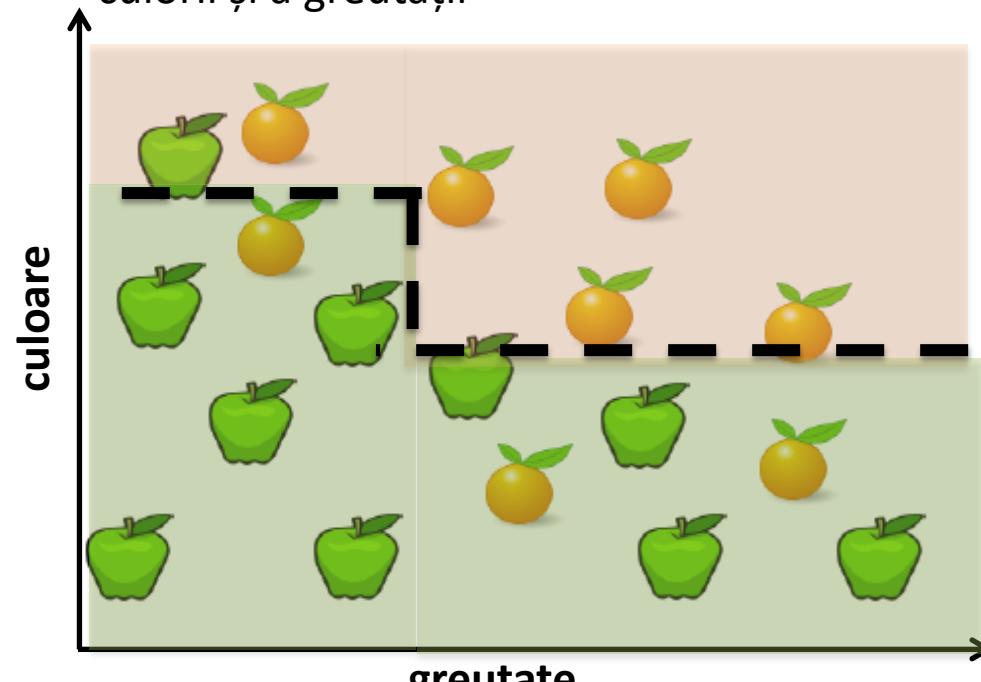


**Regresie liniară**

Regresie pe baza kNN

**Clasificare:** eticheta este o valoare discretă

Exemplu: prezicerea tipului de fruct pe baza culorii și a greutății



Cei mai apropiati k vecini (kNN),  
SVMs, Rețele neuronale

# Învățarea nesupervizată

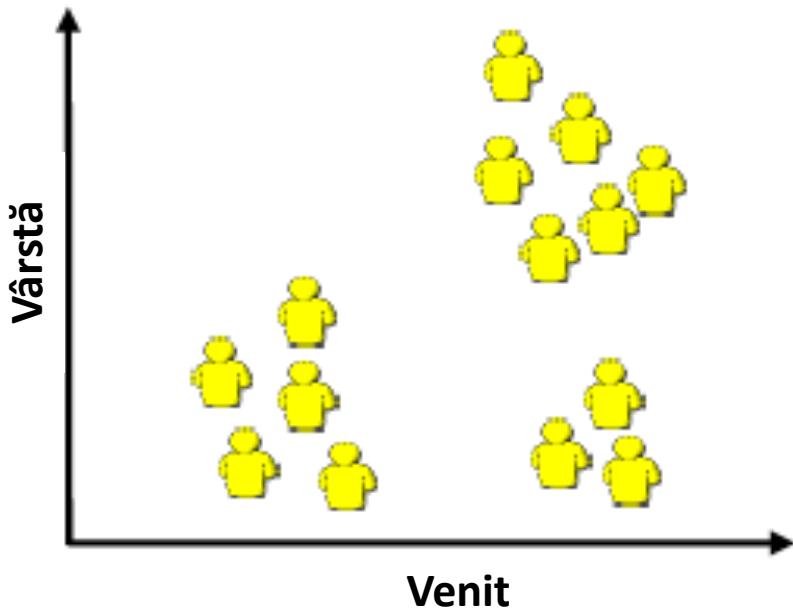
- nu trebuie să prezicem o etichetă, vrem să găsim cum sunt structurate datele

# Învățarea nesupervizată

- nu trebuie să prezicem o etichetă, vrem să găsim cum sunt structurate datele

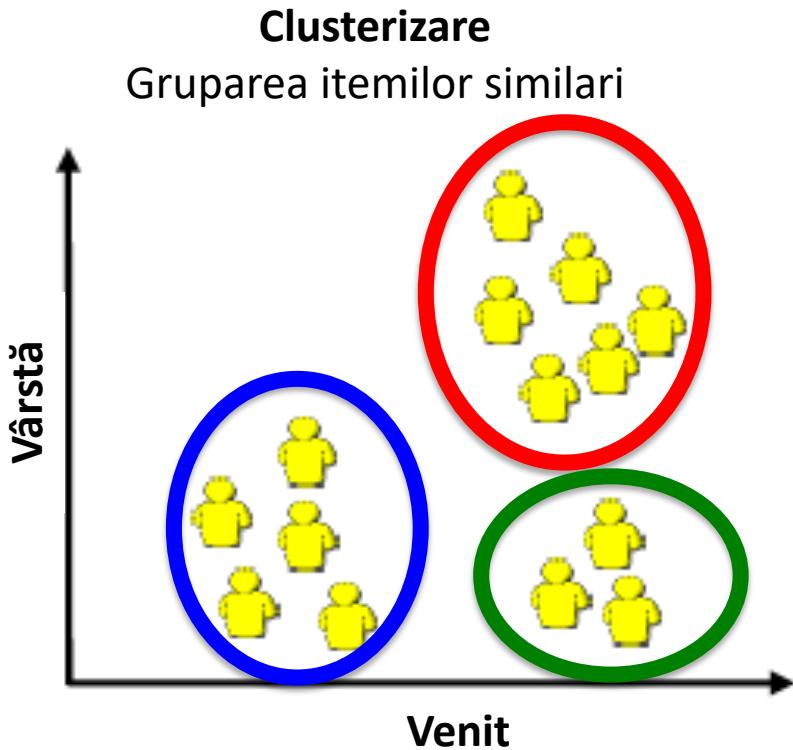
## Clusterizare

Gruparea itemilor similari



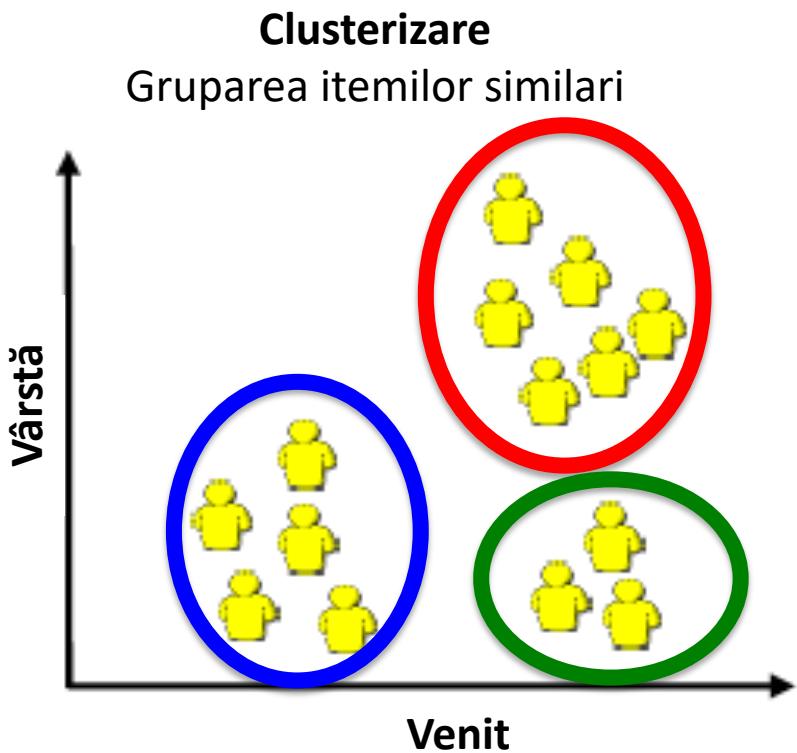
# Învățarea nesupervizată

- nu trebuie să prezicem o etichetă, vrem să găsim cum sunt structurate datele



# Învățarea nesupervizată

- nu trebuie să prezicem o etichetă, vrem să găsim cum sunt structurate datele

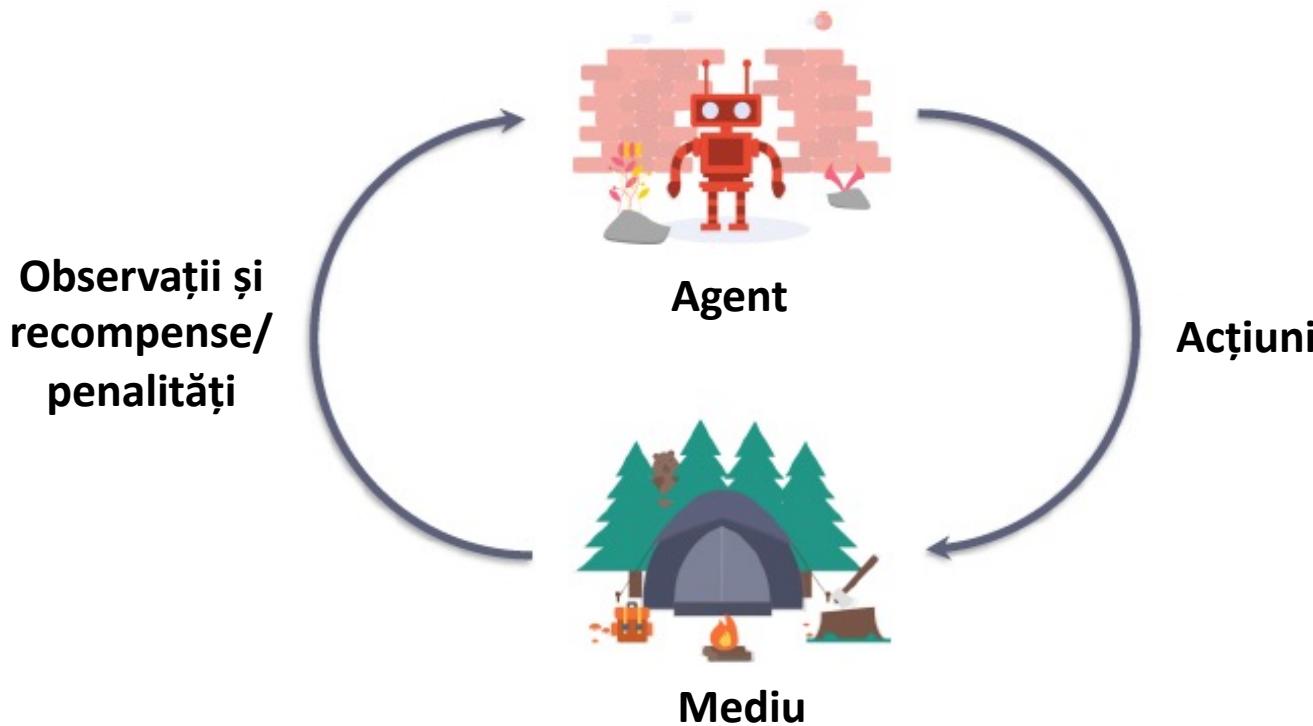


**Căutarea regulilor de asociere**  
Gruparea pattern-urilor frecvente  
și a corelațiilor



# Învățarea pe bază de recompense (reinforcement learning)

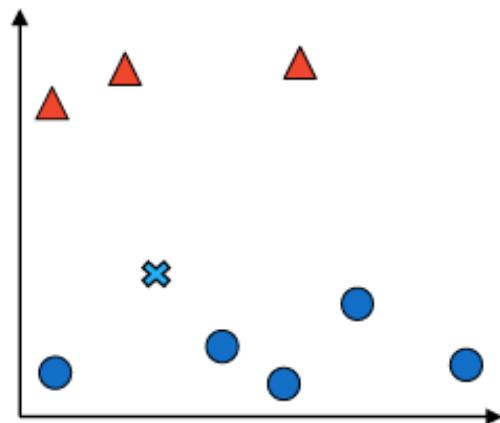
- nu există etichete, ci numai recompense sau penalități pentru realizarea unor acțiuni



# Învățarea semi-supervizată

- vrem să prezicem o etichetă
- avem la dispoziție numai un număr mic de exemple etichetate

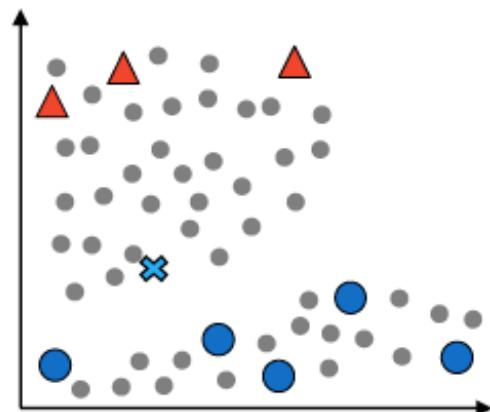
▲ or ● ?



# Învățarea semi-supervizată

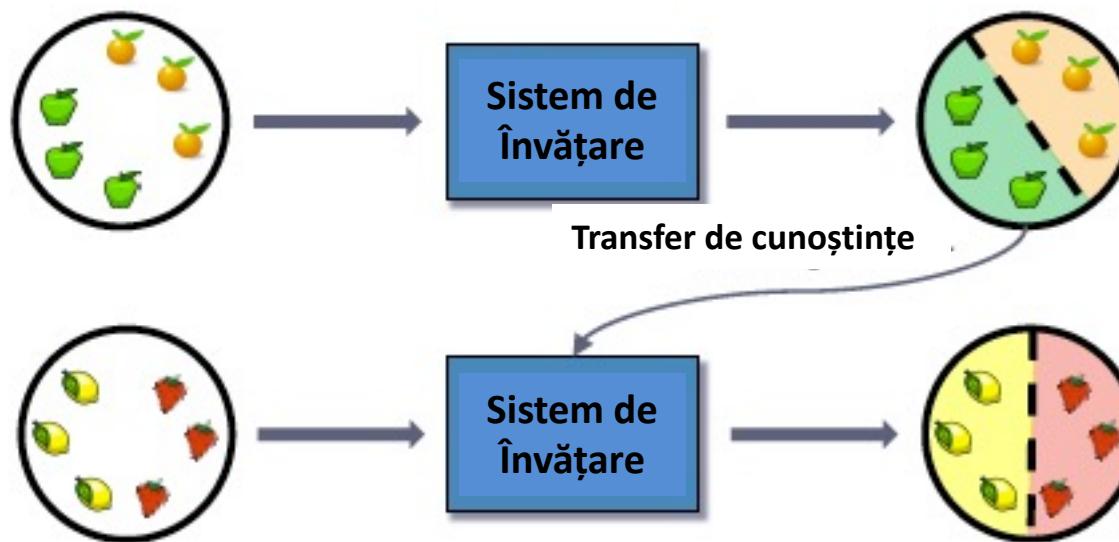
- vrem să prezicem o etichetă
- avem la dispoziție numai un număr mic de exemple etichetate
  - + un număr mare de exemple neetichetate
- datele neetichetate pot îmbunătăți performanțele algoritmilor antrenați în manieră supervizată

▲ or ● ?



# Învățarea prin transfer

- folosim cunoștințele dobândite din rezolvarea unei probleme pentru rezolvarea altei probleme înrudite
- paradigmă folosită în Deep Learning
  - folosirea unui model care a fost pre-antrenat pe un set de date cu multe exemple



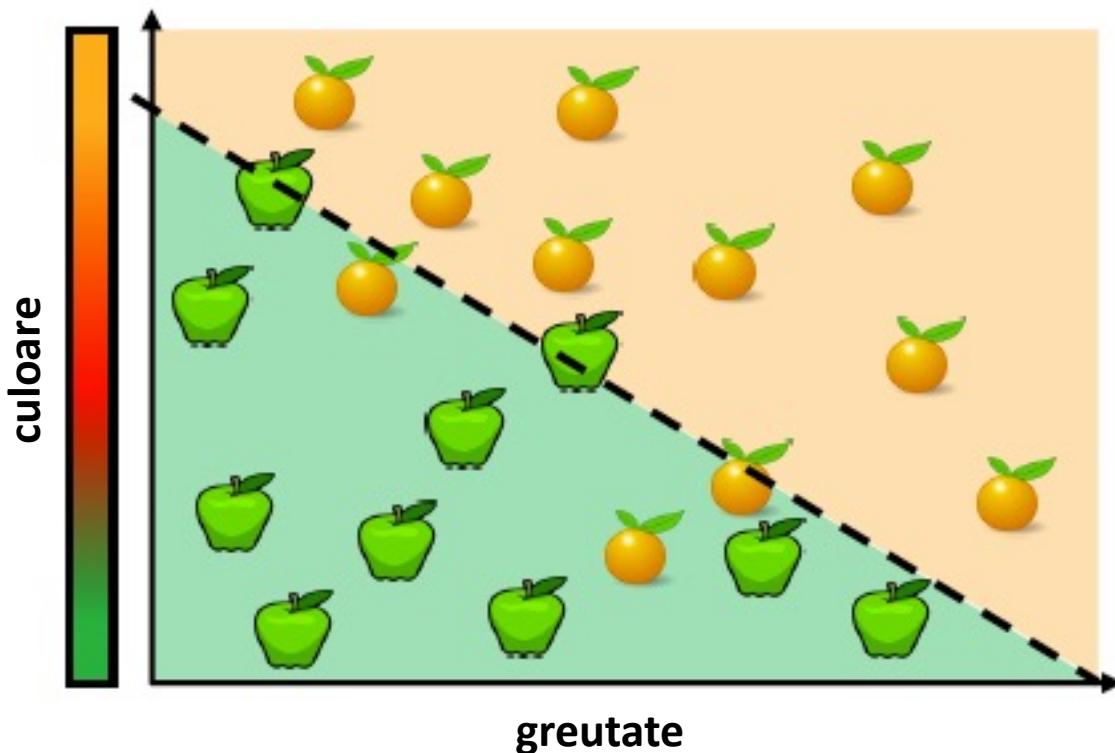
# Paradigme de învățare

- **Învățare Supervizată** → prezicem o etichetă
  - etichetă continuă → **Regresie**
  - etichetă discretă → **Clasificare**
- **Învățarea Nesupervizată** → descoperirea de structuri în date
  - gruparea itemilor similari → **Clusterizare**
  - pattern-uri frecvente → **Căutarea regulilor de asociere**
- **Învățare pe bază de recompense** → fără etichete, ci numai recompense sau penalități pentru realizarea unor acțiuni
- **Învățare semi-supervizată** → câteva date etichetate, multe neetichetate
- **Învățarea prin transfer** → folosirea unui model antrenat pe o problemă pentru învățarea mai rapidă pentru altă problemă

# Metrici de măsurare a performanței pentru clasificare

# Matricea de confuzie

- O matrice de confuzie este un tabel care descrie performanța unui model de clasificare prin precizarea numărului de erori care apar când exemplele sunt clasificate greșit



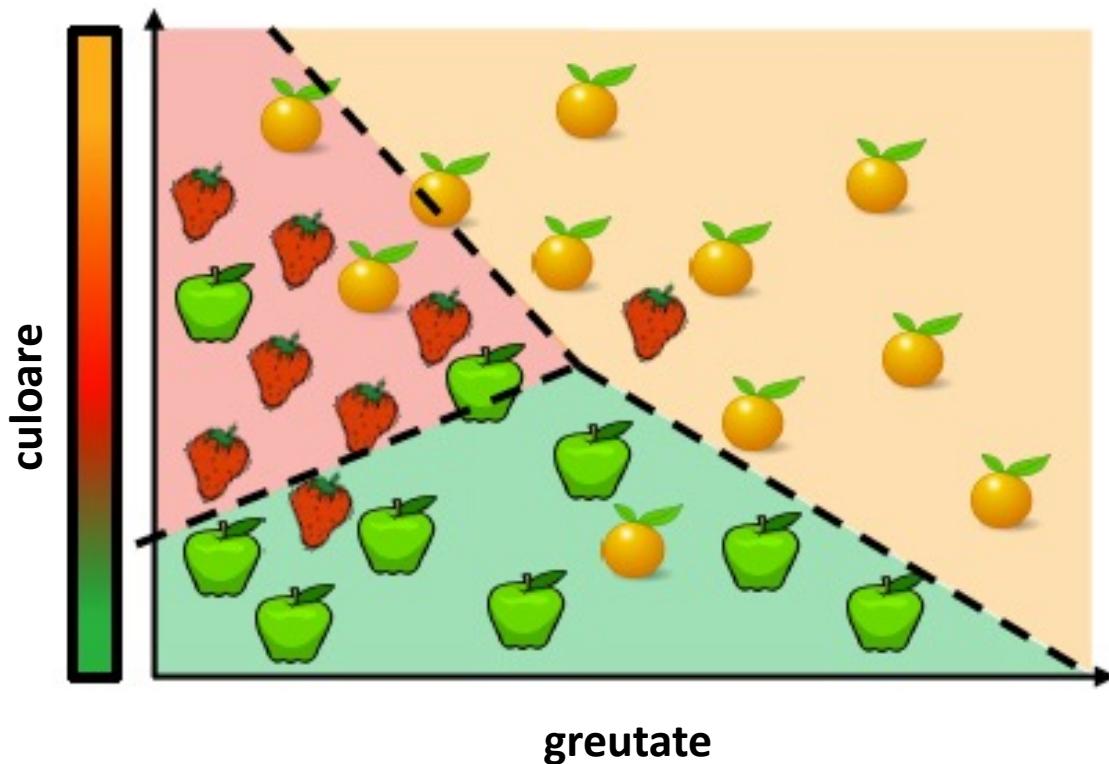
**Etichete prezise**

Etichete reale		
	9	1
	3	8

**Matricea de confuzie**

# Matricea de confuzie

- O matrice de confuzie este un tabel care descrie performanța unui model de clasificare prin precizarea numărului de erori care apar când exemplele sunt clasificate greșit

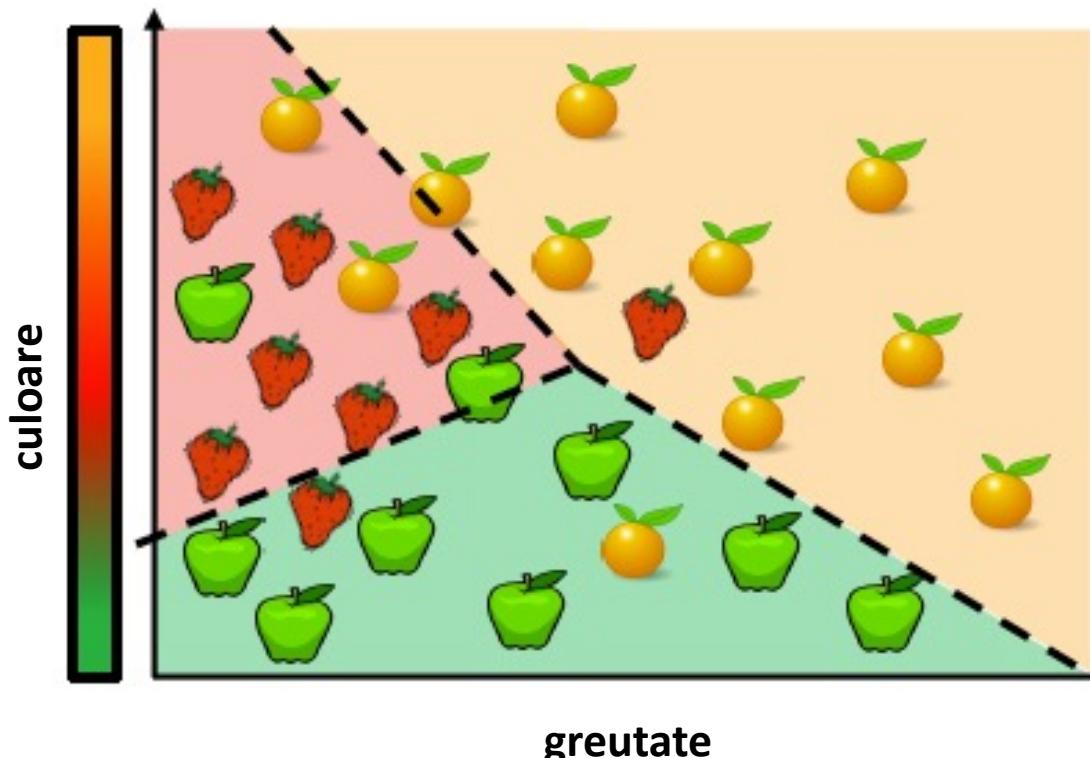


Etichete prezise			
Etichete reale	Apple	Orange	
Apple	7	0	2
Orange	1	8	2
Strawberry	1	1	6

Poate fi folosită pentru  
orice număr de clase

# Acuratețea

- Acuratețea unui model măsoară procentul de exemple din numărul total de exemple clasificate corecte de către model (elemente diagonala principală vs elemente toată matricea)



$$\text{Acuratețea} = 21/28 = 75\%$$

Apples	Oranges	Strawberries
7	0	2
1	8	2

Apples      Oranges      Strawberries

Apples      Oranges      Strawberries

Apples      Oranges      Strawberries

# Probleme cu acuratețea

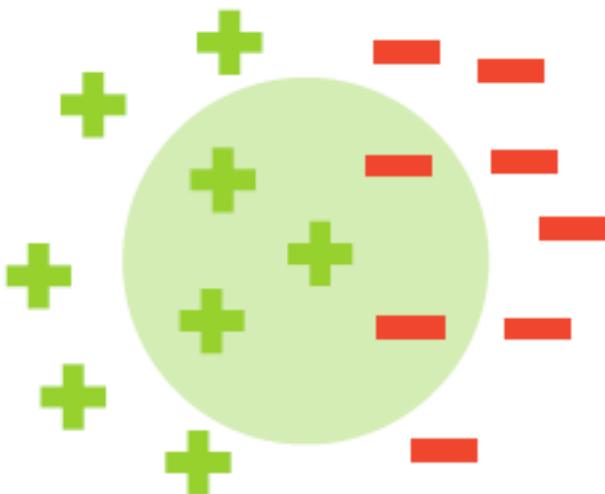
- Presupunem că vrem să antrenăm un clasificator pentru a detecta dacă un pacient are o boală extrem de rară
  - din 1000 de pacienți, numai 20 au o asemenea boală rară (2% grad de incidentă a bolii)
- Considerăm clasificatorul care prezice întotdeauna că pacientul este sănătos
  - are 98% acuratețe!!!
  - care este problema?

# Probleme cu acuratețea

- Acuratețea nu ține cont de câteva aspecte:
  - setul de date este puternic debalansat
    - 98% (pacienți sănătoși) vs. 2% (pacienți bolnavi)
  - cele două clase sunt conceptual diferite
    - o clasă este “pozitivă” – reprezintă clasa de care suntem interesați
    - o clasă este “negativă” – reprezintă starea implicită, nu suntem interesați
  - există două tipuri de erori, care diferă prin gravitatea lor
    - eroare de tipul 1 – “fals pozitiv” – un pacient este clasificat ca fiind bolnav deși este el sănătos
    - eroare de tipul 2 – “fals negativ” – un pacient este clasificat ca fiind sănătos deși el este bolnav
    - în funcție de problemă, una dintre erori poate fi mai gravă decât cealaltă

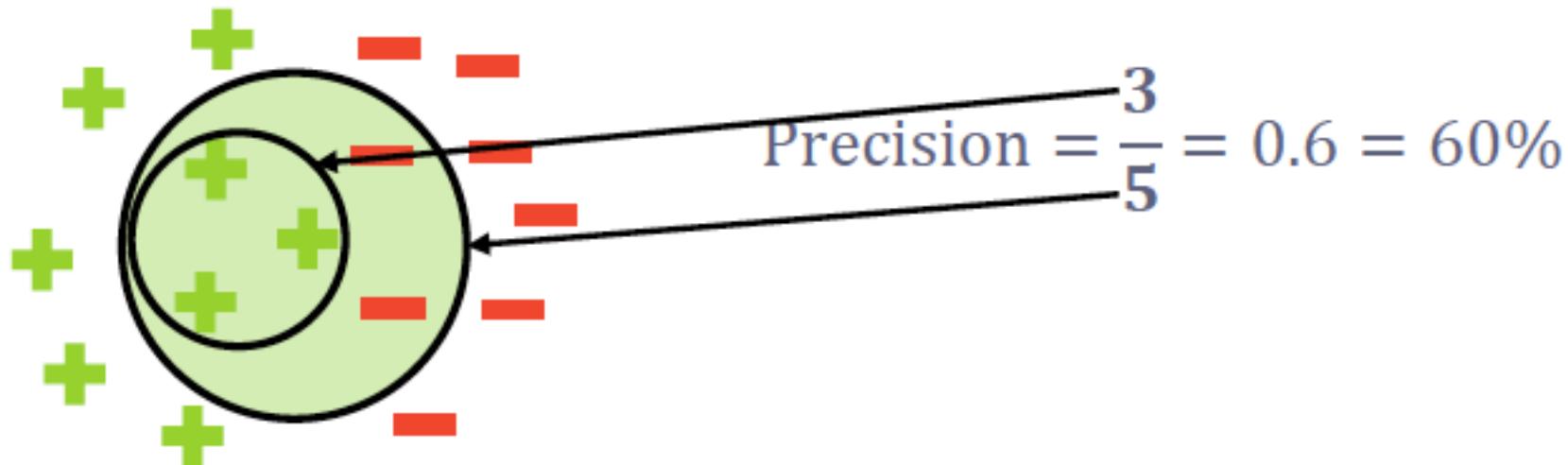
# Precizie și Recall

- Pentru problemele de clasificare în care avem o clasă pozitivă (de care suntem interesați), metricile de precizie și recall sunt mult mai informative decât acuratețea.
- **Precizia:** procentul din itemi selectați de model (clasa lor prezisă este clasa pozitivă) care sunt relevanți (fac parte din clasa pozitivă, cea care ne interesează)
- **Recall:** procentul din itemii relevanți care sunt selectați



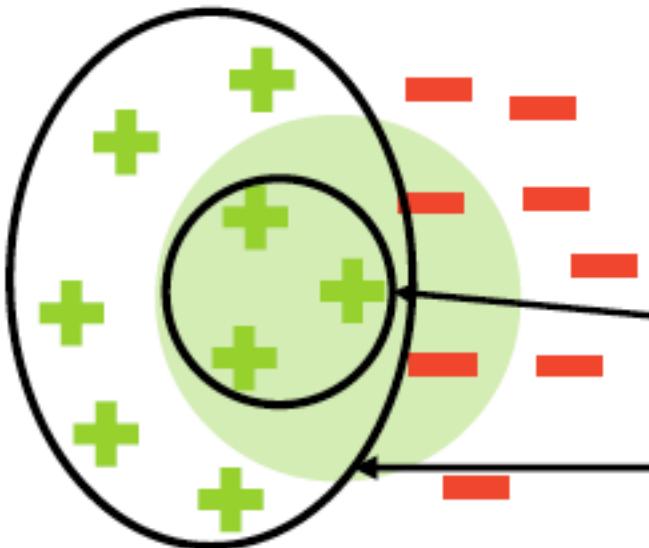
# Precizie și Recall

- Pentru problemele de clasificare în care avem o clasă pozitivă (de care suntem interesați), metricile de precizie și recall sunt mult mai informative decât acuratețea.
- **Precizia:** procentul din **itemi selectați de model** (clasa lor prezisă este clasa pozitivă) **care sunt relevanți** (fac parte din clasa pozitivă, cea care ne interesează)
- **Recall:** procentul din itemii relevanți care sunt selectați



# Precizie și Recall

- Pentru problemele de clasificare în care avem o clasă pozitivă (de care suntem interesați), metricile de precizie și recall sunt mult mai informative decât acuratețea.
- **Precizia:** procentul din itemi selectați de model (clasa lor prezisă este clasa pozitivă) care sunt relevanți (fac parte din clasa pozitivă, cea care ne interesează)
- **Recall:** procentul din **itemii relevanți care sunt selectați**



$$\text{Precision} = \frac{3}{5} = 0.6 = 60\%$$

$$\text{Recall} = \frac{3}{8} = 0.375 = 37.5\%$$

# Precizie și Recall

- Pentru problemele de clasificare în care avem o clasă pozitivă (de care suntem interesați), metricile de precizie și recall sunt mult mai informative decât acuratețea.
- **Precizia:** procentul din itemi selectați de model (clasa lor prezisă este clasa pozitivă) care sunt relevanți (fac parte din clasa pozitivă, cea care ne interesează)
- **Recall:** procentul din **itemii relevanți care sunt selectați**
- **Scorul  $F_1$**  : media armonică pentru **Precizie și Recall**

# Acuratețe, Precizie și Recall

Etichete reale

		Etichete prezise	
		+	-
+	+	TP	FN
	-	FP	TN
Matrice de confuzie			

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

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

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$F_1 = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2(\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}}$$

TP – true positive

FN – false negative

FP – false positive

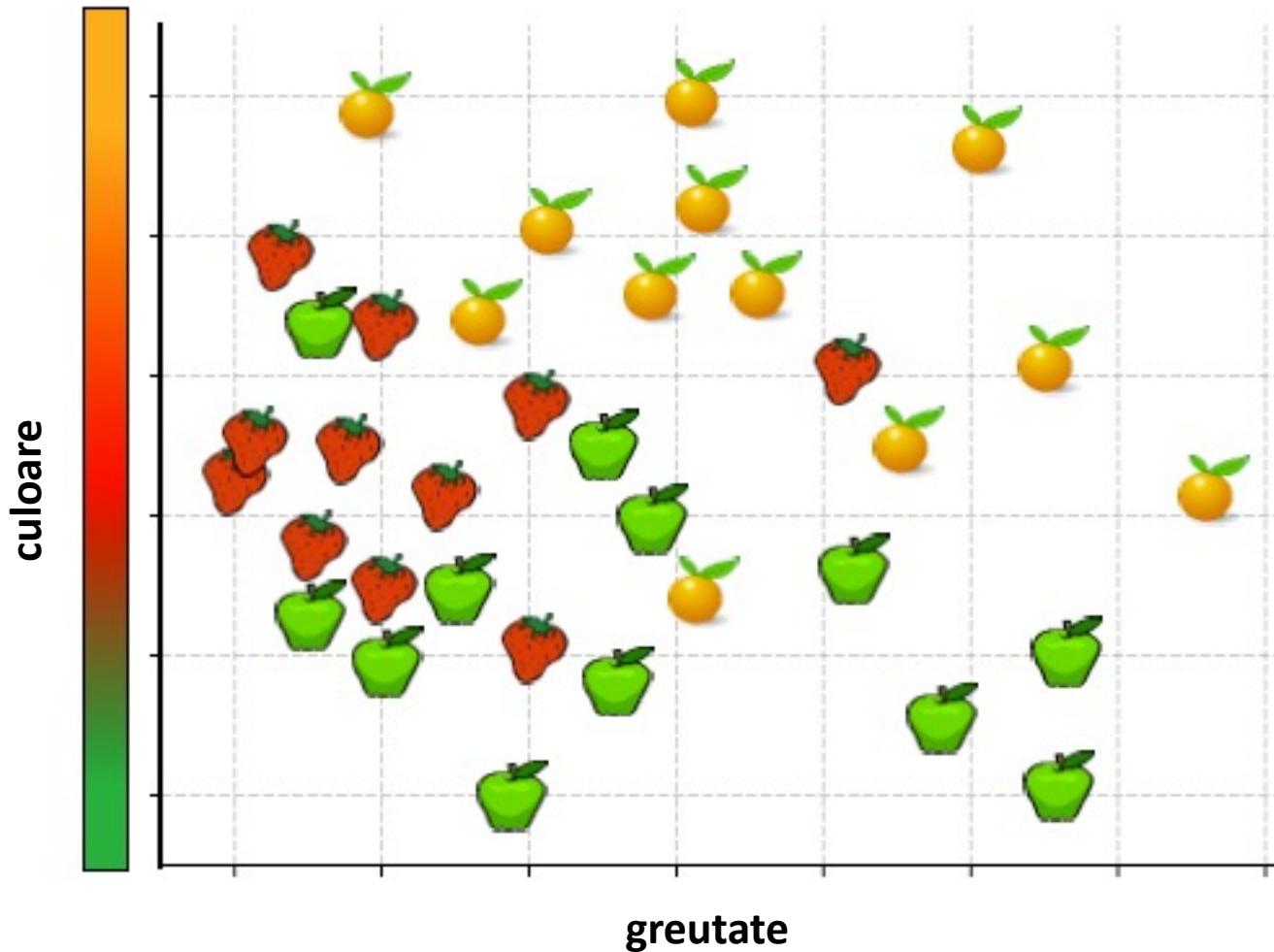
TN – true negative

# Modelul celor mai apropiati k-vecini (k-nearest neighbors)

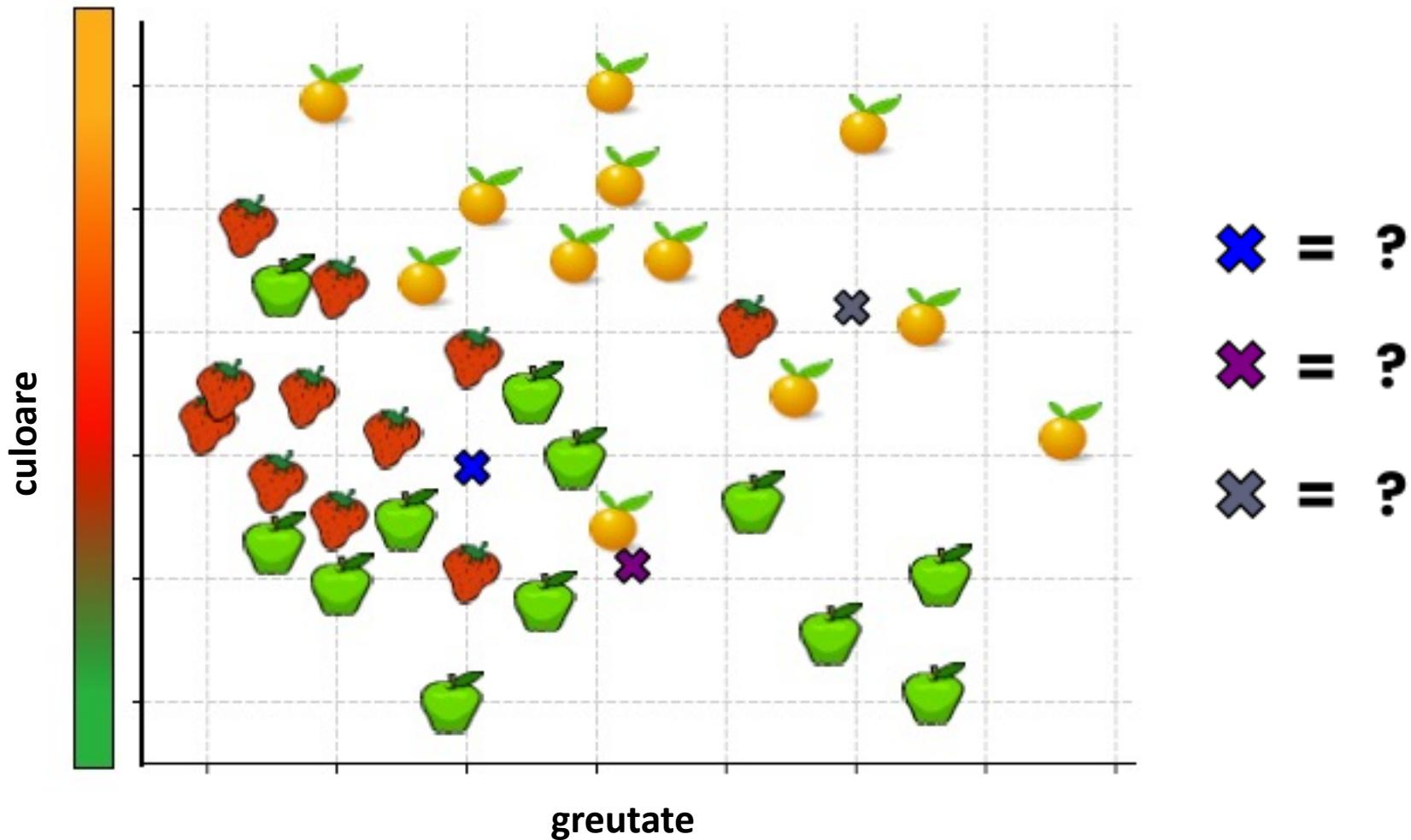
# Modelul celor mai apropiati k-vecini (kNN – k nearest neighbors)

- Modelul kNN prezice eticheta unui exemplu test ca fiind eticheta predominantă ale celor mai apropiate  $k$  exemple de antrenare (cei mai apropiati  $k$  vecini) din spațiul caracteristicilor (*feature space*).
- Învățare supervizată (avem etichete).
- Modelul kNN nu învață explicit un model:
  - în schimb, memorează datele de antrenare și le folosește în realizarea de predicții
- $k$  = numărul de vecini este hiperparametru (stabilit dinainte)

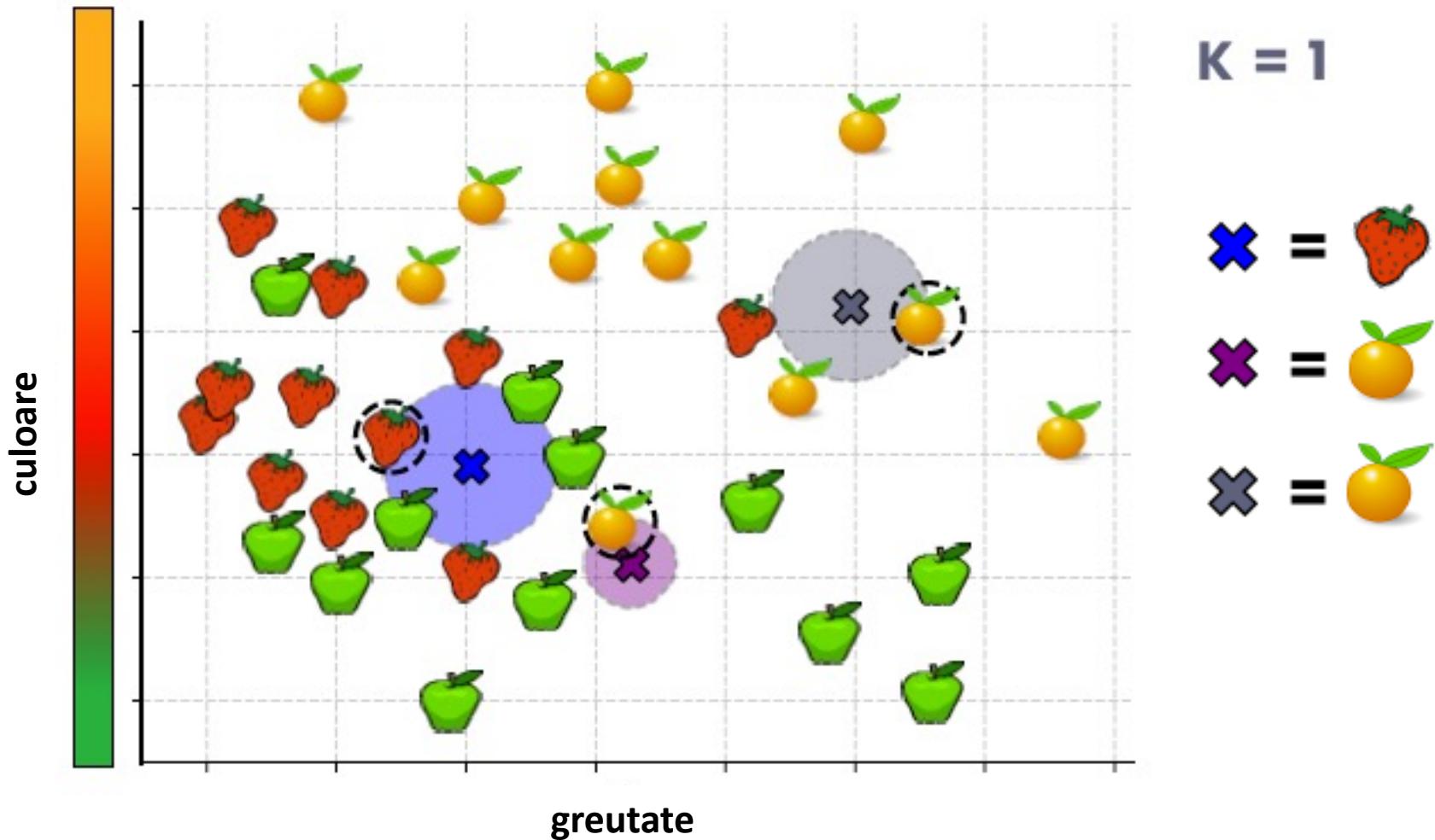
# Modelul celor mai apropiati k-vecini (kNN)



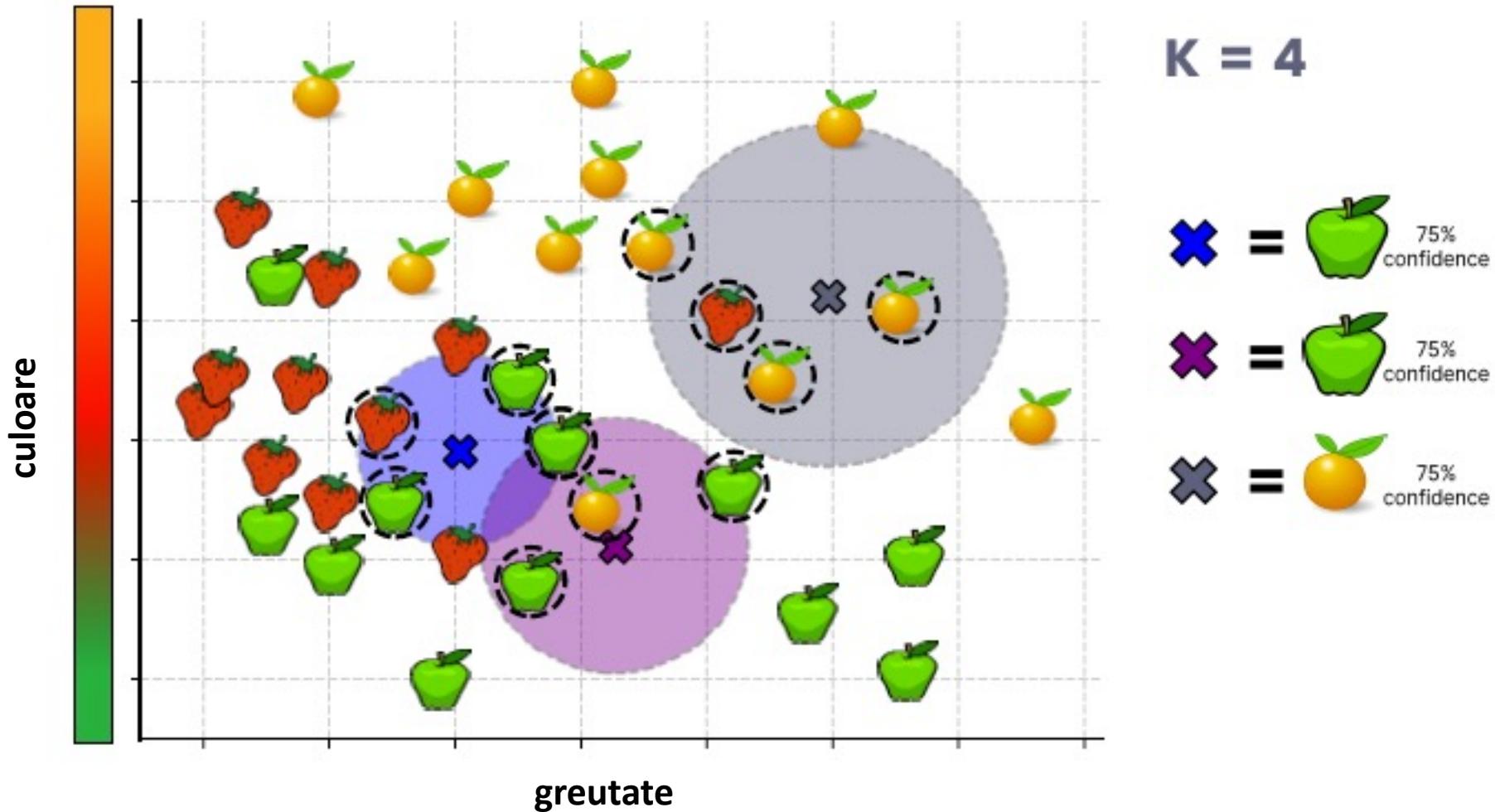
# Modelul celor mai apropiati k-vecini (kNN)



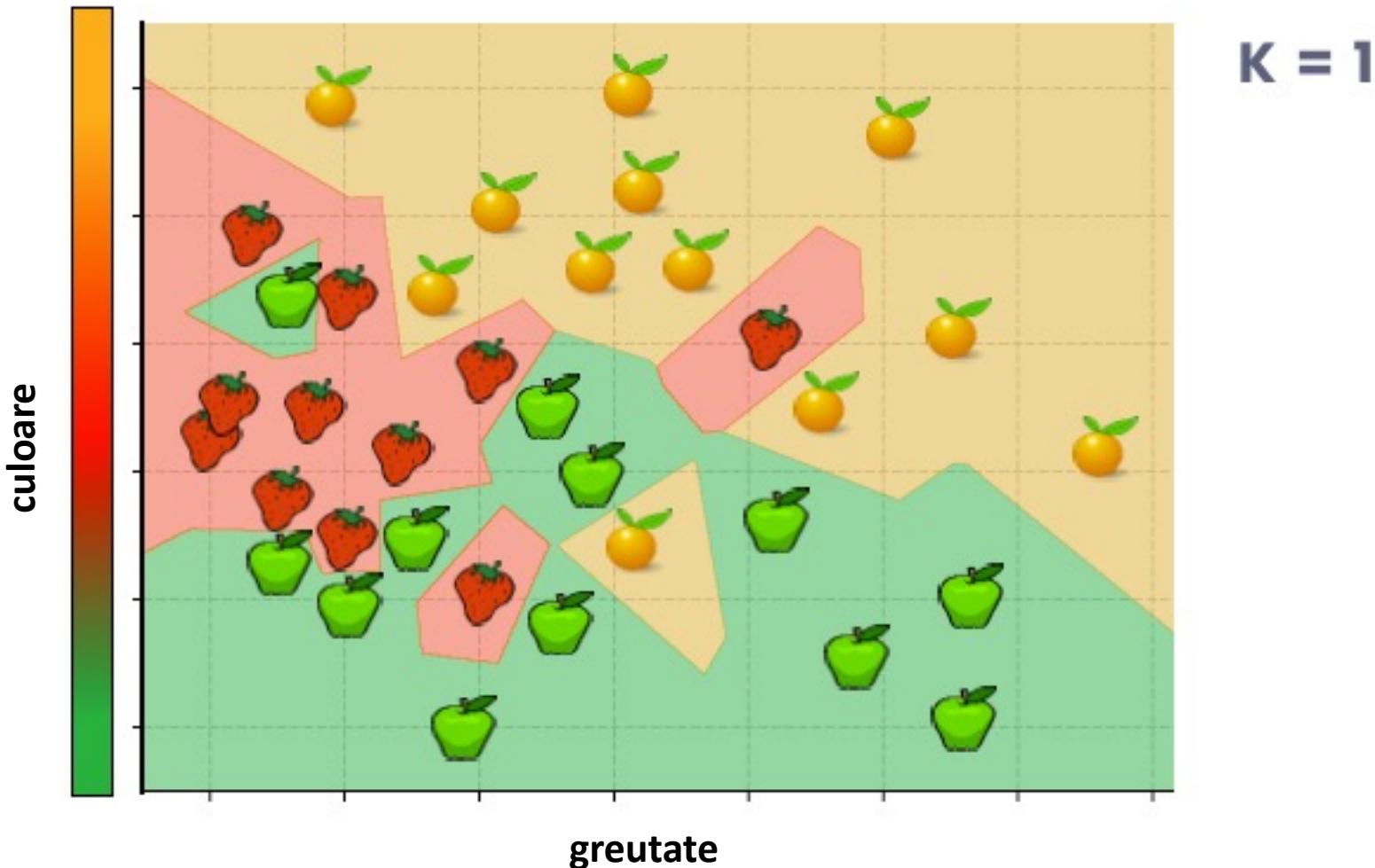
# Modelul celor mai apropiati k-vecini (kNN)



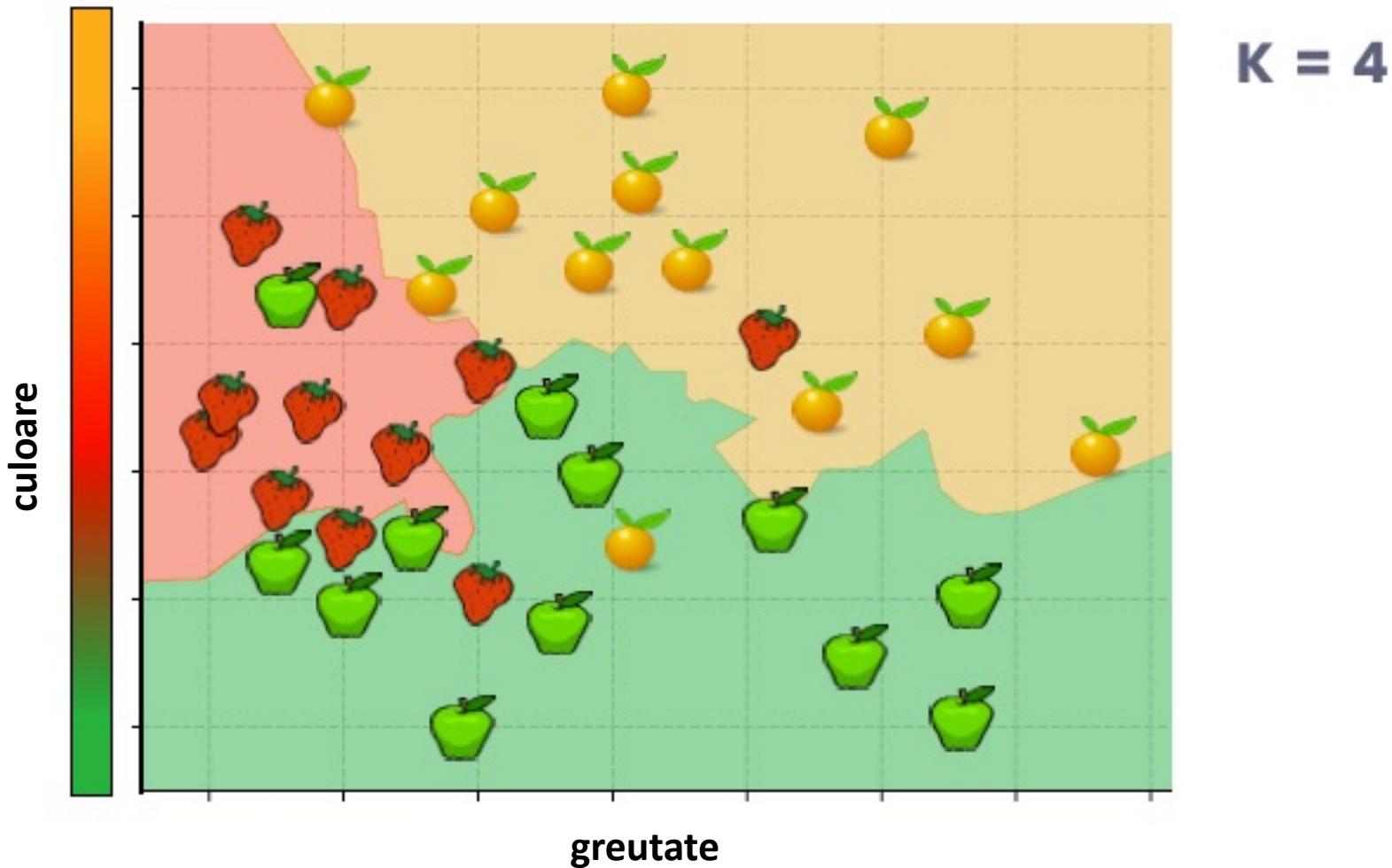
# Modelul celor mai apropiati k-vecini (kNN)



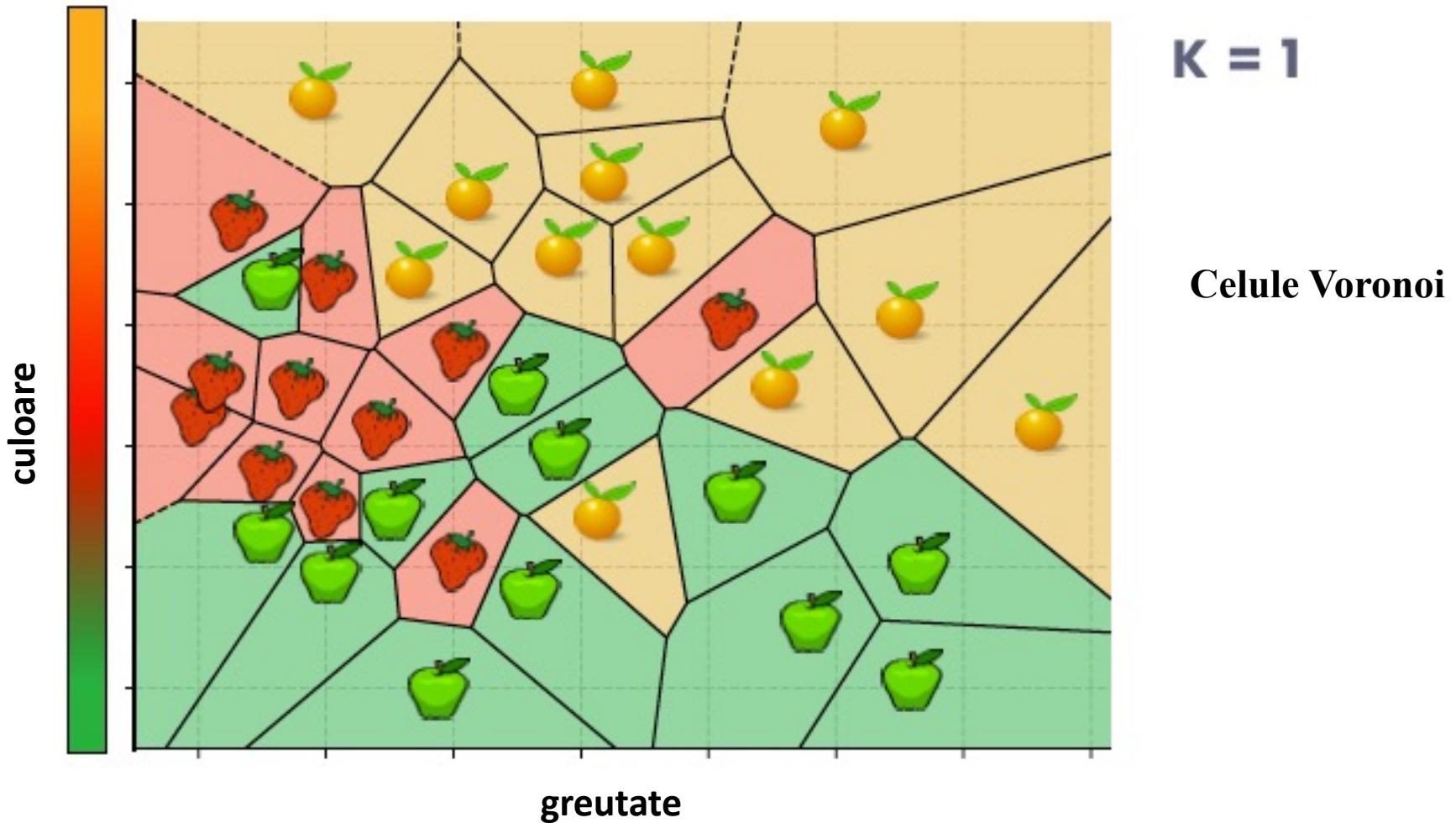
# Graniță de separare



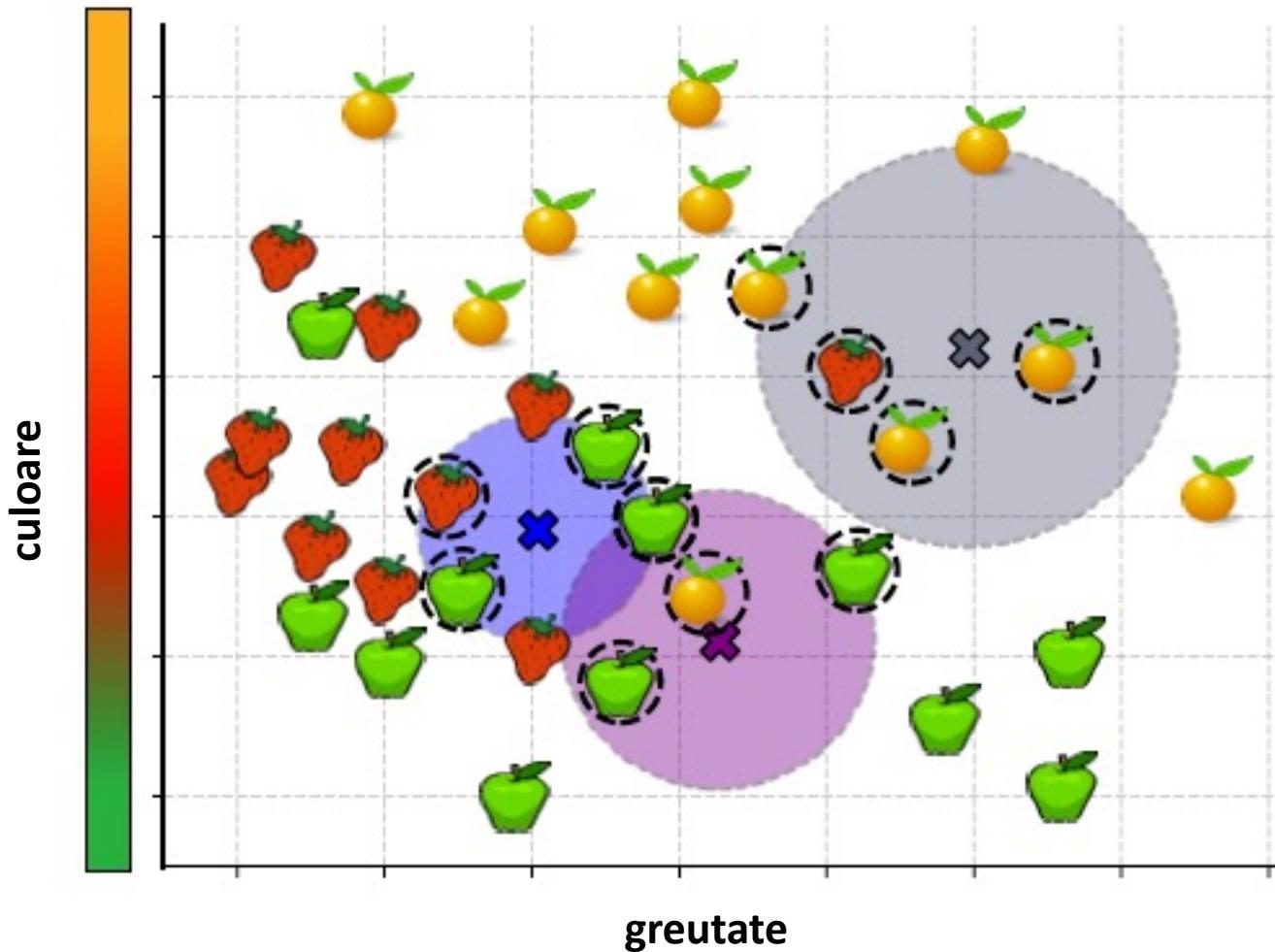
# Graniță de separare



# Graniță de separare



# Distanțe folosite de kNN

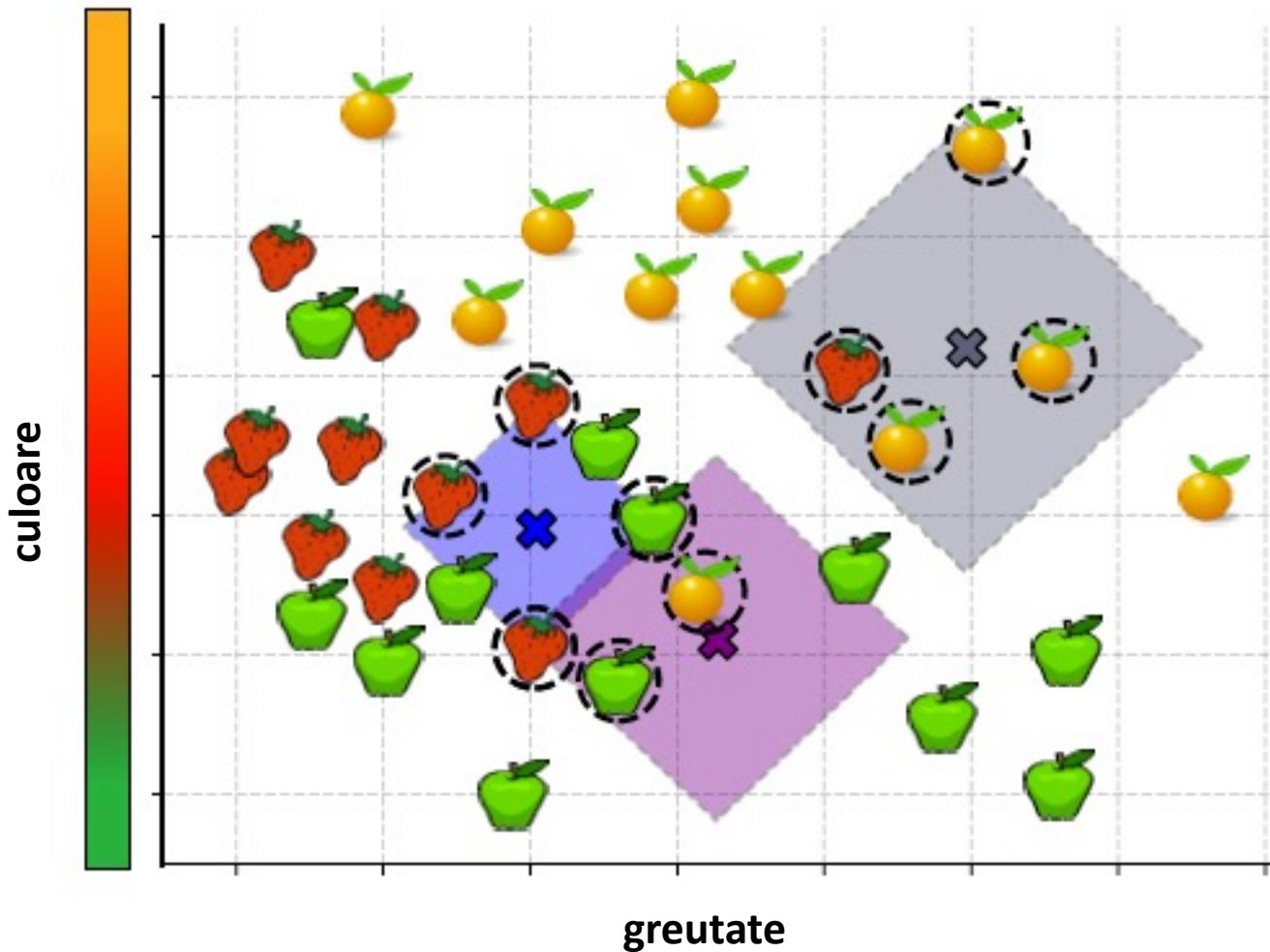


**K = 4**

- $\times$  (blue) = 75% confidence
- $\times$  (purple) = 75% confidence
- $\times$  (dark blue) = 75% confidence

**Distanță  
Euclidiană ( $L_2$ )**

# Distanțe folosite de kNN



$K = 4$

- $\times^{\text{blue}} = \text{strawberry}$  75% confidence
- $\times^{\text{purple}} = \text{apple}$  50% confidence
- $\times^{\text{dark blue}} = \text{orange}$  75% confidence

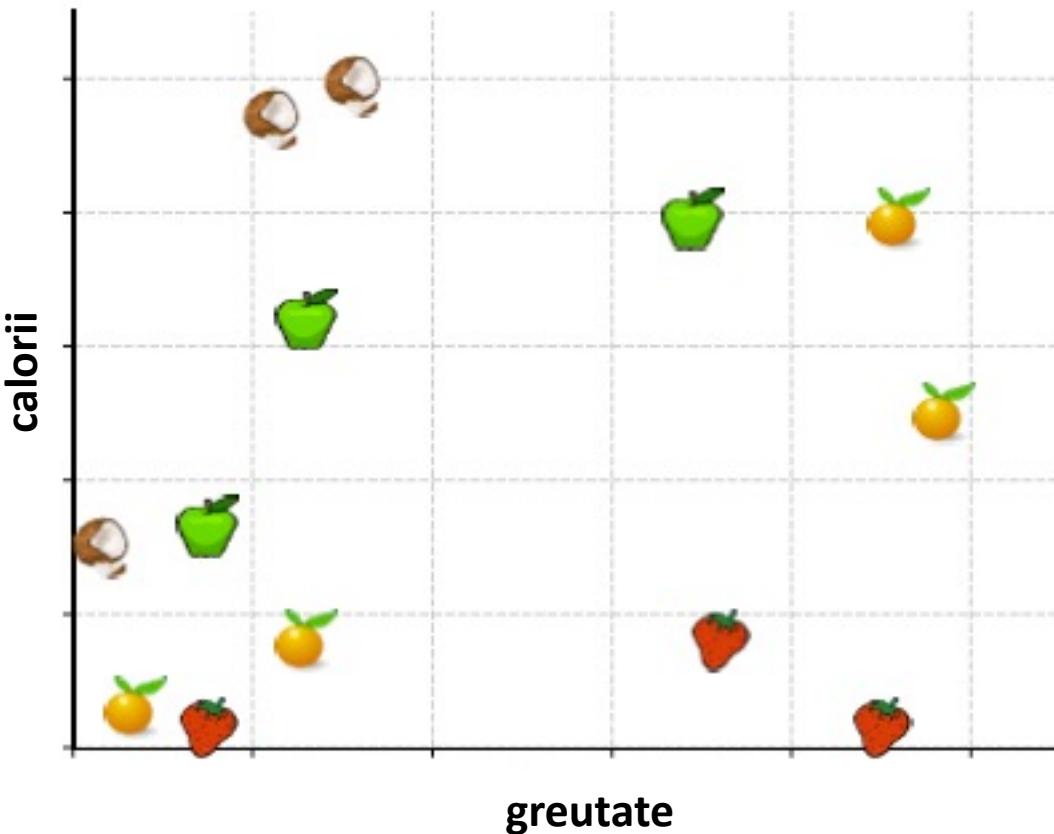
Distanță  
Manhattan ( $L_1$ )

# Distanță

- O funcție  $d: X \times X \rightarrow [0, \infty)$  se numește distanță, dacă satisface pentru orice două elemente  $x$  și  $y$  din  $X$  condițiile:
  1.  $d(x,y) \geq 0$
  2.  $d(x,y) = 0 \Leftrightarrow x = y$
  3.  $d(x,y) = d(y,x)$
  4.  $d(x,z) \leq d(x,y) + d(y,z)$
- Performanța modelului kNN depinde foarte mult de tipul distanței folosite în calculul celor mai apropiati vecini
  - distanța trebuie să reflecte natura problemei (ar trebui să ia valori mici pentru itemi cunoscuți ca similari și valori mari pentru itemi cunoscuți ca diferenți)

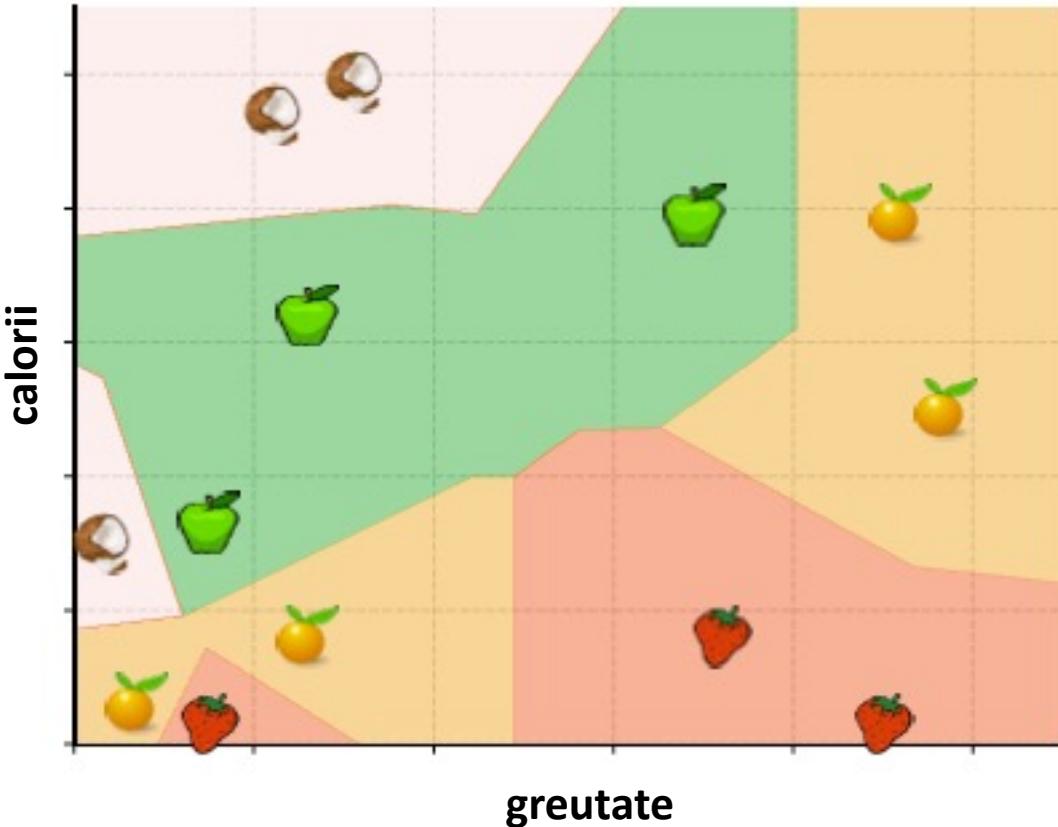
# Distanțe folosite de kNN

- Problemă de clasificare: prezicem tipul de fruct pe baza a două caracteristici, greutatea și cantitatea de calorii conținută



# Distanțe folosite de kNN

- Problemă de clasificare: prezicem tipul de fruct pe baza a două caracteristici, greutatea și cantitatea de calorii conținută

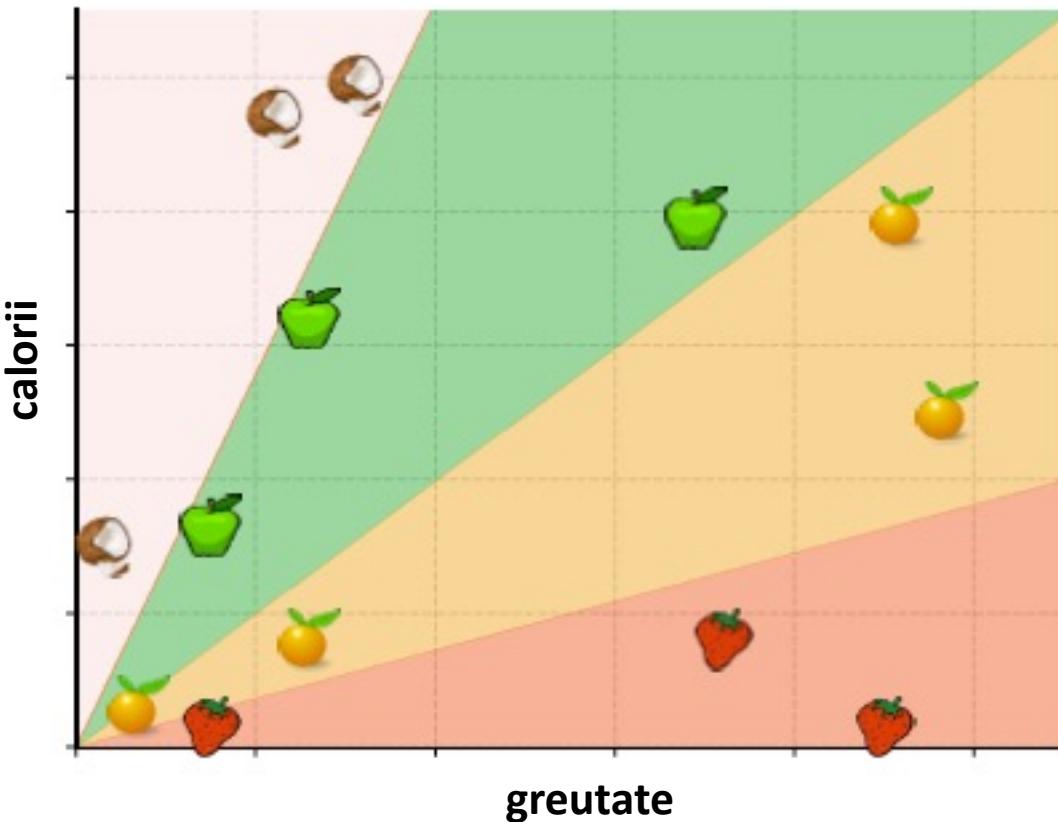


**Distanța  
Euclidiană ( $L_2$ )**

Distanța Euclidină modelează bine relația dintre caracteristici și etichete?

# Distanțe folosite de kNN

- Problemă de clasificare: prezicem tipul de fruct pe baza a două caracteristici, greutatea și cantitatea de calorii conținută



## Distanța Cosinus

Distanța Cosinus măsoară unghiul dintre vectorii de caracteristici.

Itemi similari au rapoarte similare între valorile caracteristicilor.

# Exemple de Distanțe

Euclidean distance:

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

$L_2$

Manhattan distance:

$$d(\vec{x}, \vec{y}) = \sum_i |x_i - y_i|$$

$L_1$

Chebyshev distance:

$$d(\vec{x}, \vec{y}) = \max_i (|x_i - y_i|)$$

$L_\infty$

Minkowski distance:

$$d(\vec{x}, \vec{y}) = \left( \sum_i |x_i - y_i|^p \right)^{1/p}$$

$L_p$

Cosine distance:

$$d(\vec{x}, \vec{y}) = 1 - \cos \theta = 1 - \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2} \sqrt{\sum_i y_i^2}}$$

# Distanța Euclidiană vs Manhattan

Euclidean distance:

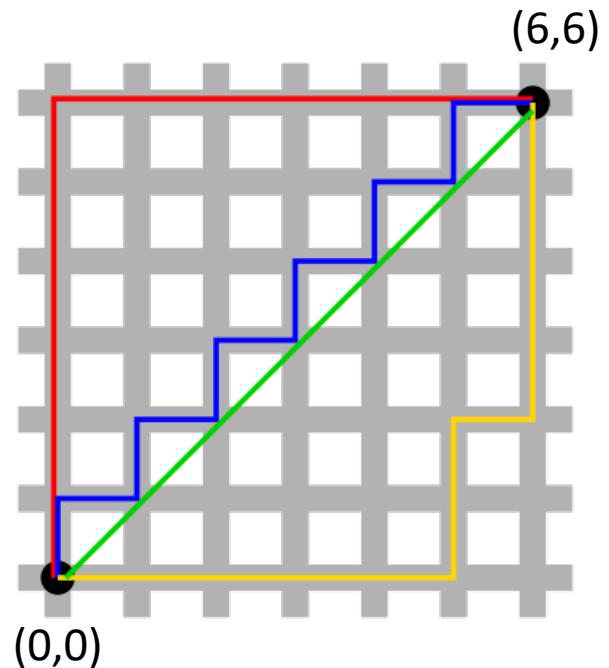
$$d(\vec{x}, \vec{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

distanța Euclidiană = lungimea drumului cel mai scurt = **lungimea drumului verde** =  $6\sqrt{2}$

Manhattan distance:

$$d(\vec{x}, \vec{y}) = \sum_i |x_i - y_i|$$

distanța Manhattan = suma absolută a diferențelor coordonateleor carteziene = **lungimea drumului roșu** = **lungimea drumului albastru** = **lungimea drumului galben** = 12



# Cuvinte cheie

Învățare automată

Etichetă

Caracteristică

Vector de caracteristici

Antrenare

Inferență

Model

Ipoteză

Algoritm

Supervizare

Clasificare

Regresie

Nesupervizare

Semi-supervizare

Acuratețe

Precizie

Recall

Matrice de confuzie