

# NF1 - INTRODUCCIÓ A L'APRENTATGE AUTOMÀTIC



*Autor: Generada ChatGPT*

*Curs d'Especialització en Intel·ligència Artificial i BigData*

*MP5072 Sistemes d'aprenentatge automàtic*

## Continguts

Aspectes Generals.....	4
Orígens i a on som.....	4
Intel·ligència Artificial .....	8
Tipus d'Intel·ligència Artificial .....	9
Intel·ligència Artificial i Aprenentatge Automàtic.....	12
Què és l'Aprenentatge Automàtic (Machine Learning)? .....	13
Com treballa el Machine Learning? .....	16
Tipus d'Aprenentatge Automàtic.....	17
Aprenentatge supervisat ( <i>Supervised Learning</i> ).....	19
Aprenentatge No Supervisat ( <i>Unsupervised Learning</i> ).....	21
Agrupament ( <i>Clustering</i> ).....	22
Reducció de dimensionalitat ( <i>Dimensionality Reduction</i> ) .....	26
Detecció d'anomalies.....	28
Aprenentatge per regles d'associació ( <i>Association Rule Mining</i> ).....	29
Aprenentatge per reforç (Reinforcement Learning).....	29
Principals algorismes de ML.....	31
Reptes del Machine Learning.....	32
Dades de poca qualitat .....	32
Dades d'entrenament no representatives.....	32
Dades desbalancejades.....	33
Dades d'entrenament insuficients ( <i>Underfitting</i> ).....	34
Overfitting Training Data (sobreajustament).....	35

Validació.....	38
MACHINE LEARNING.....	¡Error! Marcador no definido.
EJEMPLOS DE MACHINE LEARNING .....	¡Error! Marcador no definido.
SOFTWARE PARA ESTE MP.....	¡Error! Marcador no definido.
REFERÈNCIES.....	39

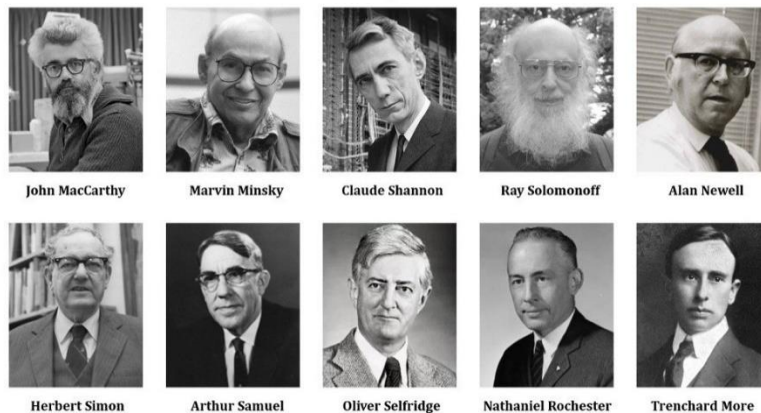
# Aspectes Generals

## Orígens i a on som

Encara que hem considerat que el tret de partida de intel·ligència artificial com a disciplina va ser el 1956, durant una conferència a Dartmouth College (Hanover, New Hampshire, Estats Units) sobre la informàtica teòrica, la humanitat ha tingut precedents per intentar desenvolupar màquines capces d'automatitzar processos. Aristòtil (348-322 a.C.) va estudiar la possibilitat de construir un sistema hidràulic que imités el comportament del cervell humà.

El segle XIII Ramon Llull va posar les bases de la computació i de la IA, més concretament l'any 1315 va publicar *Ars Magna*. En aquesta obra Llull planteja les seves primeres tesis respecte el raonament automàtic; és a dir, crear una màquina capaç de realitzar demostracions lògiques per validar o refutar teories. <https://www.youtube.com/watch?v=wmjuqVw7gh4>

### 1956 Dartmouth Conference: The Founding Fathers of AI



Medium.com

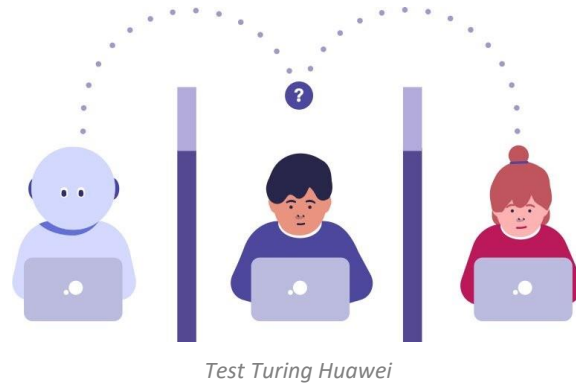
A la conferència de Dartmouth van assistir alguns dels científics que posteriorment es van encarregar de desenvolupar la disciplina en diferents àmbits i de dotar-la d'una estructura teòrica i computacional apropiada. Entre els assistents hi havia John McCarthy, Marvin Minsky, Allen Newell i Herbert Simon.

A. Newell i H. Simon van presentar un treball sobre demostració automàtica de teoremes que van denominar **Logic Theorist**. El *Logic Theorist* va ser el primer programa d'ordinador que emulava característiques pròpies del cervell humà, per la qual cosa **és considerat el primer sistema d'intel·ligència artificial de la història**. El sistema era capaç de demostrar gran part dels teoremes sobre lògica matemàtica que es presentaven en els tres volums dels *Principia Mathematica* (PM) d'Alfred N.

Whitehead i Bertrand Russell (1910-1913).

Minsky i McCarthy van fundar més tard el laboratori d'intel·ligència artificial del Massachusetts Institute of Technology (MIT), un dels grups pioners en l'àmbit.

El més rellevant des del punt de vista històric va ser proposat per **Alan Turing** en un article de 1950 publicat en la revista *Mind* titulat "*Computing Machinery and Intelligence*".



En aquest treball es proposa un test d'intel·ligència per a màquines segons el qual una màquina presentaria un comportament intel·ligent en la mesura en què fos capaç de mantenir una conversa amb un humà sense que una altra persona pugui distingir qui és l'humà i qui l'ordinador.

Encara que el test de Turing ha sofert innumbrables adaptacions, correccions i controvèrsies, posa de manifest els primers intents d'assolir una definició objectiva de la intel·ligència.

En aquest context, és d'especial rellevància el teorema d'incompleitud de Gödel de 1931, un conjunt de teoremes de lògica matemàtica que estableixen les limitacions inherents a un sistema basat en regles i procediments lògics (com són tots els sistemes d'IA).

Després dels primers treballs en IA dels anys cinquanta, en la dècada dels seixanta es va produir un gran esforç de formalització matemàtica dels mètodes utilitzats pels sistemes d'IA.

Articles referents al test de Turing:

- [Blog de Huawei](#)
- [Web Cheana](#)
- [LaMDA](#) Language Model for Dialog Applications

Els anys setanta, en part com a resposta al test de Turing, es va produir el naixement d'una àrea coneguda com a processament del llenguatge natural (NLP, natural language processing), una disciplina dedicada a

sistemes artificials capaços de generar frases intel·ligents i de mantenir converses amb humans.

L’NLP ha donat lloc a diverses àrees d’investigació en el camp de la lingüística computacional, incloent-hi aspectes com la desambiguació semàntica o la comunicació amb dades incompletes o errònies. Malgrat els grans avanços en aquest àmbit, continua sense haver-hi una màquina que pugui passar el test de Turing tal com es va plantejar en l’article original. Això no és tant a causa d’un fracàs de la IA com al fet que els interessos de l’àrea s’han anat redefinint al llarg de la història.

El 1990, el controvertit empresari Hugh Loebner i el Cambridge Center for Behavioral Studies van instaurar el premi Loebner, un concurs anual certament heterodox en el qual es premia el sistema artificial que mantingui una conversa més indistingible de la d’un humà. Avui dia, la comunitat científica considera que la intel·ligència artificial s’ha d’enfocar des d’una perspectiva diferent a la que es tenia en els anys cinquanta, però iniciatives com la de Loebner expressen l’impacte sociològic que continua tenint la IA en la societat actual.

Als anys vuitanta (80’s) es van començar a desenvolupar les primeres aplicacions comercials de la IA, fonamentalment dirigides a problemes de producció, control de processos o comptabilitat. Amb aquestes aplicacions van aparèixer els primers sistemes experts, que permetien fer tasques de diagnòstic i presa de decisions a partir d’informació aportada per professionals experts.

Entorn de 1990, IBM va construir l’ordinador d’escacs Deep Blue, capaç de plantar cara a un gran mestre d’escacs utilitzant algorismes de cerca i anàlisi que li permetien valorar centenars de milers de posicions per segon. Més enllà de l’intent de dissenyar robots humanoides i sistemes que rivalitzin amb el cervell humà en funcionalitat i rendiment, l’interès avui dia és dissenyar i implementar sistemes que permetin analitzar grans quantitats de dades de manera ràpida i eficient. Actualment, cada persona genera i rep diàriament una gran quantitat d’informació no solament per mitjà dels canals clàssics (conversa, carta, televisió) sinó mitjançant nous mitjans que ens permeten contactar amb més persones i transmetre més dades en les comunicacions (Internet, fotografia digital, telefonia mòbil). Encara que el cervell humà és capaç de reconèixer patrons i establir relacions útils entre aquests de manera excepcionalment eficaç, és certament limitat quan la quantitat de dades resulta excessiva. Un fenomen similar ocorre en l’àmbit empresarial, en què cada dia és més necessari barrejar quantitats ingents d’informació per a poder prendre decisions. L’aplicació de tècniques d’IA als negocis ha donat lloc a àmbits d’implantació recent com la intel·ligència empresarial, business intelligence, o a la mineria de dades, data mining. En efecte, avui més que mai la informació està codificada en masses ingents de dades, de manera que en molts àmbits es fa necessari extreure la informació rellevant de grans conjunts de dades abans de procedir a una anàlisi detallada. Regressió logística múltiple: S’intenta predir amb més d’una variable independent.

En resum, avui dia un dels objectius principals de la intel·ligència actual és el tractament i anàlisi de dades.

## El Hype Cycle para las tecnologías emergentes de 2024



Fuente: Gartner  
La reutilización comercial de este recurso está sujeta a la aprobación de Gartner y debe cumplir con la Política de Cumplimiento de Contenido de Gartner, disponible en gartner.es.  
© 2024 Gartner, Inc. o sus filiales. Todos los derechos reservados. 3205434

**Gartner**

<https://www.gartner.es/es/articulos/hype-cycle-para-las-tecnologias-emergentes>

### Què és el HypeCycle per les tecnologies emergents de Gartner?

El Hype Cycle de Gartner per a les tecnologies emergents, un dels informes més influents i llegits entre els clients de Gartner, proporciona una visió completa de les tecnologies més rellevants i transformadores del futur. A més, ofereix un marc per seguir i predir l'impacte i la trajectòria d'aquestes tecnologies.

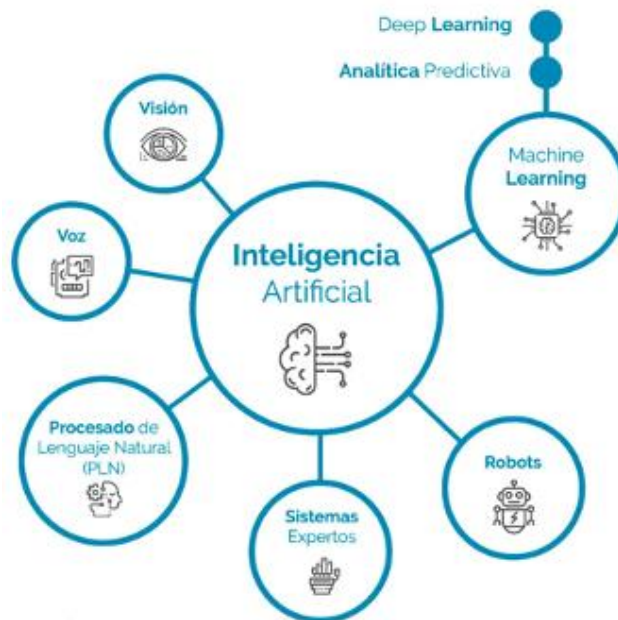


# Intel·ligència Artificial

Definir el concepte d'Intel·ligència Artificial no és fàcil, però una definició generalista seria la disciplina que estudia la manera de que les màquines pensin com els humans o dit d'una altra manera exhibeixin intel·ligència humana.

Els objectius principals de la IA inclouen:

- La deducció i el raonament
- La representació del coneixement
- La planificació
- El processament del llenguatge natural (NLP)
- L'aprenentatge
- La percepció
- La capacitat de manipular i moure objectes





## Tipus d'Intel·ligència Artificial

La fita més important de la intel·ligència artificial (IA) és aconseguir que una màquina tingui una intel·ligència de tipus general similar a la humana.

És important el matís “**de tipus general**” i no específica perquè la intel·ligència dels éssers humans és de tipus general.

Per exemple, els programes que juguen als escacs al nivell de grans mestres són incapaços de jugar a les dames. Per jugar a les dames es requereix d'un programa diferent. És a dir, el programa que juga els escacs no pot aprofitar el fet que juga als escacs per adaptar-se i jugar també a les dames.

En el cas dels éssers humans, qualsevol jugador d'escacs pot aprofitar els seus coneixements sobre aquest joc per a jugar a les dames perfectament.

El 1980 el filòsof John Searle en un article crític amb la intel·ligència artificial va introduir la distinció entre IA feble i forta. Aquest article va provocar, i continua provocant, molta polèmica.

### IA feble (Artificial Narrow Intelligence)

Es defineix com la intel·ligència artificial que es centre en una tasca específica. Aquesta és limitada i no té autoconsciència o intel·ligència genuïna.

La IA feble només pretén ser aplicada a un tipus específic de problemes, per exemple jugar els escacs.

En aquest àmbit la IA feble ha superat en molts casos l'ésser humà i s'ha demostrat àmpliament en certs dominis, com ara buscar solucions a fórmules lògiques amb moltes variables i altres aspectes relacionats amb la presa de decisions.

Siri és un clar exemple de IA feble que combina diferents tècniques de IA feble per funcionar. Siri pot fer moltes coses, però a mesura que intentem tenir conversacions amb l'assistent de veu, ens adonem de les seves limitacions.

### IA forta (Artificial General Intelligence)

La IA forta implicaria que un ordinador convenientment programat no simula una ment sinó que “és una ment” i per tant hauria de ser capaç de pensar/realitzar totes les tasques igual que un ésser humà.

Hauria de ser capaç de fer-se preguntes a ella mateixa .

Exemples de la IA forta la trobem en les pel·lícules de ciència ficció:


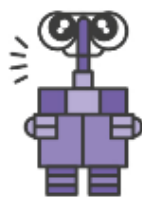
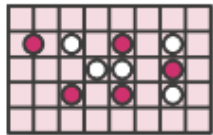




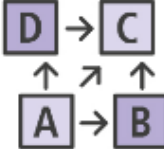
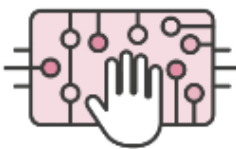
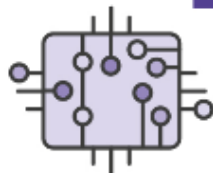
- [Ex Machina, 2015](#)

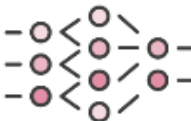
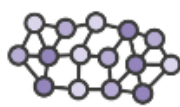


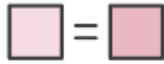

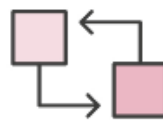
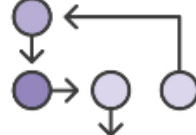

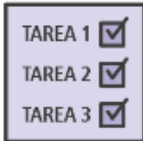
- [Her, 2014](#)
- J.A.R.V.I.S. (Just A Rather Very Intelligent System), el majordom de Iron Man.

INTEL·LIGÈNCIA ARTIFICIAL FEBLE	INTEL·LIGÈNCIA ARTIFICIAL FORTA
Existeix a la vida real	No existeix a la vida real
Orientats a problemes molt específics	Resolen problemes oberts
Reactiu	Proactiu
És programat per un humà	S'autoprogramen
No raonen, només computen/executen	Imiten el comportament humà
Aprenen a base d'exemples similars	Aprenen com les persones
No podran reemplaçar l'ésser humà	Similars a les capacitats humanes

Podem dir que tots els avenços aconseguits fins ara en el camp de la IA són manifestacions de la IA feble i específica.

A la següent infografia elaborada per Huawei es poden veure les diferències entre forta i feble

INTELIGENCIA ARTIFICIAL DÉBIL		INTELIGENCIA ARTIFICIAL FUERTE	
IAD			IAF
	VS		
EXISTEN EN LA VIDA REAL AlphaGo, Watson o Sophia.		SOLO EN LA CIENCIA FICCIÓN T-800, Sony, Wall-E o J.A.R.V.I.S.	
IAD			IAF
	VS		
ORIENTADOS A PROBLEMAS MUY CONCRETOS Juego muy bien al Go.		RESUELVEN PROBLEMAS ABIERTOS Un poco de todo: viaje temporal, matar a John Connor, etc.	
IAD			IAF
	VS		
REACTIVO Esperaré a que empieces a jugar.		PROACTIVO Necesito tu ropa, tus botas y tu motocicleta.	
APRENDEN			
IAD			IAF
	VS		
SIN FLEXIBILIDAD Solo al Go, no me lies.		SON FLEXIBLES Correr es como andar, ¡pero más rápido!	
IAD			IAF
	VS		
PROGRAMA UN HUMANO Dime qué tengo que pensar.		SE AUTOPROGRAMAN Corriendo, aprendo sobre mis límites.	

IAD			IAF
	VS		
POCAS REDES NEURONALES (p → q).		MUCHAS REDES NEURONALES, A VECES EN CONFLICTO Decido según mi programación.	
IAD			IAF
	VS		
NO RAZONAN, SOLO COMPUTAN Juego bien, pero inconscientemente.		IMITAN EL COMPORTAMIENTO HUMANO Pienso, luego existo.	
IAD			IAF
	VS		
APRENDEN DE EJEMPLOS SIMILARES Presto atención a las fichas, pero no te escucho.		APRENDEN COMO LAS PERSONAS El ajedrez se parece a las damas.	
¿ME QUITARÁN EL TRABAJO?			
No pueden reemplazar a un ser humano		Similares a las capacidades humanas	
IAD			IAF
	VS		
TAREAS REPETITIVAS ¿Otra partida?		APRENDEN NUEVAS TAREAS Puedo programar.	
No pueden reemplazar a un ser humano		Similares a las capacidades humanas	
IAD			IAF
	VS		
NO PUEDEN SALIRSE DE SU MARCO DE TRABAJO Del juego Go no me saques.		ADAPTABILIDAD A NUEVOS ESCENARIOS Connor eliminado, jefe, ¿qué toca?	

# Intel·ligència Artificial i Aprenentatge Automàtic

Moltes vegades els termes d'intel·ligència artificial (IA), aprenentatge automàtic (machine learning) i aprenentatge profund (deep learning) s'utilitzen de manera intercanviable o es confonen.

**Intel·ligència artificial:** Volem dotar a les “màquines” la possibilitat que aprenguin i raonin com els éssers humans.

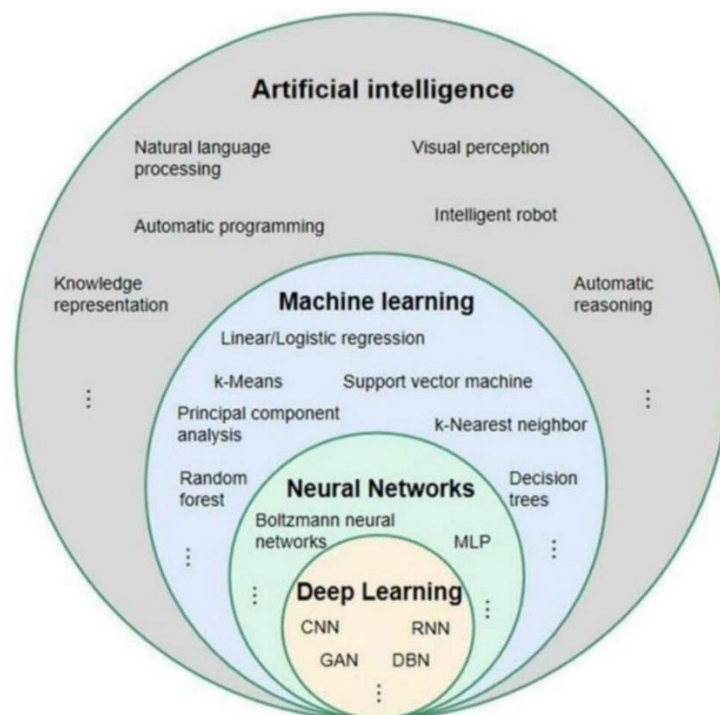
**Aprenentatge automàtic:** Subcamp de la intel·ligència artificial que proporciona als sistemes la capacitat de reconèixer patrons i prendre sense ser programats explícitament. Es centre en l'ús de les dades i algorismes per entrenar models d'aprenentatge perquè aquests puguin realitzar prediccions.

**Aprenentatge profund:** Subcamp de l'aprenentatge automàtic el qual les xarxes neuronals artificials s'adapten i aprenent mitjançant grans quantitats de dades.

En un exemple podria dir que l'aprenentatge automàtic (ML) captura moltes imatges i una persona indica si la imatge és bona o no, o sigui, captar les característiques i realitzar-ne una classificació. En canvi l'aprenentatge profund (DL) aprèn automàticament.

En el següent gràfic es pot veure quina relació hi ha entre aquests termes i alguns dels termes més significatius de cadascun d'ells.

Vídeo resum IA: <https://www.youtube.com/watch?v=oV74Najm6Nc>



# Què és l'Aprenentatge Automàtic (Machine Learning)?

A partir d'aquest punt, parlarem d'aprenentatge automàtic utilitzant també el terme en anglès Machine Learning (ML). Farem servir ambdós termes com a equivalents.

Abans d'intentar definir formalment el concepte d'aprenentatge automàtic, fem-nos abans una altra pregunta. Com aprenem els humans? Per exemple, no distingim un animal d'un altre per la seva definició sinó molt sovint ho fem mitjançant exemples i/o mostres diferents.

Quan ensenyem en els nens a distingir una zebra no ho fem pas en base a la seva definició formal, sinó que normalment ho fem a base de mostrar exemples.

## Definició de zebra:

*Nom donat a diversos mamífers perissodàctils de la família dels èquids que pertanyen als gèneres Equus, Dolichohippus i Hippotigris (considerats també a vegades com a subgèneres d'Equus), caracteritzats pel pelatge clar amb franges transversals fosques a tot el cos, el coll gruixut i el cap gros i pesant. Les dues espècies més importants són E. (Dolichohippus) grevyi i E. (Hippotigris) zebra*



A través dels diferents exemples el cervell del nen va detectant quines diferències hi ha entre una zebra i un animal que no ho és, va agafant **quines són les característiques principals** de la zebra per determinar quan aquesta ho és o no. Amb l'aprenentatge automàtic passa una mica el mateix.

Definicions:

Són algorismes capaços d'identificar i aprendre patrons dins de les dades per realitzar prediccions.

L'aprenentatge automàtic és un subconjunt de la intel·ligència artificial que proporciona als sistemes/màquines la capacitat d'aprendre automàticament i millorar a partir de l'experiència sense ser programats explícitament.

L'aprenentatge automàtic és la ciència (i l'art) de programar ordinadors perquè puguin aprendre de les dades.

L'aprenentatge automàtic és el camp d'estudi que proporciona als ordinadors la capacitat d'aprendre sense estar programat explícitament. *Arthur Samuel, 1959*



*Arthur Samuel* no era un bon jugador de les dames i va programar milers de partides contra ell mateix i veure quines posicions en el taulell tendien a conduir a victòries i quines a derrotes. El programa prenia quines eren les bones posicions en el taulell i quines dolentes. Finalment, aprenia a jugar a les dames millor que el propi Arthur Samuel.

Es diu que un programa informàtic aprèn de l'experiència **E** respecte a alguna tasca **T** i alguna mesura de rendiment **P**, si el seu rendiment en **T**, mesurat per **P**, millora amb experiència **E**. *Tom Mitchell, 1997*

A l'exemple de les dames l'experiència **E** seria l'experiència del programa jugant desenes de milers de partides contra ell mateix. La tasca **T** seria la tasca de jugar a les dames i la mesura/indicar del rendiment **P** seria la probabilitat de guanyar la pròxima partida contra algun oponent nou.

**Exercici / exemple:** Donat un client de correu electrònic amb un botó de Spam per marcar o no un correu com spam i basant-se en els correus marcats com spam el programa aprèn a filtrar millor altres correus spam.

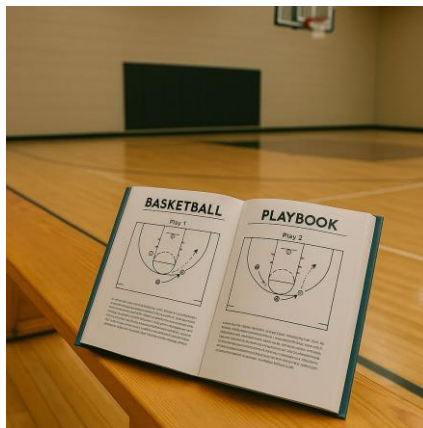
Quina és la tasca **T**, l'experiència **E** i el rendiment **P** en aquest exemple?

- a) Classificar els emails com spam o no spam
- b) Mirar els teus emails etiquetats com spam i quins no
- c) El nombre o fracció dels emails correctament classificats com spam/no spam
- d) Cap de les anterior. Aquest no és un problema d'aprenentatge automàtic

No hem de veure l'aprenentatge automàtic com la programació tradicional a on apliquem un conjunt d'instruccions de manera seqüencial, sinó que aprenem a base d'exemples.

## PROGRAMACIÓ TRADICIONAL

## MACHINE LEARNING



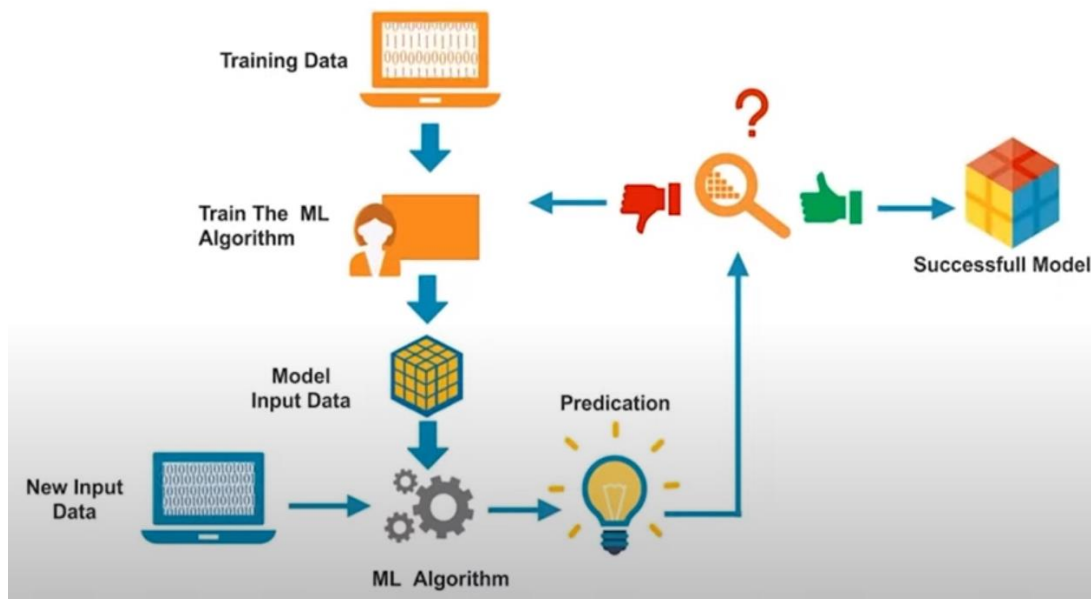
Exemple d'aprenentatge automàtic a CODE.org



<https://www.code.org/oceans>

<https://studio.code.org/courses/oceans/units/1/lessons/1/levels/2>

## Com treballa el Machine Learning?

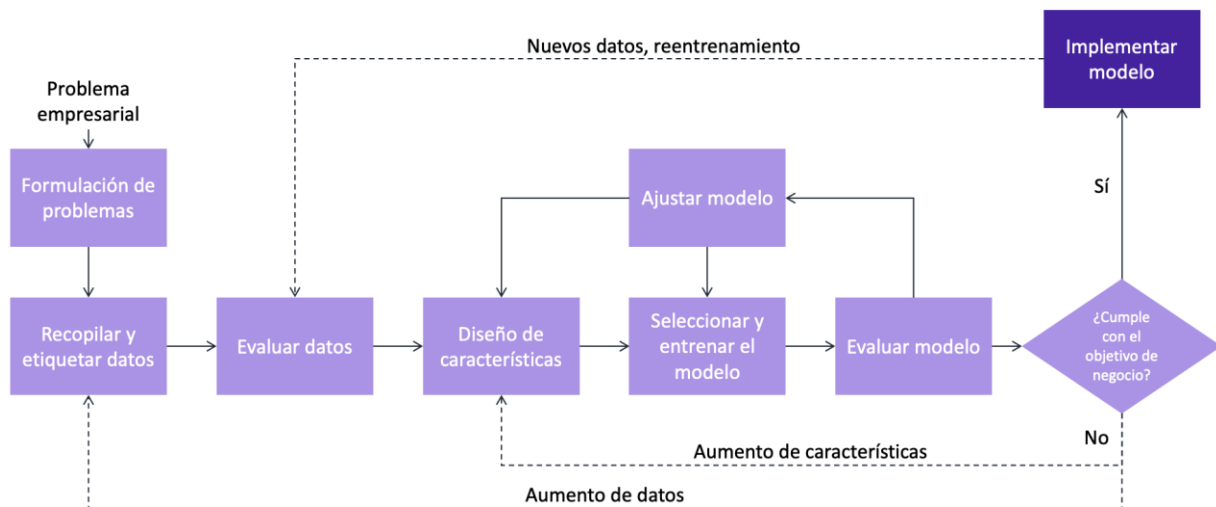


El funcionament general en un procés de Machine Learning és el següent. Ens basem en una sèrie de dades. Aquestes dades han passat per una fase prèvia de preparació i processat. En aquesta fase prèvia s'extrauran les dades d'un o diferents orígens, s'analitzaran estadísticament per determinar si les dades són consistents i es poden utilitzar per una aprenentatge automàtic.

Aquestes dades es passen a l'algorisme de Machine Learning per obtenir un model. Un cop tenim el model utilitzem un altre conjunt de dades per testear el model per veure si prediu bé o no.

La predicció s'avalua per a la precisió i si aquesta és acceptable, s'implementa o es dona per bo el model; altrament s'entrena una i altra vegada augmentant el conjunt de dades d'entrenament fins obtenir un model que tingui la precisió desitjable.





Hi ha una regla no escrita que diu que el 80% del temps es dedica a l'exploració, preparació i neteja de dades abans d'utilitzar-les com a dades d'entrenament i el 20% del temps es dedica a executar el model, avaluar-lo i refinar-lo.

També hi ha la regla del 80/20 en Machine Learning, que consisteix a dividir les dades en un 80% per entrenament i un 20% per validació/test. Aquesta divisió és una pràctica habitual i consensuada per la comunitat de ciència de dades i Machine Learning. Aquesta guia pràctica funciona bé en la majoria de casos i va sorgir com a una recomanació empírica que equilibrava dues necessitats: Tenir prou dades per entrenar bé el model i tenir suficients dades per validar-lo amb fiabilitat.

## Tipus d'Aprenentatge Automàtic

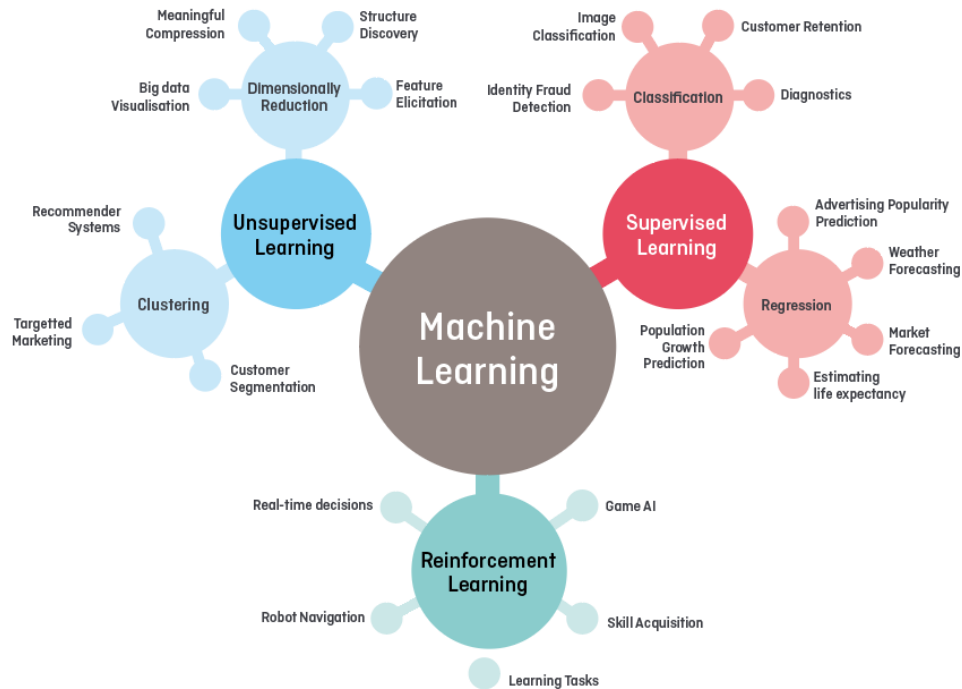
Existeixen molts tipus diferents de sistemes d'aprenentatge automàtic (Machine Learning), i per entendre'ls millor és útil classificar-los segons diversos criteris. A continuació, presentem les principals formes de classificació:

### a) Segons el tipus de supervisió durant l'entrenament

Aquesta és la classificació més habitual, i es basa en la quantitat i el tipus d'informació que el sistema rep durant l'aprenentatge.

- **Aprenentatge supervisat:** el sistema rep exemples d'entrada amb la seva resposta correcta (etiqueta) i aprèn a predir aquestes etiquetes.
- **Aprenentatge no supervisat:** no hi ha etiquetes; el sistema ha de trobar estructures ocultes dins les dades (com agrupaments o patrons).

- **Aprenentatge semi-supervisat:** es combina una petita quantitat de dades etiquetades amb una gran quantitat de dades sense etiquetar.
- **Aprenentatge auto-supervisat:** el mateix model genera etiquetes aparti de dades no etiquetades.
- **Aprenentatge per reforç (Reinforcement Learning):** l'agent aprèn a prendre decisions mitjançant proves, errors i recompenses.



Com s'ha dit la forma més habitual de classificació és segons el tipus de supervisió, però podem trobar altres formes de classificació.

### b) Segons la forma d'aprenentatge al llarg del temps

Aquí ens fixem en com aprèn el model amb el temps i com gestiona les dades:

- **Aprenentatge per lots (batch learning):** el sistema aprèn un cop amb tot el conjunt de dades, i si volem millorar-lo cal tornar-lo a entrenar des de zero.
- **Aprenentatge en línia (online learning):** el sistema aprèn progressivament, a mesura que arriben noves dades. Ideal per a sistemes en temps real o que treballen amb grans volums de dades.

### c) Segons l'estratègia d'aprenentatge

Aquest criteri diferencia com el sistema fa les seves prediccions:

- **Aprentatge basat en instàncies:** el sistema memoritza exemples i compara les noves dades amb els exemples coneguts. Exemple: k-NN (k-Nearest Neighbors).
- **Aprentatge basat en models:** el sistema construeix un model general a partir dels exemples, buscant patrons que expliquin les dades. Exemple: regressió lineal, xarxes neuronals, etc.

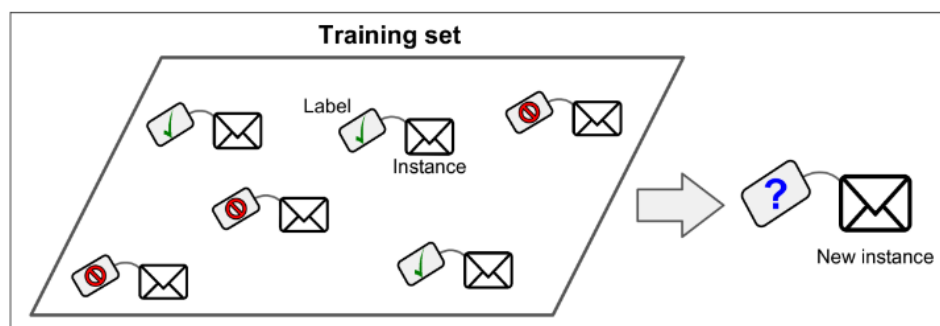
En molts casos, un model concret pot combinar diversos d'aquests aspectes. Per exemple, un sistema pot ser supervisat, basat en models i amb aprenentatge per lots.

## Aprentatge supervisat (*Supervised Learning*)

En l'aprenentatge supervisat, les dades d'entrenament que introduïm a l'algorisme inclouen les solucions, anomenades etiquetes (*labels*).

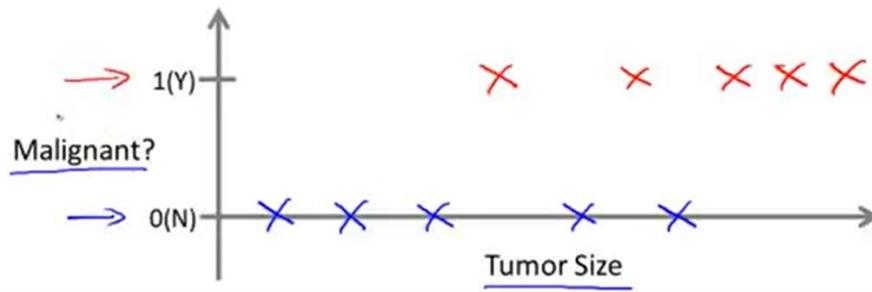
Si l'aprenentatge supervisat el descrivim matemàticament és quan tenim una variable d'entrada  $X$  i una de sortida  $Y$  i utilitzem un algorisme per trobar la relació entre aquestes dues variables de tal manera que tenim una funció  $f(X) = Y$ . El nostre objectiu és apropar la funció de mapeig tan com sigui possible de tal manera que podem predir qualsevol  $Y$  per qualsevol  $X$  donada.

Una tasca típica d'aprenentatge supervisat és la classificació. El filtre de correu brossa és un bon exemple d'això: s'entrena amb molts correus electrònics d'exemple juntament amb la seva classe (spam o no spam), i ha d'aprendre a classificar nous correus electrònics.

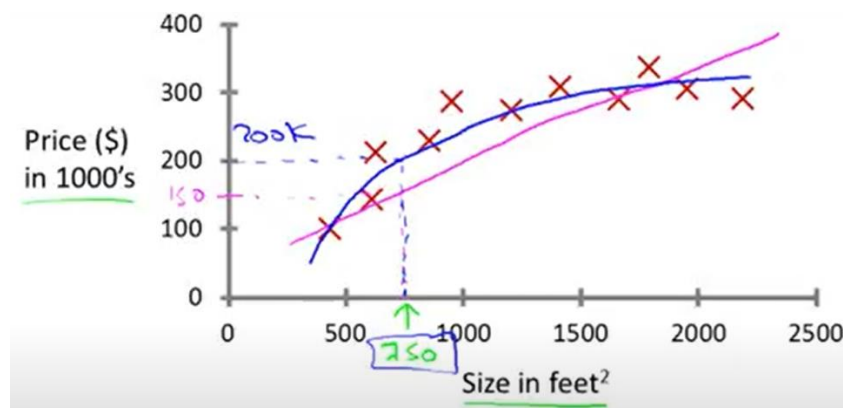
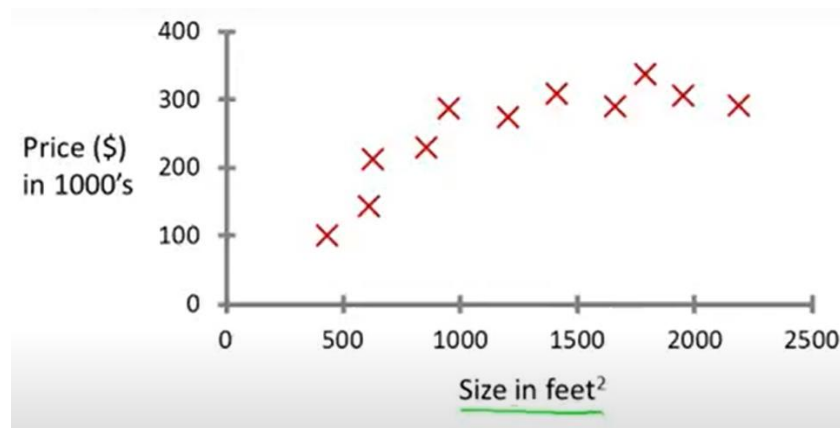


*Origen Veure apartat de Referències*

Un altre exemple de classificació podria ser predir si un tumor és maligne o benigne



Una altra tasca típica és predir un valor numèric objectiu, com ara el preu d'una casa, donat un conjunt de característiques (ubicació, metres quadrats,...) anomenades predictors. Aquest tipus de tasca és anomenada regressió. Per entrenar el sistema, cal posar-ne molts exemples de cases, incloent-hi els seus predictors i les seves etiquetes (és a dir, els seus preus).

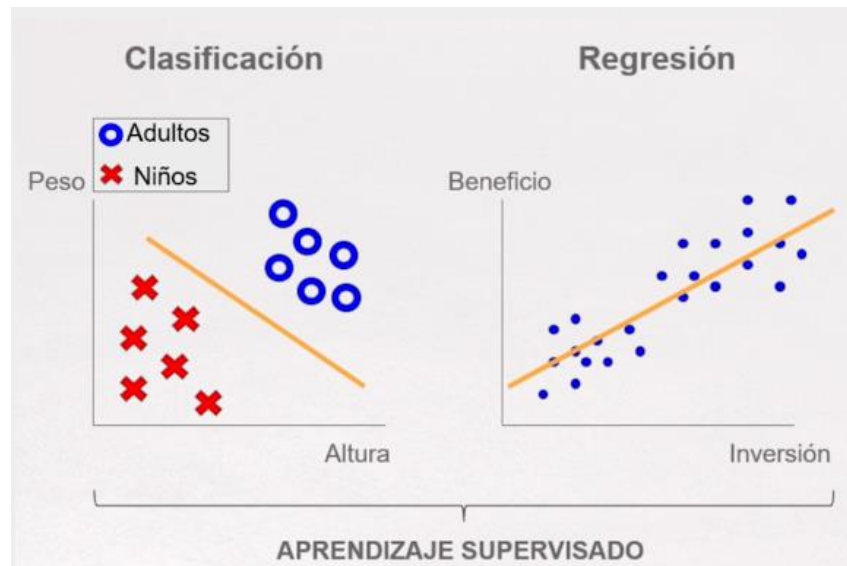


Alguns dels algorismes d'aprenentatge supervisat són:

- Regressió Lineal
- Regressió Logística
- K-Nearest Neighbors (kNN) k-veïns més propers (kNN)

- Support Vector Machines (SVM)
- Arbres de decisió i Random Forests
- Naive Bayes
- Xarxes Neuronals Artificials i Deep Learning

La llista dels anteriors algorismes els podem dividir en **algorismes de regressió** o **algorismes de classificació**. Els primers s'utilitzen per predir una dada numèrica i els segons s'utilitzen per predir una dada categòrica

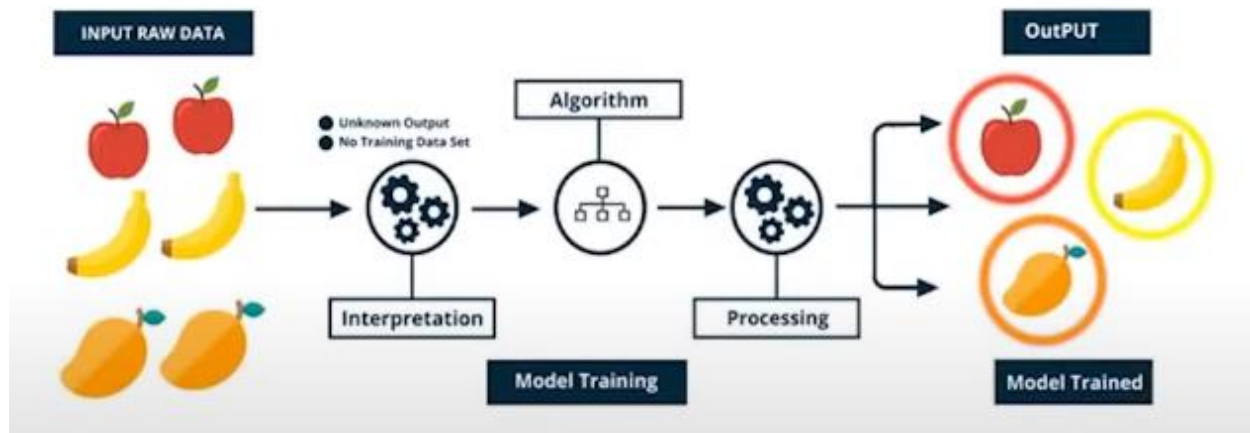


**Exercici.** Donats aquests dos problemes Indica per cadascun d'ells si és un problema de regressió o de classificació:

- Problema 1. Tens un gran inventari de productes idèntics. Volem predir quina serà la quantitat d'aquests productes vendrem en els pròxims 3 mesos. → **Regressió**
- Problema 2. Volem tenir un software capaç d'examinar totes les comptes d'usuari dels nostres clients i per cada compte decidir si és un compte que ha estat hackejat/compromés → **Classificació**

## Aprentatge No Supervisat (*Unsupervised Learning*)

En l'aprenentatge no supervisat les dades d'entrenament no estan etiquetades . El sistema intenta aprendre sense "professor". Això significa que no proporcionem etiquetes (*labels*) ni resultats esperats al model. El sistema rep només les característiques d'entrada (*features*) i ha de descobrir, pel seu compte, patrons, estructures o relacions dins de les dades. Aquí li diem a l'algorisme que ens trobi certes agrupacions amb característiques similars. En l'aprenentatge supervisat, el conjunt de dades d'entrenament conté tant les **entrades** (*features*) com les **sortides esperades** (*labels*).



Per exemple en les dades d'entrada que tenim pomes, plàtans i "taronges" l'algorisme interpreta totes les dades i detecta que hi ha 3 tipus de fruites, però en cap moment sap si són pomes, plàtans o "taronges".

Les instàncies de les dades d'entrenament no tenen associat un valor de sortida esperat, sinó que l'algorisme d'aprenentatge no supervisat detecta un patró basat en algunes característiques inicials de les dades d'entrada.

**L'algorisme no pot identificar etiquetes del grup**, l'algorisme només sap quines instàncies tenen **característiques similars**, però no pot identificar el seu significat.

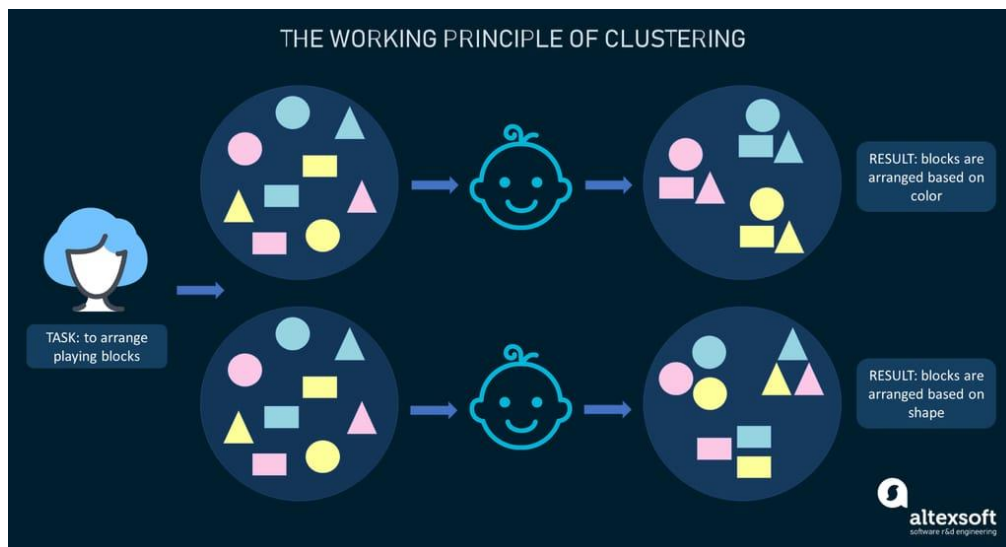
Dins de tots els models no supervisats en diferenciem diferents tècniques.

## Agrupament (*Clustering*)

D'entre totes les tècniques d'aprenentatge no supervisat, **l'agrupament** (clustering) és, sens dubte, la més utilitzada. Aquesta metodologia consisteix a agrupar dades similars en clústers que no estan definits prèviament.

Imaginem que estem en una classe d'infantil de primària i la mestra demana als nens que organitzin un conjunt de blocs de fusta.

Sense que la mestra indiqui com fer-ho, els nenes podrien decidir agrupar-los per forma, per color o per alguna combinació de les dues característiques. Això és el que fa el *clustering*: **trobar maneres naturals d'ordenar la informació** sense regles preestablertes.



Origen: Altexsoft

Com que **no hi havia una tasca prèviament definida**, no hi ha una manera correcta o incorrecte de fer l'agrupament. Aquesta és precisament la gràcia del *clustering*: permet **descobrir patrons i coneixements inesperats** dins de les dades, revelant informació que pot ser molt valuosa quer a negocis o per la presa de decisions.

Per exemple, imaginem un model d'aprenentatge que li proporcionem dades sobre les pel·lícules que han vist els susuaris en una plataform de streaming com Flimnin o Netflix.



Aquest dades podrien ser: **Historial de visualitzacions** de cada usuari, **Gèneres** de les pel·lícules, **Actors i actrius** protagonistes, **Directors**, **Descripcions** de les pel·lícules.

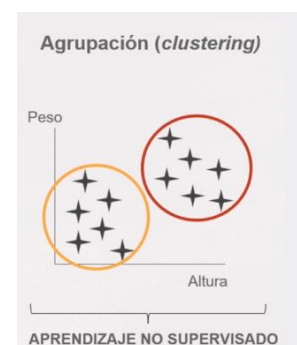
Aquí el model analitza totes aquestes característiques i **intenta agrupar les pel·lícules similars** segons el seus trets comuns creant categories com: comèdies romàntiques, thrillers, pel·lícules d'acció,...

Cas d'ús detecció d'anomalies o frau. Amb el clustering, és possible detectar qualsevol tipus de valors atípics (*outliers*) dins les dades. Per exemple en bancs o entitats financeres poden aplicar aquesta tècnica per detectar transaccions fraudulentament i actuar amb rapidesa, cosa que pot suposar un estalvi econòmic molt important. Mirar el [vídeo](#)



Casos d'ús o exemples:

- Segmentació de clients i de mercat. Els algorismes de clustering permeten agrupar persones amb característiques similars per tal de crear perfils de client. Amb aquesta informació es poden dissenyar campanyes de màrqueting i estratègies comercials més eficients per augmentar l'impacte de l'inversió.
- Suposem que està davant d'una festa a on no coneixes a ningú i no saps dins dels diferents grups a on es trobaràs més a gust. Per tant l'algorisme no supervisat mirarà de les teves característiques (edat, interessos, ...) a quin grup de la festa pots encaixar millor.
- En un partit de futbol podem classificar els jugadors de moltes maneres: per camiseta, per si juguen de porter o no, si són atacants o defensors.
- Sistema de recomanacions d'Amazon. Quan comprem a Amazon automàticament recomana altres articles. Aquí aplica sistema d'aprenentatge no supervisat basant-se amb diferents usuaris. Si Amazon detecta quin tipus de client compra un producte concret el podrà suggerir en una altre client del mateix tipus
- Un altre exemple seria si tenim un conjunt de dades de persones de les quals en tenim el seu pes i l'alçada. En funció d'aquestes dues variables podem separar dos grups clarament diferenciats tot i que no sabem el significat. Per exemple podrien se grups per sexe (homes, dones), edats (nens petits, adolescents),....



**Tipus de clustering:**

- Clustering exclusiu (*hard clustering*):** Cada element de les dades només pot pertànyer a un únic clúster. És el tipus més senzill i habitual quan les categories són clarament diferenciades.
- Clustering superposat (*soft clustering*):** Un element pot pertànyer a més d'un clúster amb diferents graus de pertinença. En aquest context, sovint s'utilitza el *clustering* probabilístic per:
  - Resoldre problemes de soft clustering



- Estimar la densitat de les dades
- Calcular la probabilitat que un punt de dades pertanyi a un clúster concret

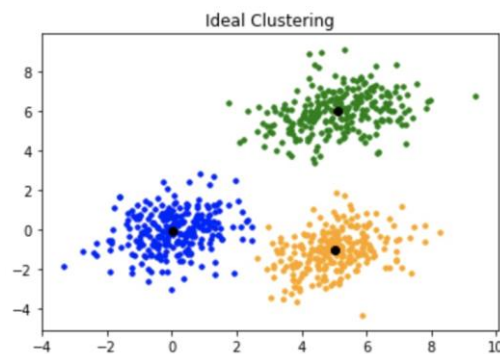
c) **Clustering jeràrquic:** Crea una jerarquia d'elements agrupats. Els clústers es poden obtenir:

- Descomponent grups grans en subgrups més petits (divisió — top-down)
- Fusionant elements individuals o grups petits en grups més grans (aglomeratiu — bottom-up)

Aquest tipus és útil per visualitzar les relacions entre grups a través d'un dendrograma.

**Algorismes populars de clustering:**

- **K-means Clustering:** La idea principal és dividir el conjunt de dades en un nombre predefinit de clústers, indicat per la lletra K.



- **Fuzzy K-means:** és una extensió de l'algorisme K-means. A diferència del K-means tradicional, en què cada punt de dades només pot pertànyer a un únic clúster, el Fuzzy K-means permet que un punt pertanyi a diversos clústers alhora. Aquesta pertinença múltiple es defineix mitjançant un grau de proximitat o pertinença (*membership*).



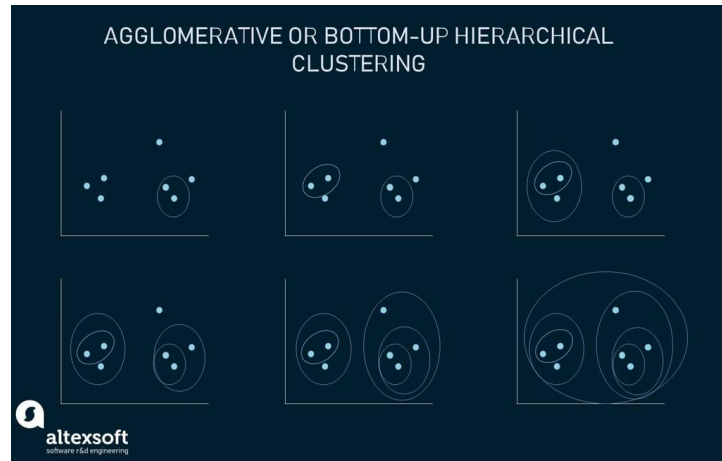
- **Gaussian Mixture Models (GMMs):** És un algorisme utilitzat en el clustering probabilístic. El model assumeix que les dades provenen d'una combinació d'un cert nombre de distribucions gaussianes (campanes de Gauss), on cada distribució representa un clúster diferent. Com que la mitjana i la variància de cada distribució són inicialment desconegudes, l'algorisme les estima a

partir de les dades. L'objectiu és determinar a quin clúster té més probabilitat de pertànyer cada punt de dades, assignant una probabilitat de pertinença per a cadascun.

Aquesta aproximació és útil quan els clústers no són clarament separats i poden solapar-se parcialment, ja que treballa amb graus de probabilitat en lloc de fronteres rígides. Ideal si les dades poden ajustar-se bé a “campanes” i vols assignacions soft (*soft clustering*)

- **DBSCAN** (*Density-Based Spatial Clustering of Applications with Noise*): És un algorisme de clustering basat en la densitat, molt utilitzat quan els grups de dades no tenen una forma clara i poden contenir soroll (outliers). Encara que aquest algorisme treballa amb densitats aquest no suposa que les dades tinguin una densitat probabilística suposada.
- **Clustering jeràrquic** (*Hierarchical Clustering*): L'enfocament jeràrquic crea una estructura en forma d'arbre (dendrograma) que mostra com es formen o es divideixen els grups de dades. Pot funcionar de dues maneres principals: aglomeratiu (*bottom-up*) o divisor (i)top-down

L'exemple mostra com 7 clústers diferents (punts de dades) es fusionen pas a pas en funció de la distància fins a formar un únic clúster gran.



## Reducció de dimensionalitat (*Dimensionality Reduction*)

Un altre tipus important d'aprenentatge no supervisat és la reducció de dimensionalitat.

L'objectiu és **reduir el nombre de característiques** (features) d'un conjunt de dades **mantenint la màxima informació rellevant possible**. Molt sovint podem pensar que quan més dades més precisos seran els nostres resultats. Però això no és així en moltes ocasions ja que moltes dades poden estar correlacionades donant problemes com el soroll (informació poc rellevant o redundant), sobreajustament (overfitting), cost computacional més alt, dificultat per visualitzar i entendre les relacions entre variables.

Per tant, utilitzem la reducció de la dimensionalitat per:

- Simplificar les dades
- Facilitar la visualització

- Millorar el rendiment d'altres algorismes de machine learning

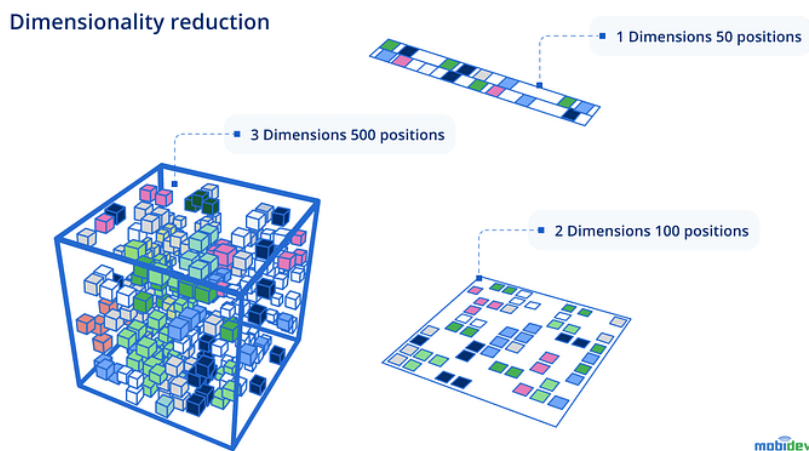
Exemple:

Suposem que tenim un conjunt de dades amb moltes característiques altament correlacionades. Les tècniques de reducció de dimensionalitat poden identificar els components principals que concentren la major part de la variància de les dades, i així reduir el nombre de característiques necessàries.

Per exemple:

- Característiques originals: nombre d'unitats venudes i nombre d'unitats retornades
- Nova característica: unitats netes venudes = vendes – devolucions

Això redueix dues característiques a una sense perdre informació essencial. Aquest procés també es coneix com **extracció de característiques** (*feature extraction*).



Origen: peerdh.com

Casos d'ús

La tècnica de reducció de dimensionalitat pot aplicar-se durant la fase de preparació de dades en projectes d'aprenentatge supervisat.

L'objectiu és eliminar dades redundants o irrelevants, mantenint només aquelles característiques més importants per al projecte.

Exemple pràctic: Imagina que treballes en un hotel i vols predir la demanda de diferents tipus d'habitacions. Tens un gran conjunt de dades amb:

- Informació demogràfica dels clients
- Nombre de vegades que cada client ha reservat un tipus concret d'habitació l'últim any

Analitzant les dades, observes:

- Tots els clients són dels EUA → aquesta variable té variància zero i es pot eliminar.
- L'esmorzar amb servei d'habitació està inclòs en tots els tipus d'habitació → no aporta informació útil per predir la demanda.
- Les característiques “edat” i “data de naixement” són duplicades (una es pot calcular a partir de l'altra) → es poden fusionar.

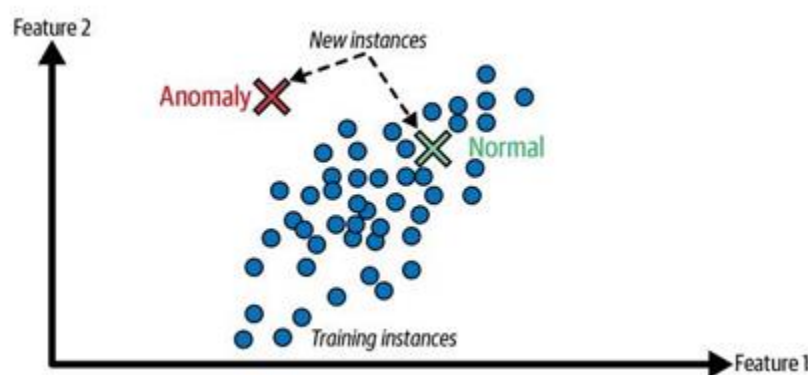
Així, redueixes la dimensionalitat del conjunt de dades, fent-lo més petit, net i rellevant per a l'anàlisi.

Tècniques habituals:

- PCA (Principal Component Analysis)
- Factor Analysis
- t-SNE (t-distributed Stochastic Neighbor Embedding)

## Detecció d'anomalies

La detecció d'anomalies és una aplicació molt rellevant de l'aprenentatge no supervisat. Aquests models estan dissenyats per identificar punts de dades inusuals o inesperats que es desvien significativament de la majoria del conjunt de dades. Això és el que coneixem habitualment o anomenem valors atípics (*outliers*).



Models utilitzats:

- SVM d'una sola classe (One-class SVM)
- Bosc d'aïllament (Isolation Forest)

## Aprentatge per regles d'associació (*Association Rule Mining*)

Per últim una altra tasca que utilitza models no supervisats és l'aprenentatge de regles d'associació. L'objectiu és explorar quantitats enormes de dades i descobrir relacions interessants entre les diferents característiques.

Per exemple en un anàlisi de dades de vendes de supermercat podem trobar que la gent que compra salsa barbacoa i patates fregides també tendeix a comprar bistecs. Per tant, potser ens convingui col·locar aquests productes a prop.

Alguns models més comuns:

- Algorisme Apriori
- Algorisme FP-growth (*Frequent Pattern Growth*)

Aquests mètodes identifiquen conjunts d'articles que apareixen junts amb freqüència i calculen mesures com:

- Suport (support): freqüència amb què apareix la combinació.
- Confiança (confidence): probabilitat que un article sigui comprat quan s'ha comprat l'altre.
- Elevació (lift): grau en què la presència d'un article incrementa la probabilitat de compra d'un altre.

## Aprentatge per reforç (*Reinforcement Learning*)

L'algorisme d'aprenentatge de reforç és força diferent dels altres que s'han comentat. En lloc d'aprendre a partir d'un conjunt de dades fix, un agent de reforç aprèn interactuant de manera iterativa constant amb el seu entorn i basada en "prova i error".

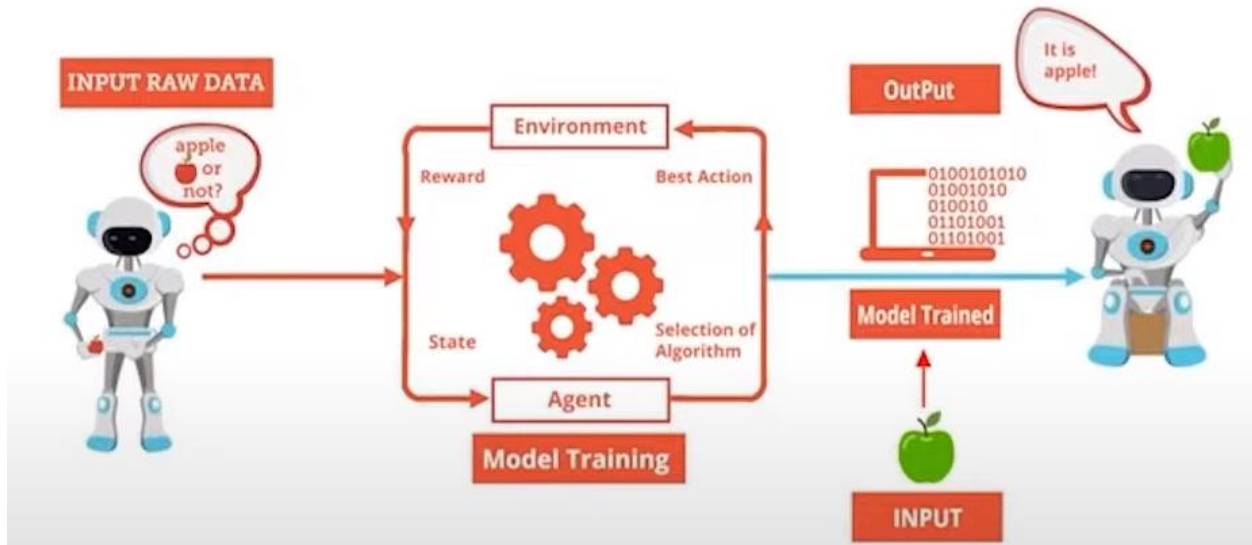
Es refereix a la interacció entre l'entorn i l'agent d'aprenentatge. L'agent d'aprenentatge es basa en l'exploració i l'explotació.

Hi ha diversos conceptes clau en l'aprenentatge per reforç:

- **Agent:** L'aprenent que pren decisions.
- **Accions:** Les opcions que l'agent pot prendre dins de l'entorn.
- **Estat:** La situació actual de l'entorn.
- **Recompenses:** Retroacció positiva que rep l'agent per dur a terme accions desitjables.
- **Penalitzacions:** Retroacció negativa que rep l'agent per dur a terme accions indesitjables.

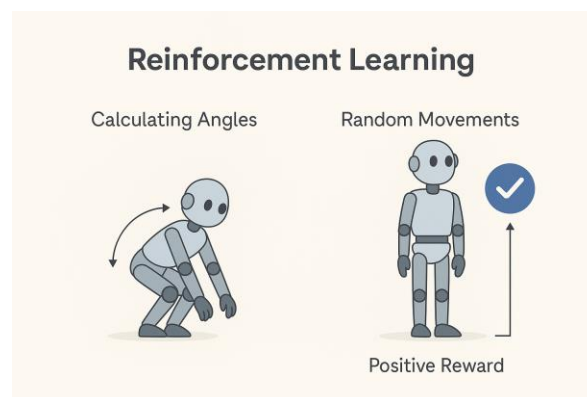
- **Entorn:** El món amb el qual l'agent interactua.

L'agent observa l'estat actual de l'entorn i després pren una acció. Aquesta acció provoca que l'entorn transiti cap a un nou estat, i l'agent rep una recompensa o una penalització segons el resultat de la seva acció. L'objectiu de l'agent és aprendre una política, és a dir, una estratègia que indiqui quina és la millor acció a prendre en qualsevol estat per tal de maximitzar les recompenses acumulades al llarg del temps.



Origen: Edureka

Un exemple podria ser el següent. Volem construir un robot capaç d'aixecar-se de terra. Tindriem 2 opcions o programar el robots de tal manera que calculem els angles i la força de cada articulació/motors tenein en compte el centre de gravetat perquè no caigui, etc... o bé podríem dir-li al robot que realitzés moviments aleatoris de tal manera que si el moviment és més proper a l'objectiu (estar dret) se li doni una recompensa i així successivament fins que el robot trobi la seqüència de moviments correctes per aixecar-se.



Autor: ChatGPT

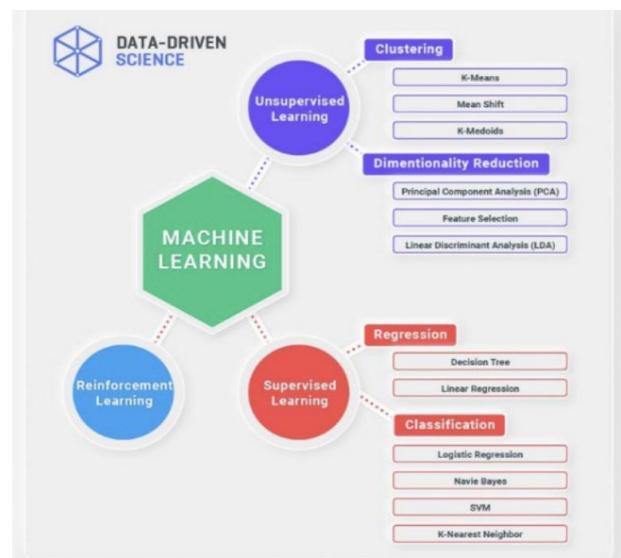
Un clar exemple el tenim en **AlphaGo**, el sistema de DeepMind (filial de Google) que va derrotar el campió mundial Lee Sedol en el joc de Go l'any 2017, fet que va tenir un gran ressò mundial. El model va ser entrenat analitzant milions de partides de Go jugades per humans per aprendre les regles bàsiques i les estratègies. Encara més important, després va jugar innumbrables partides contra ell mateix, perfeccionant la seva política sobre quina és la millor jugada en cada situació mitjançant assaig i error, i maximitzant la seva "recompensa" (guanyar la partida).

El curiós del cas és que a AlphaGo no se li van ensenyar les regles del joc GO. [Vídeo1 \(ENG\)](#), [Vídeo2\(ES\)](#).

[Documental AlphaGo \(2017\) amb subtítols en castellà.](#)

Un altre clar exemple és [AWS Deep Racer](#).

## Principals algorismes de ML



[Documental sobre Machine Learning i l'entrenament humà](#)

# Reptes del Machine Learning

El fet que hàgim de seleccionar un model i entrenar-lo amb dades, ens podem trobar amb dos problemes. En seleccionar un "mal model" o tenir "males dades"

## Dades de poca qualitat

Una de les **primeres coses que hem de revisar** abans d'entrenar un model és si les dades d'entrenament contenen **errors, valors atípics o soroll**

Tot i que pugui semblar una tasca menor, val molt la pena dedicar temps a netejar les dades abans de qualsevol entrenament. Aquesta etapa rep el nom de **Data Cleaning** i dins el món *Data Science* és una de les activitats que consumeix més temps i esforç.

### Valors atípics (*outliers*)

Són dades que estan molt allunyades del comportament normal.

#### Què podem fer-hi?

- Eliminar les instàncies que contenen aquests valors extrems.
- Corregir-los si sabem que són deguts a un error.
- O, en alguns casos, mantenir-los si tenen sentit (per exemple: un client amb una despesa molt alta però real).

### Valors que falten (*missing values*)

És molt habitual trobar dades incompletes. Per exemple, algunes instàncies poden no tenir registrat un atribut.

#### Opcions per tractar-ho:

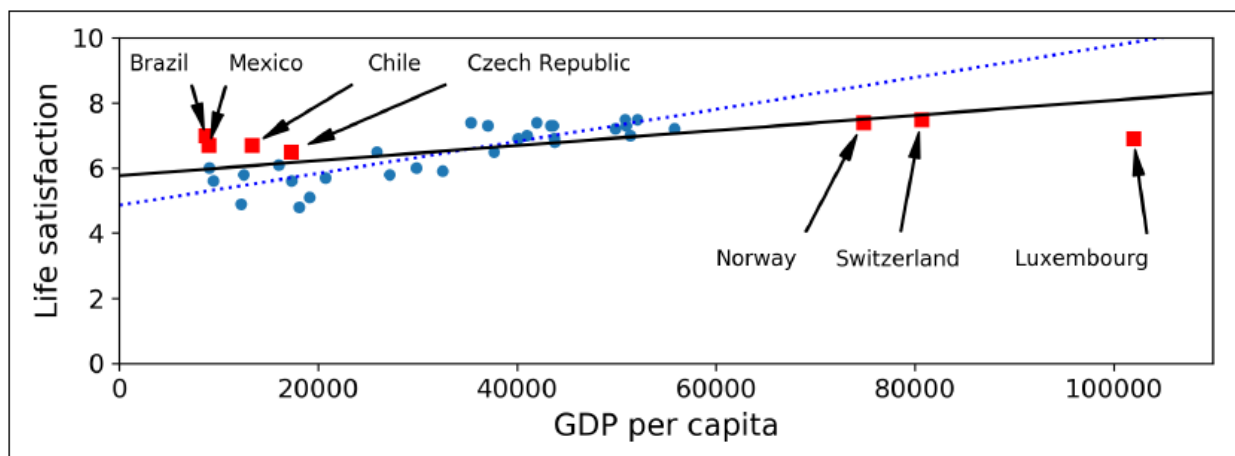
- Descartar l'atribut si té massa valors buits (i no és rellevant).
- Eliminar només les files amb valors que falten.
- Omplir els valors que falten amb: La mitjana (si són números), El valor més comú (si és categòric). Una estimació basada en altres valors similars.
- Entrenar dos models: un amb l'atribut complet i un altre sense, i comparar quin funciona millor.

## Dades d'entrenament no representatives



Per generalitzar és bàsic que les dades d'entrenament siguin representatives dels nous casos als quals volem generalitzar.

Per exemple, el conjunt de països que hem utilitzat per entrenar el model no era perfectament representatiu, **faltaven alguns països (quadrats vermells)**. Es veu que si s'haguessin afegit aquests països la recta de regressió seria diferent.



## Dades desbalancejades

En problemes de **classificació**, un **conjunt de dades desbalancejat** és aquell en què hi ha molts més exemples d'una classe que d'una altra. Això pot fer que el model aprengui a ignorar les classes minoritàries i no sigui capaç de detectar-les correctament.



El model pot tenir una **precisió aparentment alta**, però **no detectar** mai la **classe** que realment ens interessa.

Per exemple si volem detectar una certa malaltia rara a una població i aquesta malaltia és molt minoritària 1% de la població. Si agafem una representació de la població per l'entrenament pot ser que no detectem la malaltia perquè aquell subconjunt no hi ha ningú amb aquella malaltia.

Per arreglar aquest tipus de problemes podem:

- **Down-sampling:** Reduir el número d'exemples de la classe majoritària fins tenir una població balancejada i poder entrenar el model.

Per exemple Si tenim 10.000 exemples de "sans" i només 100 de "malalts", seleccionem aleatòriament 100 "sans" i els combinem amb els 100 "malalts" → 200 exemples balancejats.

Un dels invonvenients que podem tenir és que perdrem molta informació útil de la classe majoritària.

- **Over-sampling:** Augmentar el número d'exemples de la classe minoritària duplicant dades o creant-ne de noves sintèticament amb tècniques com SMOTE (*Synthetic Minority Over-sampling Technique*).

Per exemple tenim 100 "malalts" i 10.000 "sans". Generem 9.900 exemples nous de "malalts" (amb SMOTE o duplicació) per tenir 10.000 de cada

El problema que ens podem trobar utilitzant aquest tècnica és caure en el sobreajustament (*overfitting*)

- **Matriu de cost:** Penalitzar més els errors comesos al classificar erròniament una classe minoritària durant l'entrenament.

Si el model confon un "malalt" amb "sà", li assignem una penalització més alta que si confon un "sà" amb "malalt". Això fa que li doni més importància a classificar bé la classe minoritària.

Aquesta tècnica és útil quan no volem tocar les dades, però requereix d'ajustar bé els pesos de penalització.

Quan treballem amb dades reals, no sempre són equilibrades. Si ignorem aquest fet, podem construir models que semblin bons però que no serveixin en la pràctica. Per això, és fonamental:

- Analitzar la distribució de les classes
- Aplicar tècniques com *down-sampling*, *over-sampling* o matrius de costos
- Avaluar bé amb mètriques adequades (no només *accuracy*!)

## Dades d'entrenament insuficients (*Underfitting*)

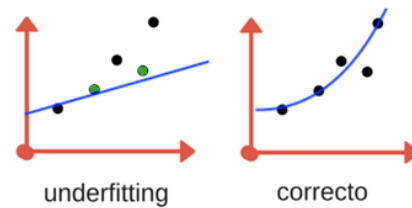
Si volem que un nen petit aprengui què és una poma i la distingeixi de les altres fruites hem de presentar-li una poma i dir-li que això és una poma. El cervell del nen assimilarà la forma i color de la poma per distingir-les de les altres fruites. Potser hem de repetir aquest procés algunes vegades per ensenyar-li que hi ha pomes d'altres colors (vermelles, grogues, verdes,...) però amb pocs exemples serà suficient.

Aquest no és el cas del machine learning. La majoria d'algoritmes requereixen de moltes dades perquè funcionin bé.

La quantitat de dades de l'entrenament han de ser suficients. Si tenim poques dades, és difícil trobar un

patró generalista.

En el gràfic veiem que les dades d'entrenament només han estat dues (les de color verd) amb aquestes dades el model ens marca una recta que no té res a veure amb el que hauria de ser.



Un altre exemple és si en el nostre model utilitzem només una raça de gossos per entrenar el model i llavors volem que reconegui a altres 10 races diferents l'algorisme no serà capaç de donar-nos un bon resultat.

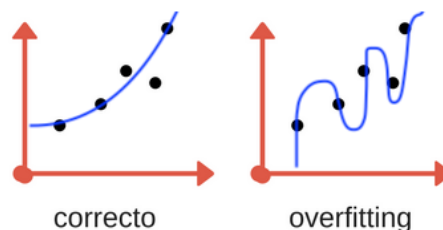
## Overfitting Training Data (sobreajustament)

L'overfitting es produeix quan un model aprèn massa bé les dades d'entrenament, fins al punt que memoriza els exemples en lloc d'entendre'n els patrons generals.

Això fa que funcioni molt bé amb les dades amb què ha estat entrenat, però fracassi quan es troba amb dades noves, encara que siguin correctes o similars.

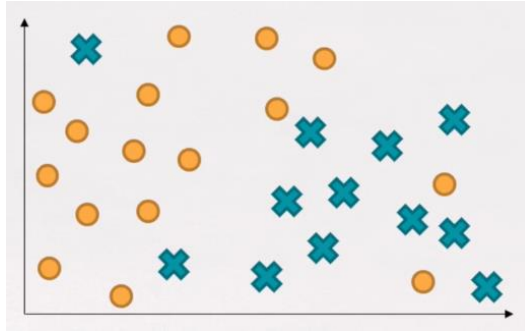
En la següent gràfica veiem que el que volem modelar és una paràbola, però el nostre model dona invàlides dades vàlides.

Resumint, tenim overfitting quan el nostre model només pot produir resultats singulars i amb la impossibilitat de comprendre noves dades d'entrada.

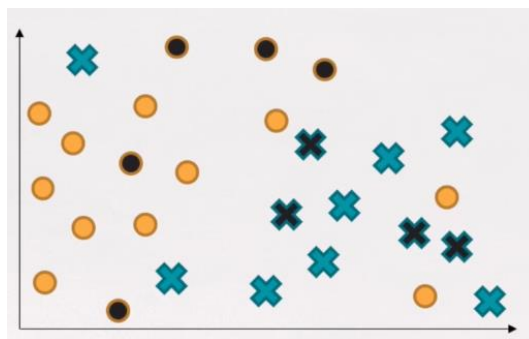


### Exemple

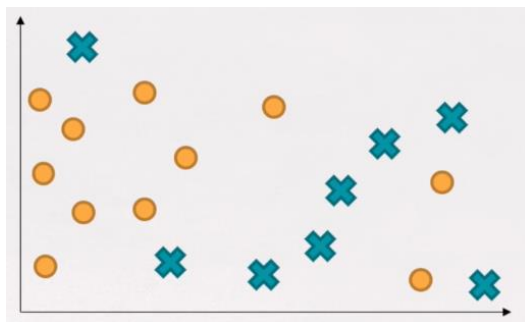
Donades aquestes dades volem aplicar un model de classificació perquè ens classifiqui cercles i creus.



Tal com hem dit hem d'agafar un subconjunt per test/validació i un altre per l'entrenament. En aquest cas marquem amb color negre les dades de test.

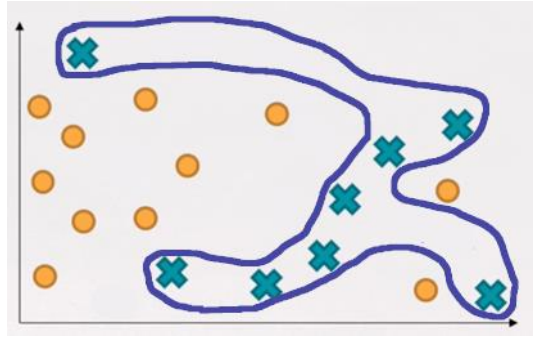


Per tant les dades d'entrenament són aquestes:

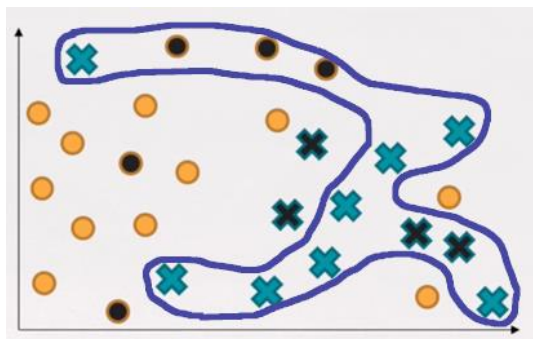


Si sobreentrenem (*overfitting*), intentem d'agafar tots els valors com a bons. En aquest exemple tenim un valor atípic a dalt a l'esquerra (una creu) i dos a baix a la dreta (dos cercles)

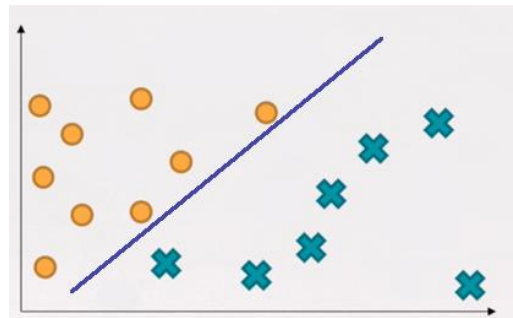
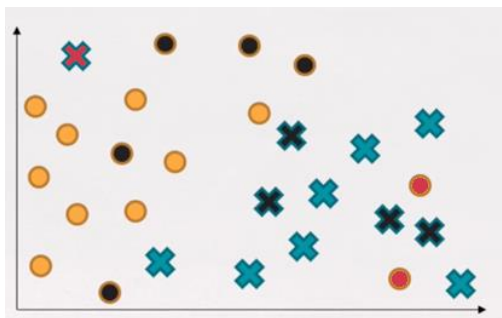
El resultat del model amb overfitting seria aquest.



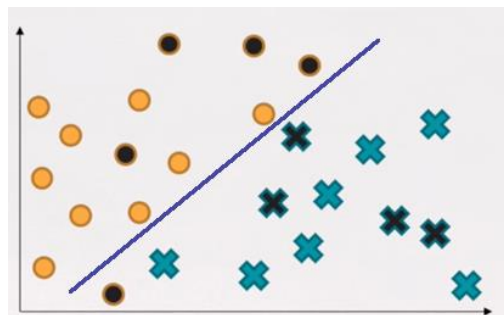
Veiem que el model amb les dades de test les classifica malament. De les 9 dades de test 5 les classifica malament això és un 55% d'error.



En canvi si el model l'entrenem sense les dades atípiques.



Amb aquest simple canvi veiem que només classifica malament 1 valor de 9, un 11% d'error. Però a simple vista veiem que el model generalitza molt millor.

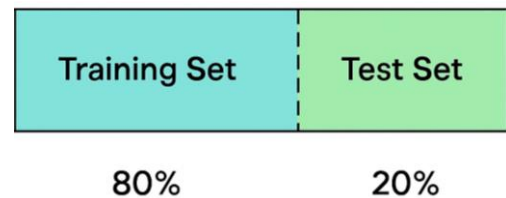


# Validació

El gran repte que tenim quan entrenem un model de Machine Learning és veure si funciona bé o no. La única forma de saber si el nostre model generalitza bé o no és provant nous valors.

Podríem pensar que la millor manera de saber si el model funciona és posar-lo en producció i monitoritzar-lo per a veure què passa. Això podria funcionar, però és arriscat. Si el model no funciona bé, els usuaris rebran males recomanacions o decisions incorrectes i això pot fer que perdin la confiança en el sistema i costi molt de recuperar-la.

Una opció millor, tal i com s'ha comentant anteriorment, és **dividir** les dades que tenim en **dos conjunts**: el **conjunt d'entrenament** i el **conjunt de test**. La idea és entrenar al model amb el subconjunt de les dades que hem anomenat d'entrenament i validar el model amb dades del subconjunt que hem anomenat test. D'aquesta manera podrem provar el model amb dades que no ha vist mai i així poder avaluar el model. Si l'error d'entrenament és baix amb les dades d'entrenament i alt amb dades de test significarà que tenim el nostre model sobreentrenat i estem caient en *overfitting*.



Com també s'ha dit anteriorment és habitual utilitzar el 80% de les dades per l'entrenament i reserva el 20% per el test.

---

*Collage de fotos de muffins y chihuahuas (<https://blogs.upm.es/pasd/2023/06/12/uso-de-aprendizaje-multi-tarea-para-abordar-el-problema-del-muffin-chihuahua/>)*

## REFERÈNCIES

- Aurélien Géron (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.
- <https://medium.com/@vishalgimhan/types-of-machine-learning-approaches-part-1-3-ml02-0042e443d989>
- Continguts de la docent Cristina Gómez de IES de Teis Vigo