



Universidad Nacional Autónoma de México
Facultad de Ingeniería

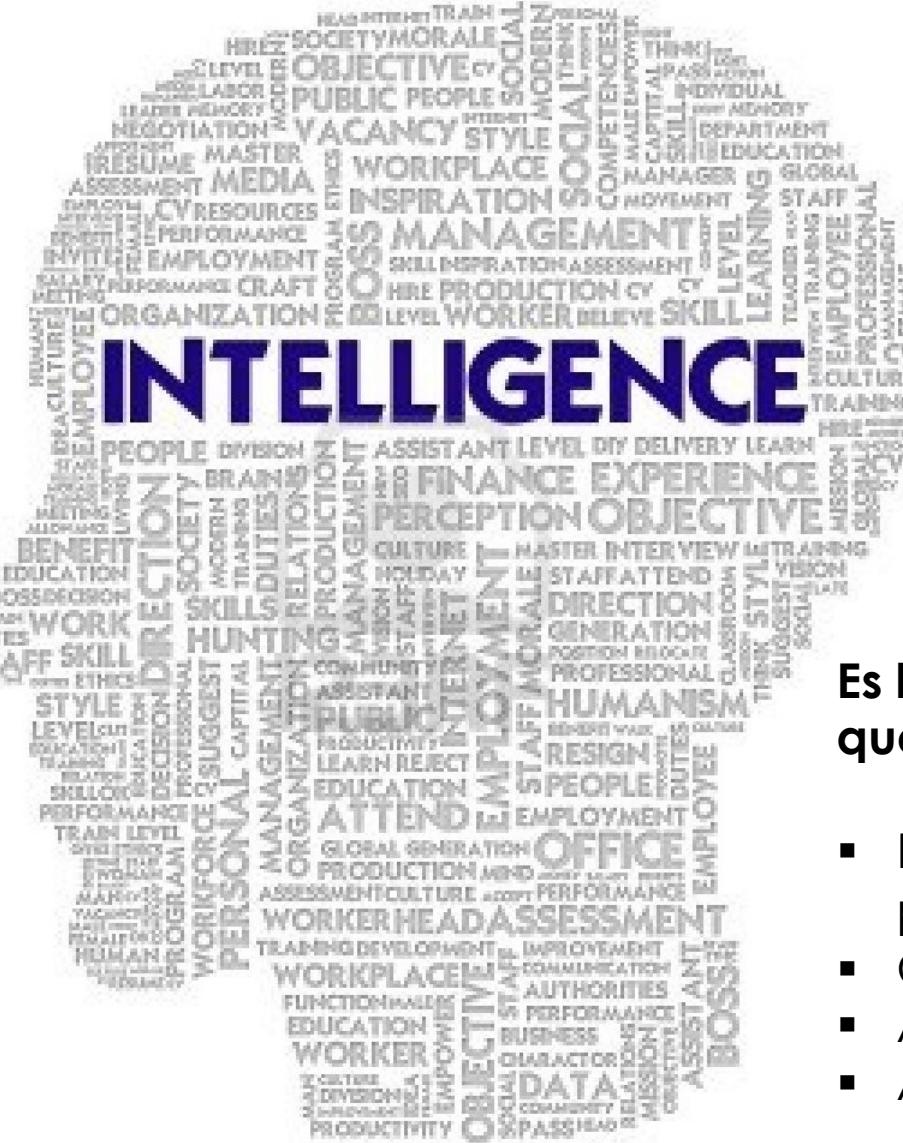
Inteligencia Artificial

Guillermo Molero-Castillo

guillermo.molero@ingenieria.unam.edu

Septiembre, 2021

INTELLIGENCE



**Es la capacidad de decidir bien,
que implica:**

- Habilidad para razonar, planificar y resolver problemas.
 - Comprender ideas complejas.
 - Aprender con rapidez.
 - Aprender de la experiencia.

1. Definición

Inteligencia Artificial

Definición

Es la ciencia y la ingeniería de crear máquinas inteligentes, especialmente programas de computación inteligentes, que comprendan la inteligencia humana.

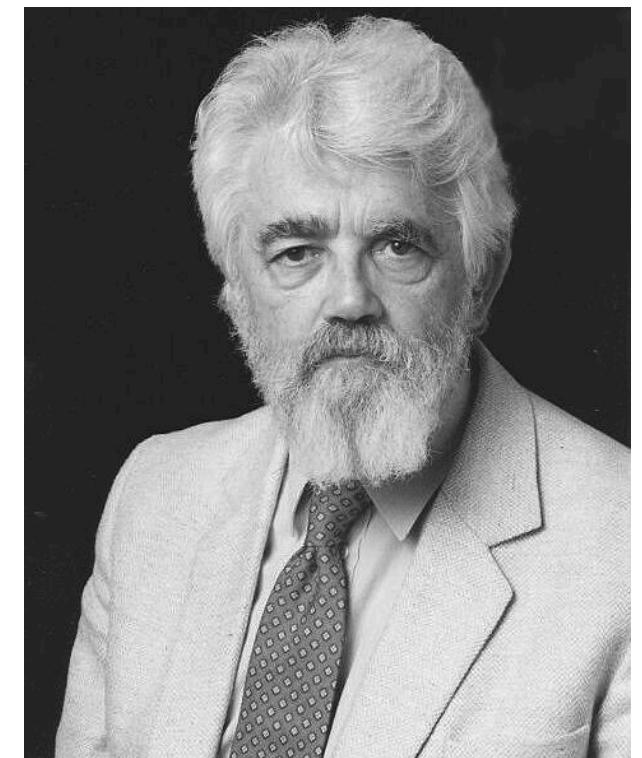
John McCarthy
Inteligencia Artificial
1927-2011
1954-1956

Definición

Padre de la Inteligencia Artificial

Bautizó el campo de estudio como Inteligencia Artificial

- Primera conferencia en 1956: *Dartmouth Summer Research Project on Artificial Intelligence*.
- Un nuevo lenguaje de programación nació (1958) de las ideas de dicha conferencia: LISP (LISt Processor).
- Recibió el Premio Turing en 1971.



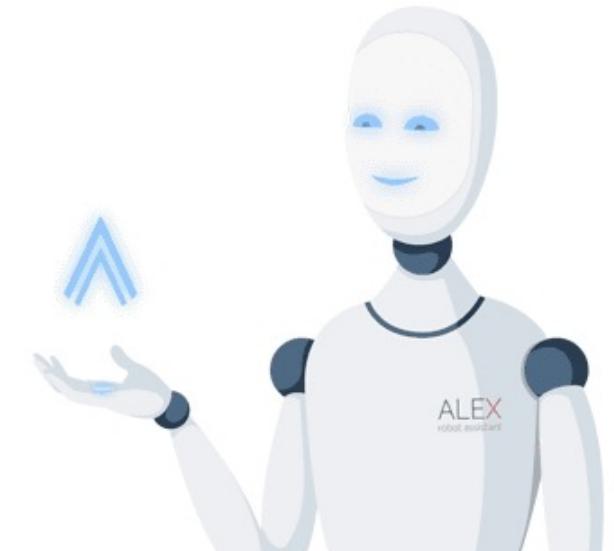
John McCarthy (1927-2011)

Information Processing Language (IPL)

IPL fue un lenguaje de programación de bajo nivel (casi tanto como el ensamblador) que fue creado en 1956 para probar teoremas a través de la computación (Bertrand Russell y Alfred North Whitehead).

- IPL introdujo en la programación características que hoy siguen vigentes como el manejo de símbolos, recursividad y el uso de listas.
- Se desarrollaron los primeros programas de IA: *Logic Theorist* (1956) y el programa de ajedrez NSS (1958).
- **Pese a su importancia en la historia de la IA**, varios factores (el primero de ellos, lo complejo de su sintaxis) hizo que fuera rápidamente sustituido por LISP.

IPL
is an acronym for
Information Processing
Language
⊕



Cronología de dialectos LISP (lenguaje de programación)



Presentado por McCarthy en 1958 en el MIT.

Es el segundo lenguaje de programación de alto nivel de mayor antigüedad; apareció un año después de FORTRAN.

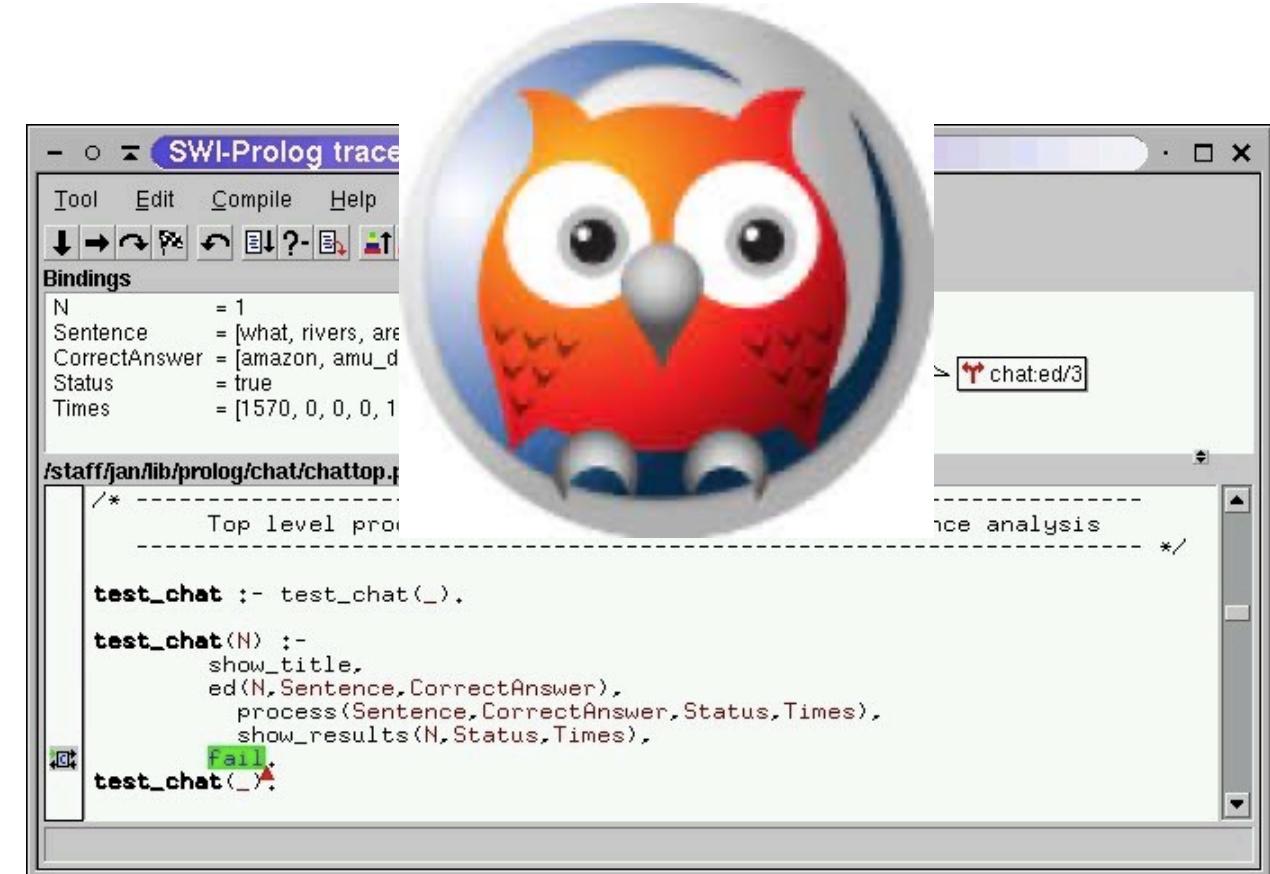
LISP (LIST Processor) se convirtió en el lenguaje de referencia en el ámbito de la IA.

Con el tiempo, **LISP** se fragmentó en toda una serie de 'dialectos' aún vigentes en varios ámbitos de la computación, como Common LISP, Emacs LISP, Clojure, Scheme o Racket.

PROLOG

PROLOG (del francés '*programmation en logique*') nació en 1972 (Alain Colmerauer), cuando el furor inicial por esta tecnología disminuyó debido al escepticismo provocado por la ausencia de avances, lo que generó una reducida inversión pública y privada en su desarrollo.

Aunque globalmente nunca llegó a ser tan usado como LISP, sí se convirtió en el principal lenguaje de desarrollo de IA en su continente de origen (así como en Japón).



* La facilidad que PROLOG proporciona para gestionar métodos recursivos y las coincidencias de patrones provocaron que **IBM** apostara por implementar PROLOG en su IBM Watson para tareas de procesamiento de lenguaje natural.

Python

Python fue creado en 1991 (Guido Van Rossum), es hoy en día el lenguaje de programación más usado en proyectos de IA, sobre todo en el campo del 'machine learning'.

Cuenta con amplias bibliotecas de IA (Keras, TensorFlow, SciPy, Pandas, Scikit-learn, y otros), que están diseñadas para trabajar con Python.



* En el presente, aprender IA casi se ha convertido en sinónimo de aprender a programar en Python.

Inteligencia Artificial

Otras definiciones

Área de estudio que tiene por objetivo resolver problemas complejos, para los cuales no se conocen soluciones algorítmicas exactas computables en la práctica, ya sea por su complejidad o los niveles de incertidumbre de los datos que manejan.

**Introducción a la Computación
Cap. 11. - Inteligencia Artificial
Cengage Learning
2008**

Inteligencia Artificial

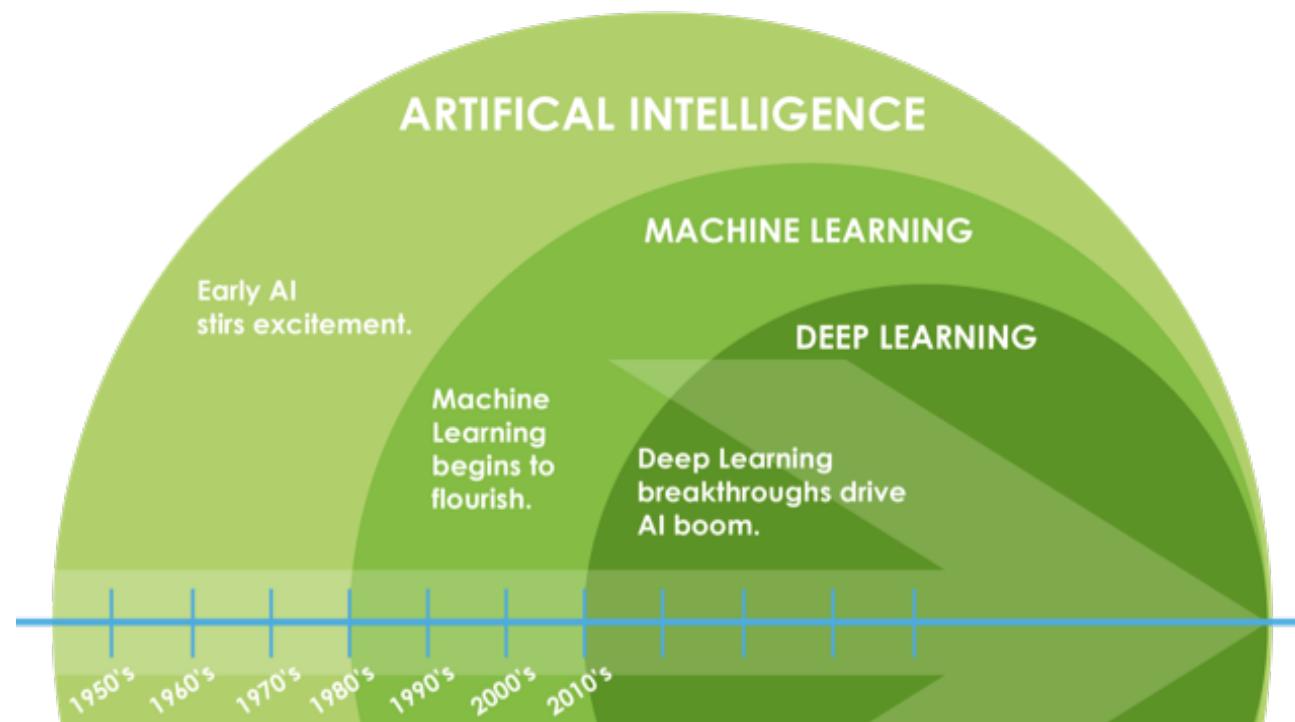
Otras definiciones

Inteligencia Artificial hace referencia a la forma de representación de la lógica humana en un sistema. Se usa generalmente para realizar tareas que requieren inteligencia y especialización.

**Inteligencia artificial y aprendizaje automático
FICO, Fair Isaac Corporation
2019**

Otras definiciones

En concreto, es un tipo de inteligencia no biológica impulsada por el **aprendizaje automático y aprendizaje profundo**.



Inteligencia Artificial

En el contexto actual

Inteligencia Artificial

Emula el comportamiento humano a través de las computadoras (máquinas). [IBM Deep Blue Chess](#)

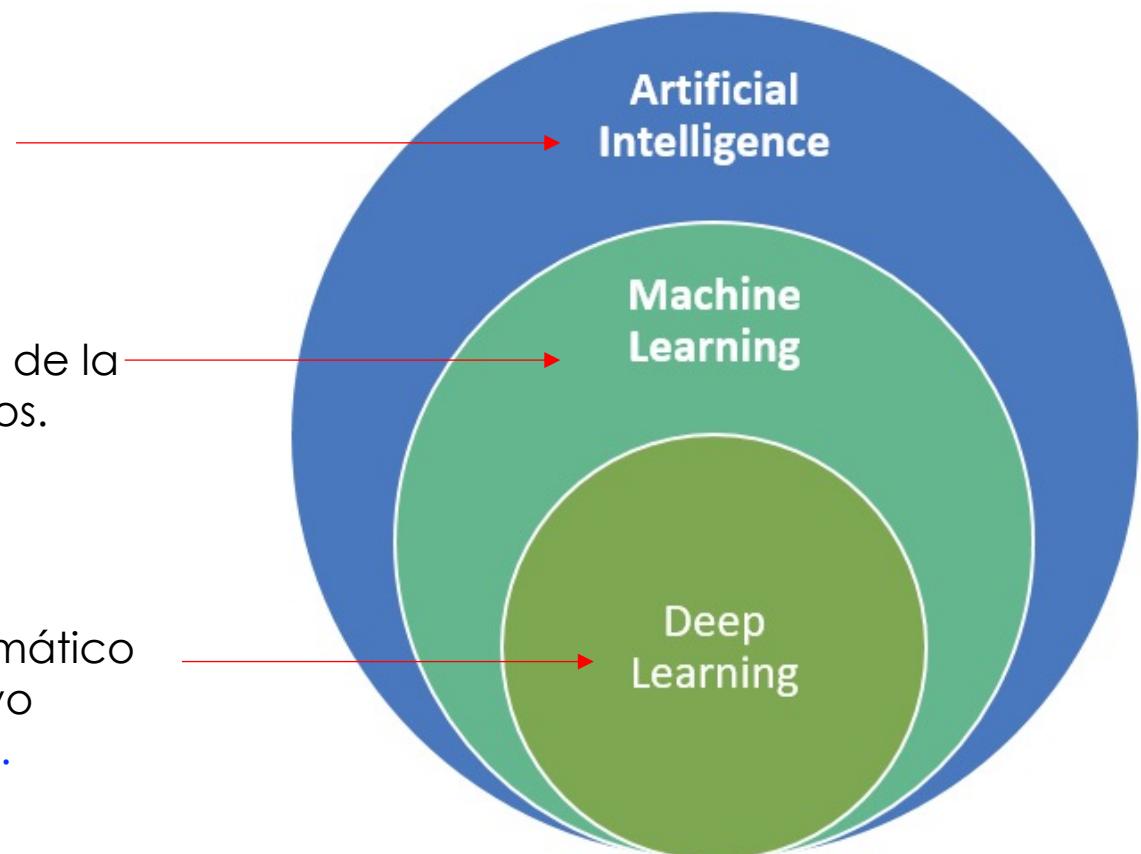
Aprendizaje Automático

Es un subconjunto de algoritmos de IA que se ocupa de la extracción de patrones a partir de conjuntos de datos.

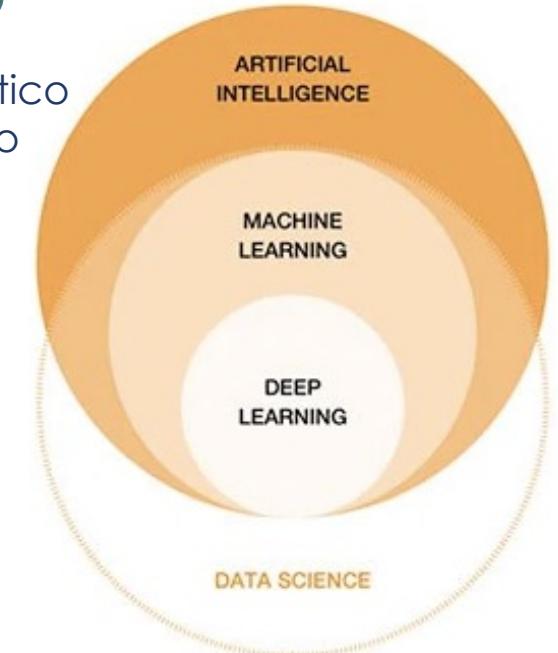
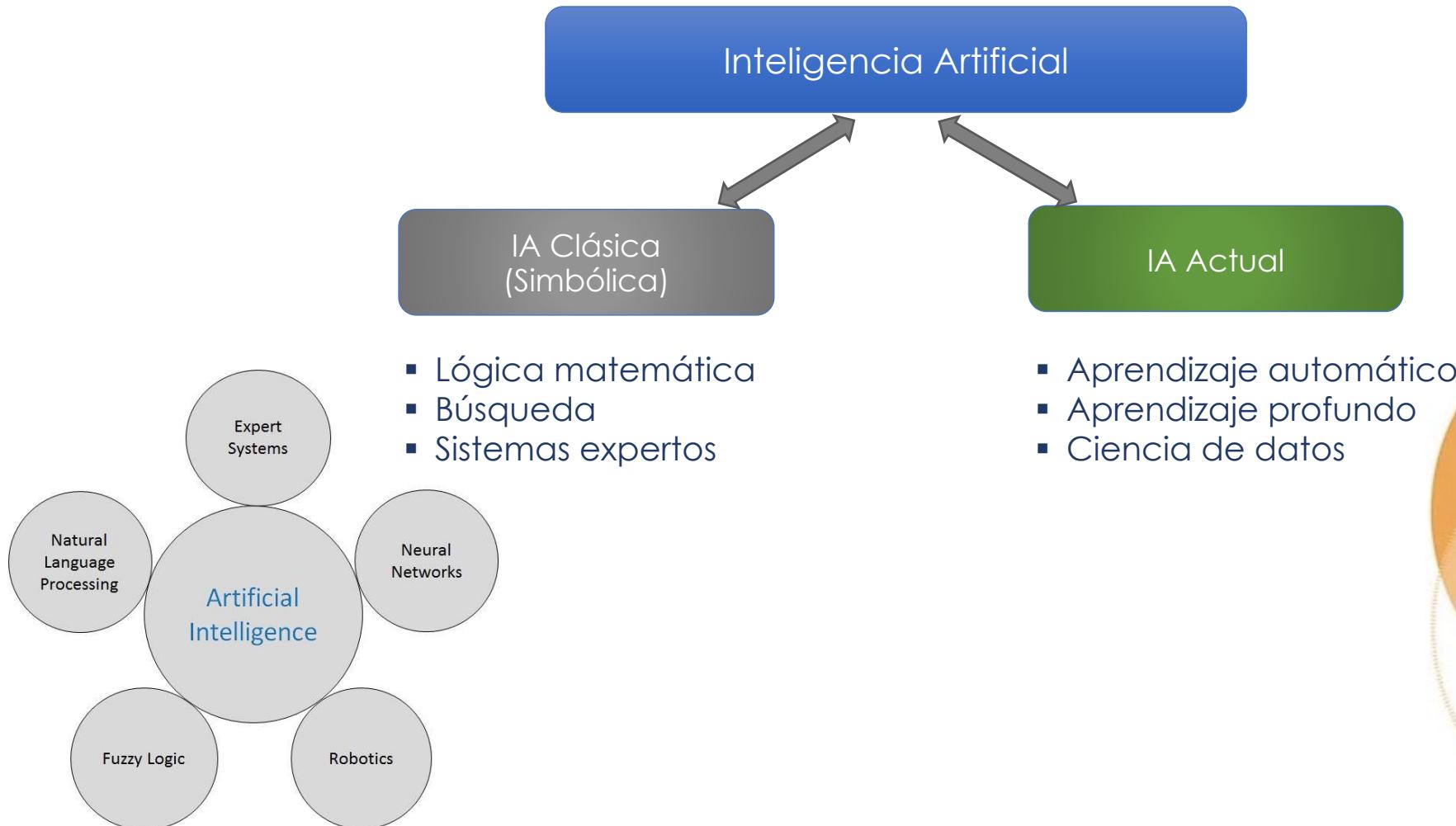
[Recomendaciones de Amazon](#)

Aprendizaje Profundo

Es un subconjunto de métodos de Aprendizaje Automático que utiliza redes neuronales complejas con el objetivo emular el aprendizaje humano. [AlphaGo de Google](#).



Inteligencia Artificial



Lectura de apoyo

En busca de una nueva definición para la inteligencia de las máquinas

<https://theconversation.com/en-busca-de-una-nueva-definicion-para-la-inteligencia-de-las-maquinas-118742>

En busca de una nueva definición para la inteligencia de las máquinas

THE CONVERSATION

Academic rigor, journalistic flair

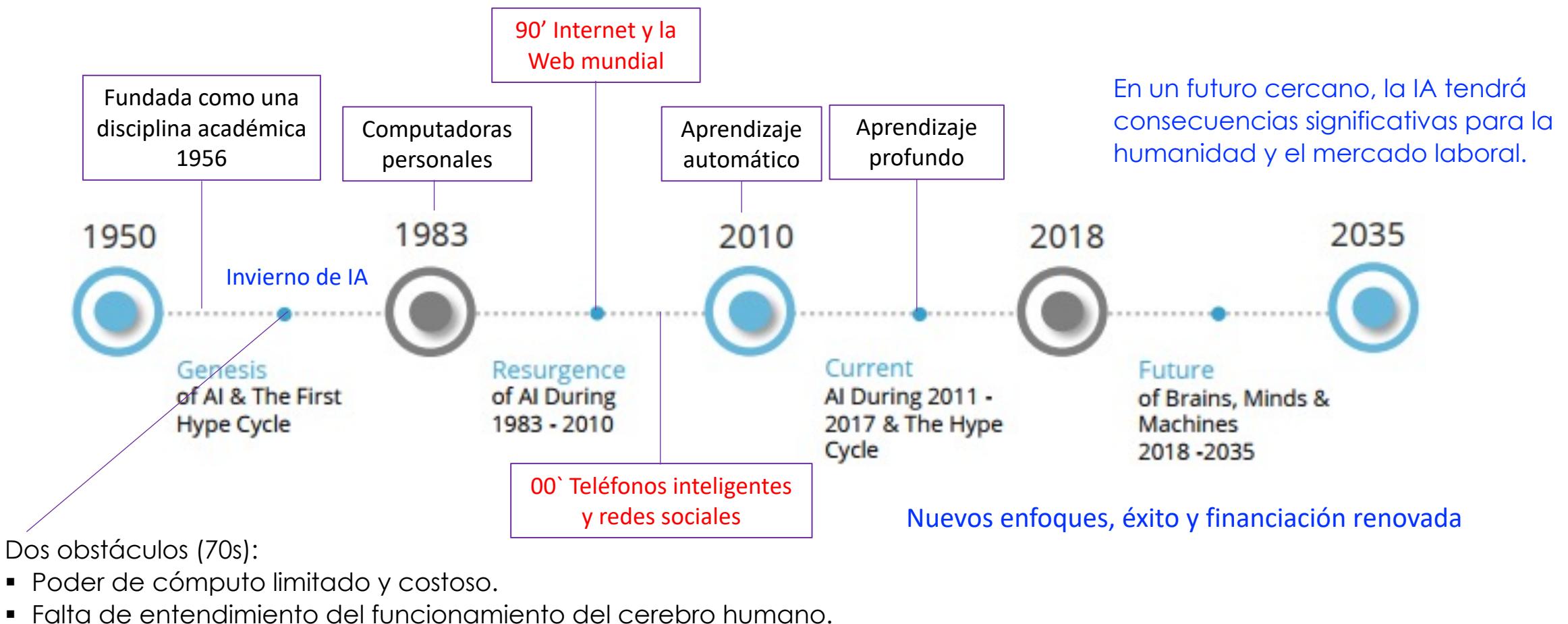
COVID-19 Arts + Culture Economy + Business Education Environment + Energy Ethics + Religion Health Politics + Society Science + Technology



En busca de una nueva definición para la inteligencia de las máquinas

2. Trayectoria

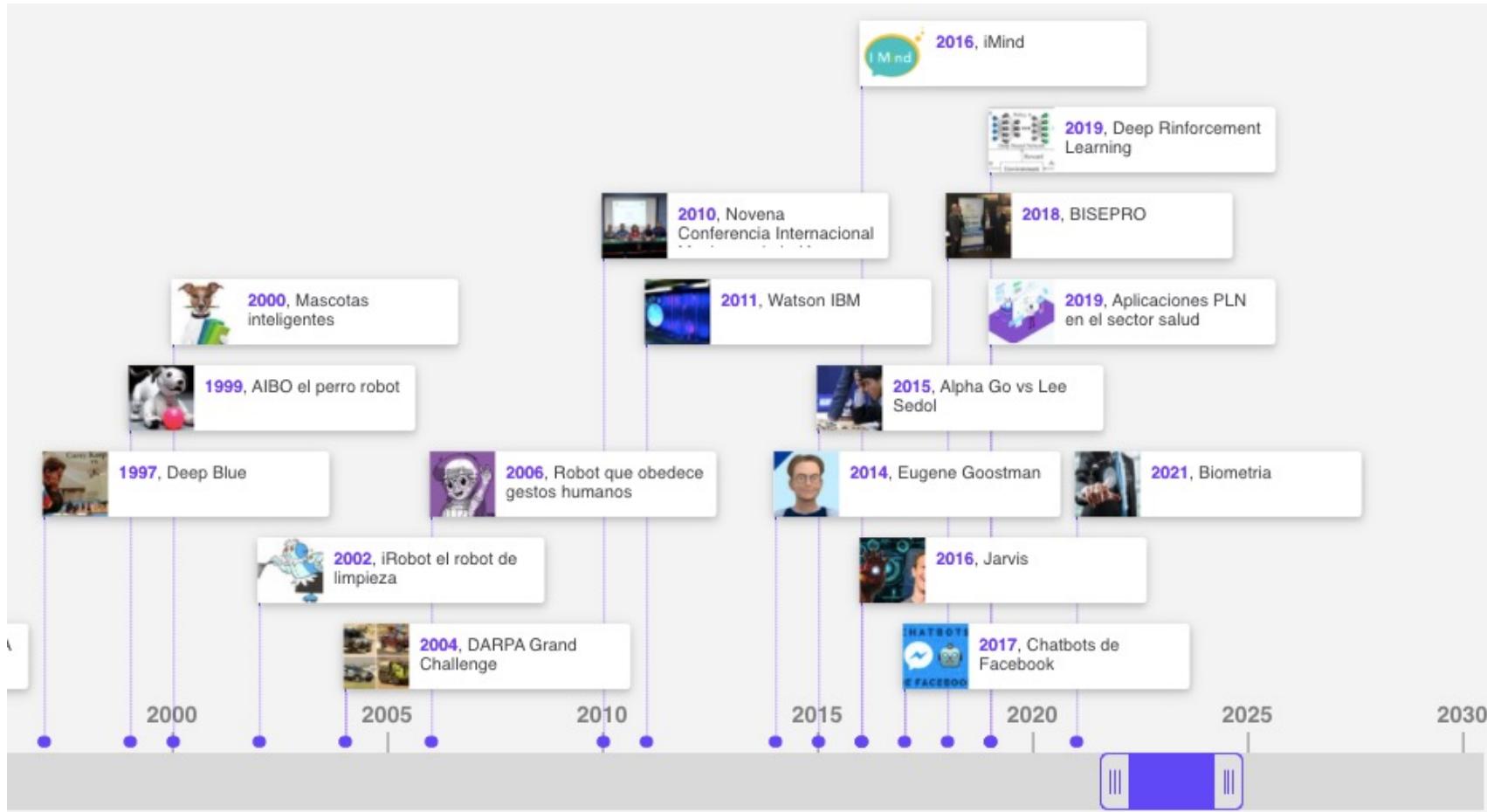
Trayectoria: Ciclo de la Inteligencia Artificial



Dos obstáculos (70s):

- Poder de cómputo limitado y costoso.
- Falta de entendimiento del funcionamiento del cerebro humano.

Trayectoria

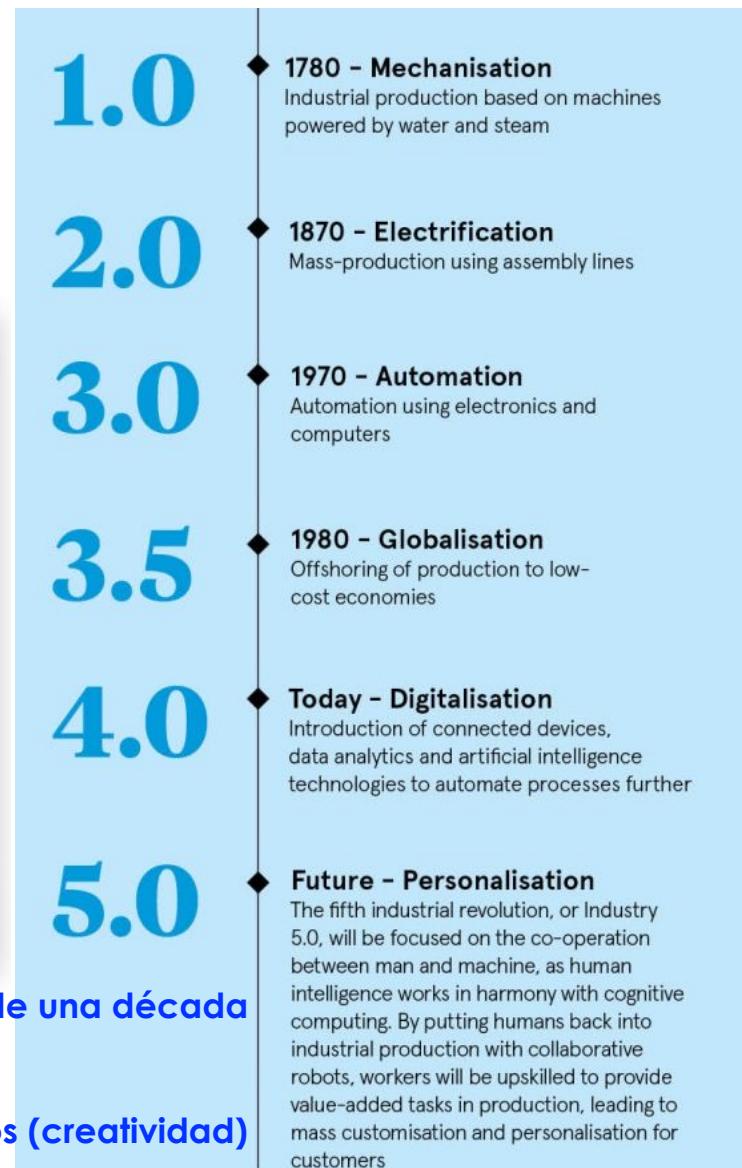
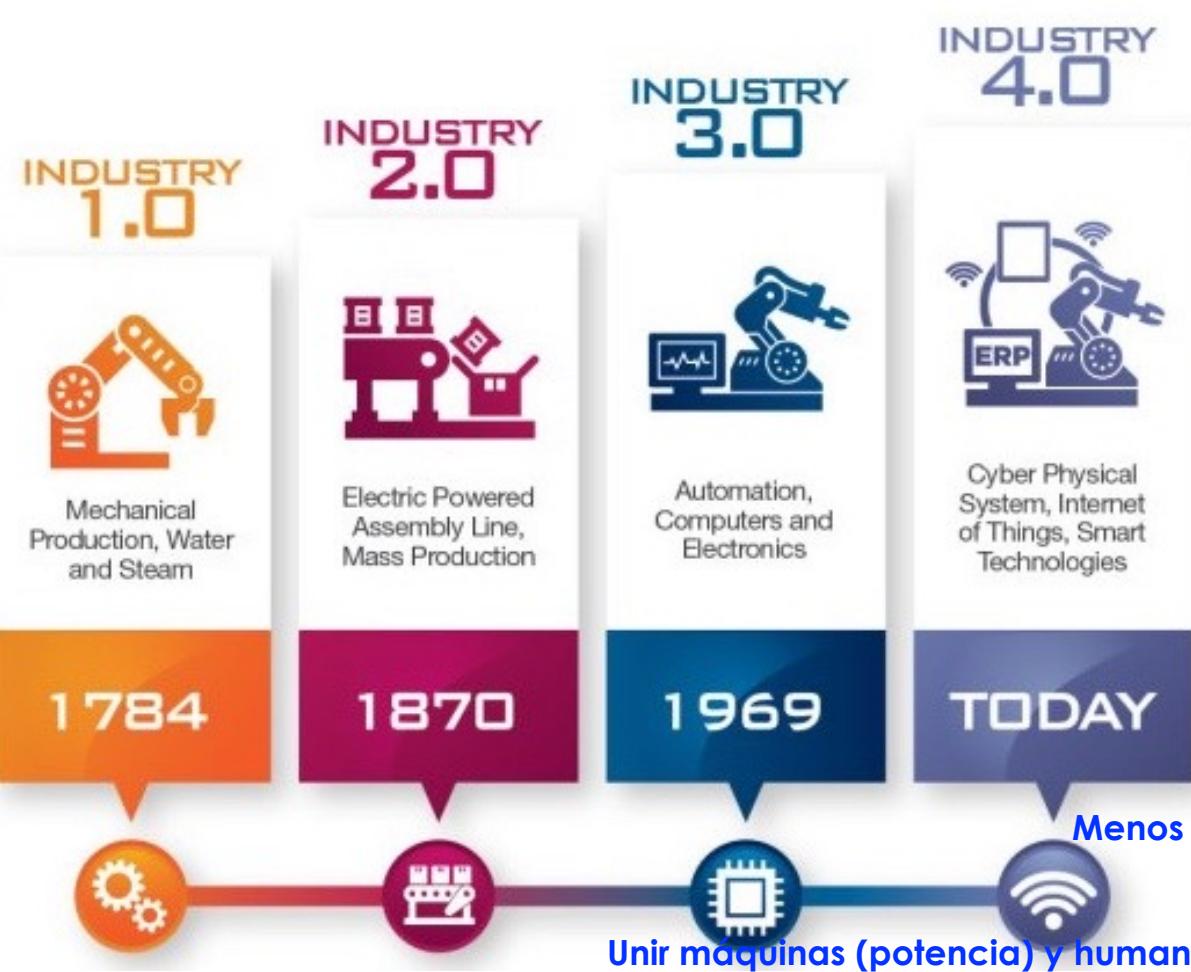


Revisar:

www.timetoast.com/timelines/inteligencia-artificial-6bc8c061-b8f1-4236-a3fe-eb3f2fc49dbb

www.timetoast.com/timelines/historia-y-antecedentes-de-la-ia-e158a78b-e302-41ce-9465-db3bc8ff5894

Revolución industrial

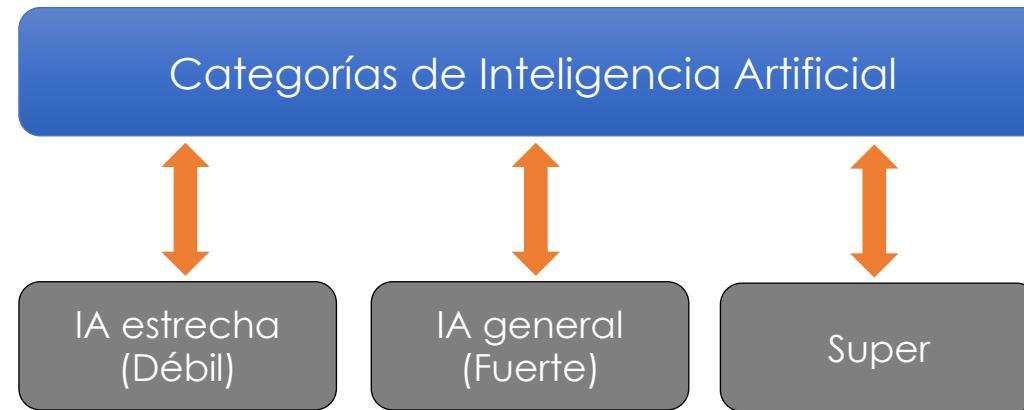


Automatización total
Basada en datos

Mayor humanización
Conectar lo mejor entre
hombre (inteligencia
humana) y máquina
(inteligencia artificial)

Industria 5.0 es una
nueva revolución
tecnológica que
busca la
transformación del
sector **industrial** en
espacios inteligentes
basados en IoT y
sistemas inteligentes.

Inteligencia Artificial Actual



- **IA estrecha.** Carece de la capacidad de entender el contexto. Está enfocada en realizar tareas específicas, por ejemplo, Alexa, Google Assistant, Siri, Cortana, predicciones meteorológicas, control de dispositivos inteligentes y otros.
- **IA general.** Es hacia donde vamos. Comprende el contexto y hace juicios (razonamiento). Con el tiempo, aprende, siendo capaz de tomar decisiones (funcionaría como un cerebro humano). Hasta ahora no se ha logrado, aunque se cree que se podría alcanzar este siglo.
- **Super.** En el futuro lejano, la IA puede llegar a ser superior a los humanos en todos los aspectos. Serían capaces de pensar por sí mismos y operar sin ninguna participación humana. Podría ser el comienzo de una nueva era de innovación.

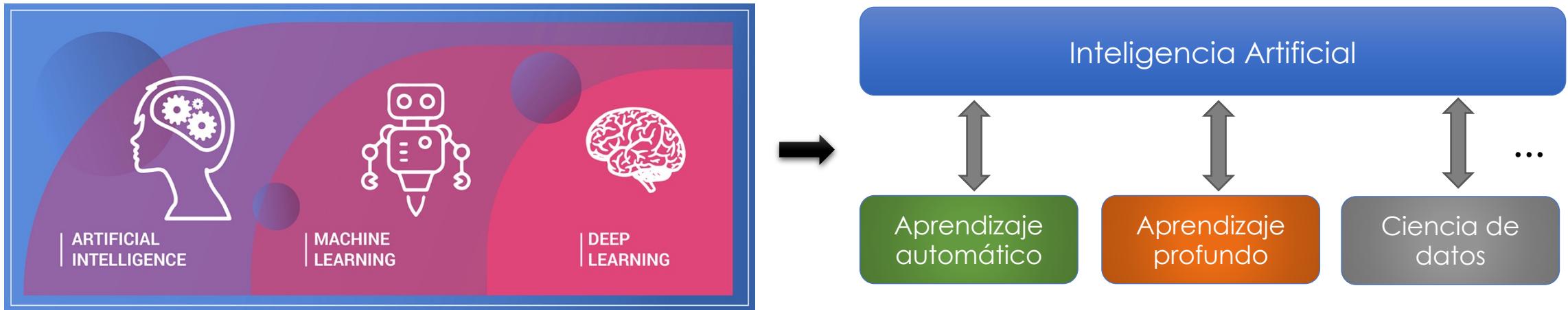
Impacto

En **1978 McCarthy** consideró que para crear una **verdadera IA** se necesitaría el trabajo de "1.7 Einsteins, 2 Maxwells, 5 Faradays y la financiación de 0.3 Proyectos Manhattan, siempre y cuando ese proyecto viniera después de los necesarios descubrimientos conceptuales".

- El propio **McCarthy** mencionó en varias ocasiones que si tuviera que bautizar nuevamente la IA, habría preferido llamarla "**Inteligencia Computacional**".
- Para el 2002, **McCarthy** también mencionó no saber cuánto tiempo será necesario para que la inteligencia de las máquinas alcance el nivel humano, "quizás 50, quizás 500 años, quien sabe".

Inteligencia Artificial Actual

En el contexto actual



- La IA-Actual busca procesar y responder a los datos como lo haría cualquier humano.
- En la actualidad, se están convirtiendo aplicaciones a inteligencia de tipo humana.
- **Se pronostica** que el desarrollo de la IA será el mayor reto tecnológico de la historia
¿pero cuánto puede tardar esto?

Impacto

Nick Bostrom, de la Universidad de Oxford, compara nuestro destino con el de los caballos, cuando fueron sustituidos por los automóviles y los tractores. En 1915, en EE. UU. había alrededor de **26 millones** de equinos. En los años 50 quedaban solo **dos millones**. Los caballos fueron sacrificados como comida para perros.

- En algún momento ¿los robots podrán tomar conciencia propia (control) y determinar una posible destrucción de la humanidad?
- **Caballo que alcanza, gana.**

Inteligencia Artificial

Impacto

Habrá un riesgo para la humanidad, siempre que se pueda construir máquinas pensantes.

- **¿Podrán las máquinas algún día pensar como los humanos?**
- **¿Será una amenaza?**
- **¿Existirá una superinteligencia artificial?** Se convertirá exponencialmente más inteligente.

No hay un argumento científico que lo apoye. Analizar: Singularidad y Trascendencia.

Actividad en clase

Lectura: Invierno de la Inteligencia Artificial



The screenshot shows the BBC News website's header in Spanish. At the top left is the BBC logo. To its right is a "Menú" button. Further right is a search bar with a magnifying glass icon. Below the search bar is a red navigation bar with the word "NEWS" and "MUNDO" separated by a vertical line. Below this are several menu links: Noticias, América Latina, ¿Hablas español?, Internacional, Economía, Tecnología, Ciencia, Salud, Cultura, and Más. The entire header is set against a white background.

Qué es el "invierno de la inteligencia artificial" y por qué hay expertos que creen que estamos acercándonos a uno

Sam Shead
Reportero de Tecnología de BBC News

⌚ 14 enero 2020

    Compartir

URL: <https://www.bbc.com/mundo/noticias-51097189>

Actividad en clase

Lectura: El invierno de la Inteligencia Artificial ¿cada vez más cerca?



El invierno de la inteligencia artificial, ¿cada vez más cerca?

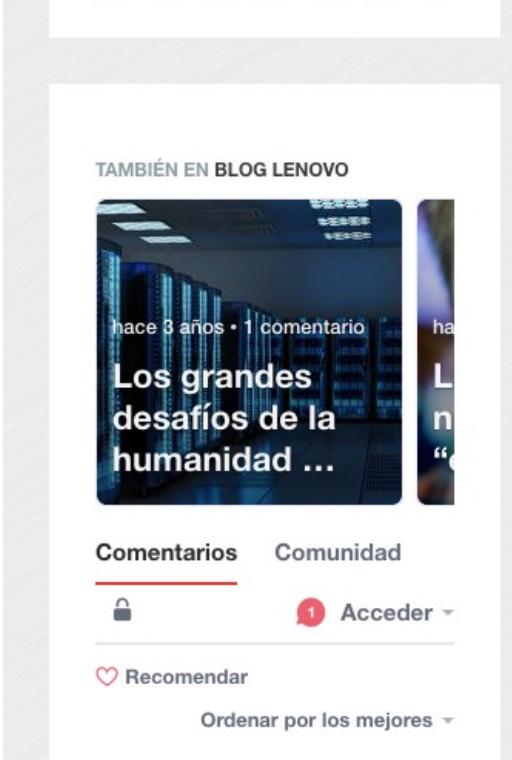
por Marcos Martínez en General, Vida tecnológica / 27 de febrero de 2020

f t p

IS WINTER COMING?

Lenovo

El fenómeno **invierno de la inteligencia artificial** es un periodo de tiempo sin apenas inversión en el campo. Kai-Fu Lee, en su libro 'Superpotencias de la inteligencia artificial' (2020), lo define como el momento en que "la decepcionante falta de resultados prácticos conducía a importantes recortes de financiación".



TAMBIÉN EN BLOG LENOVO

hace 3 años • 1 comentario

Los grandes desafíos de la humanidad ...

Comentarios Comunidad

Acceder

Recomendar

Ordenar por los mejores

URL: <https://www.bloglenovo.es/el-invierno-de-la-inteligencia-artificial-cada-vez-mas-cerca>

Ensayo 1

Elaborar un breve ensayo, de dos hojas, sobre **singularidad y trascendencia**.

Fecha de entrega: martes 14 de septiembre de 2021

Hora: antes de las 11:00 horas

Formato: digital (LaTex), subir al la carpeta compartida los archivos 'pdf' y 'tex'.

Fuentes sugeridas

<https://cs.nyu.edu/~davise/papers/singularity.pdf>

<https://nickbostrom.com/papers/survey.pdf>

<https://www.mdpi.com/2078-2489/10/2/73/htm>

<https://jcer.com/index.php/jcj/article/view/150/161>



Universidad Nacional Autónoma de México
Facultad de Ingeniería

Tecnologías emergentes

Guillermo Molero-Castillo

guillermo.molero@ingenieria.unam.edu

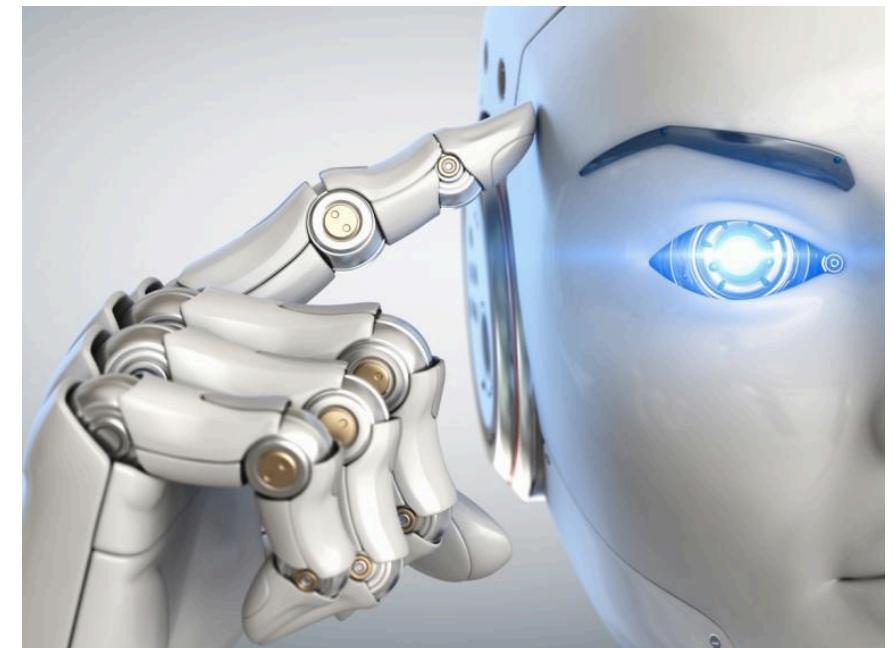
Septiembre, 2021

Inteligencia Artificial

Impacto

IA actual es ahora **más inteligente** y **menos artificial**.

- Se está expandiendo y transformando las operaciones empresariales.
- El panorama actual es la búsqueda de la especialización (STEM –ciencia, tecnología, ingeniería y matemáticas–).



Inteligencia Artificial

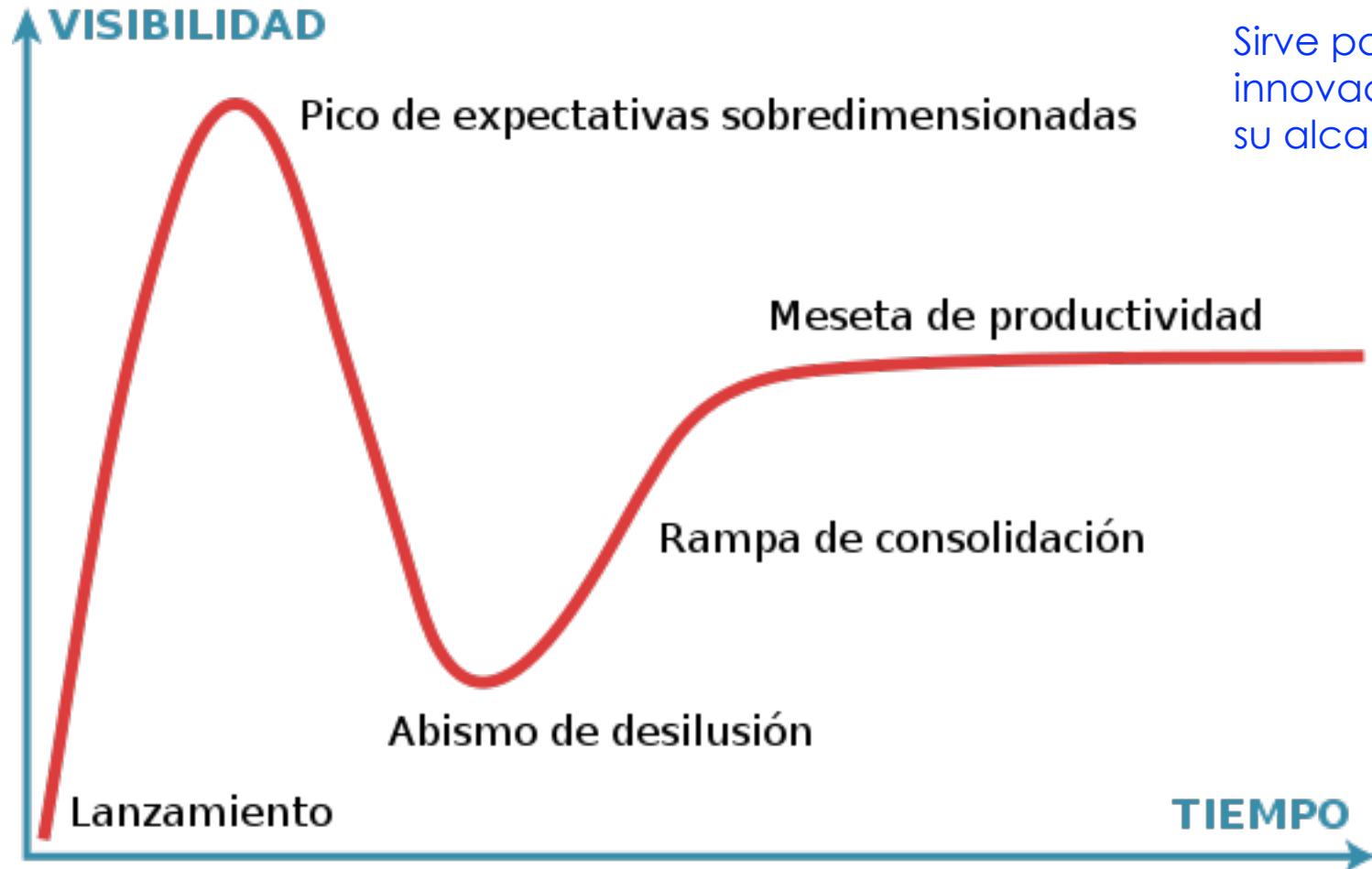
Impacto

En la actualidad, se convive con aplicaciones de IA. Es probable que se haya utilizado IA de alguna forma.

- Asistentes virtuales: **Siri, Alexa, Cortana, o Google Assistant.**
- Transporte: **Uber, Cabify o Didi**
- Servicios de **Watson de IBM**
- **DeepMind de Google.**

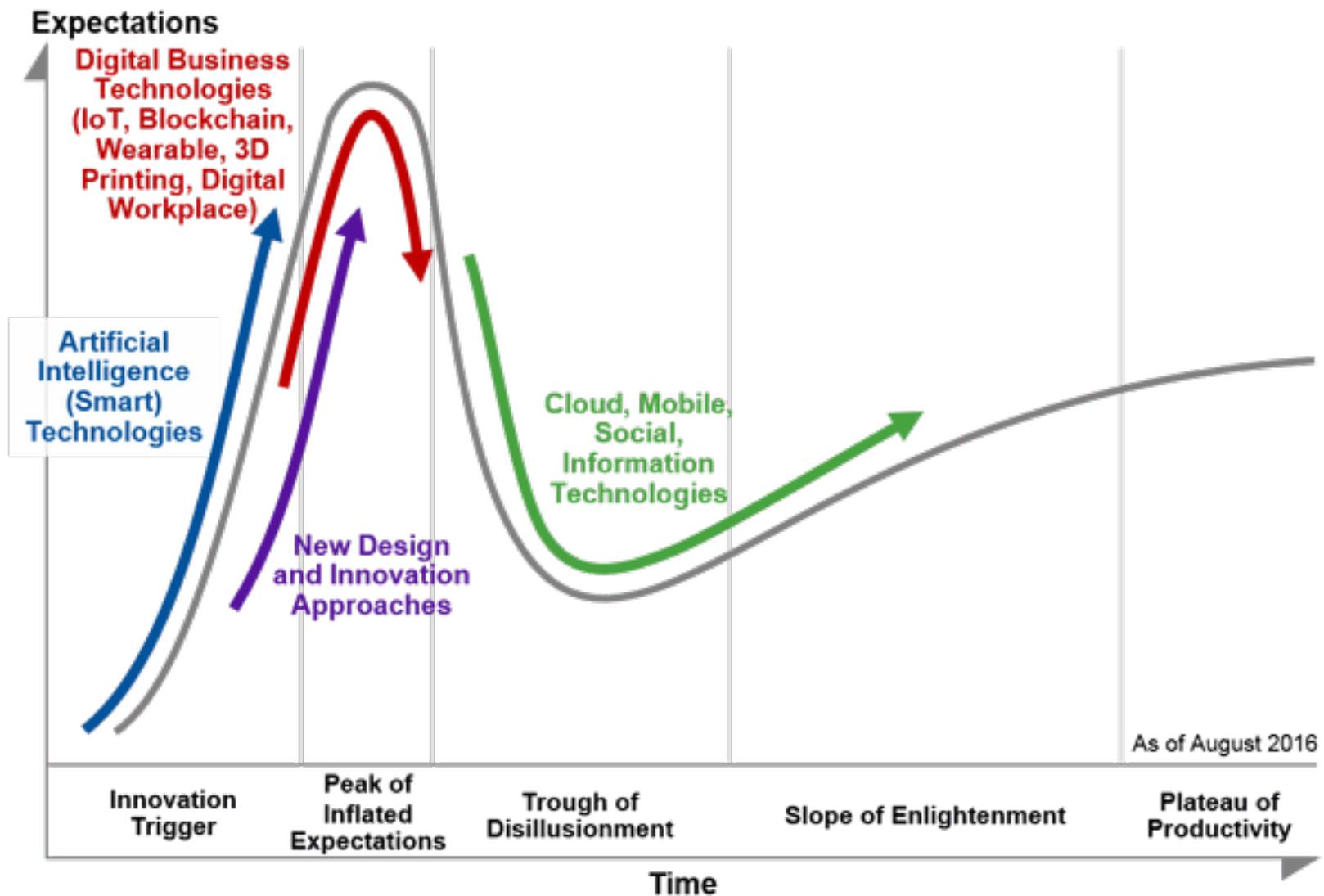
Tendencia Inteligencia Artificial

Tendencia (Tecnologías emergentes)

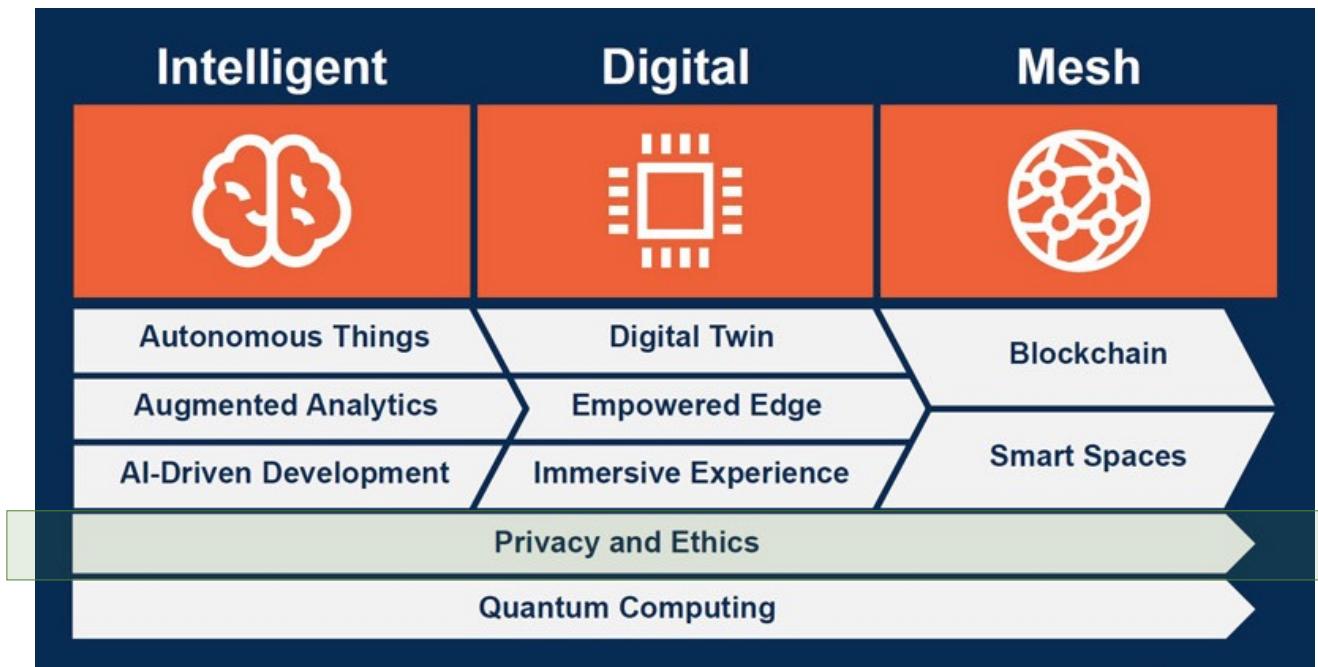


Sirve para rastrear tendencias e innovaciones con el fin de identificar su alcance, estado y valor.

Tendencia de la Inteligencia Artificial



Tendencia de la Inteligencia Artificial, 2019



Inteligente

La IA y la analítica de datos está en todas las tecnologías y crea nuevas categorías.

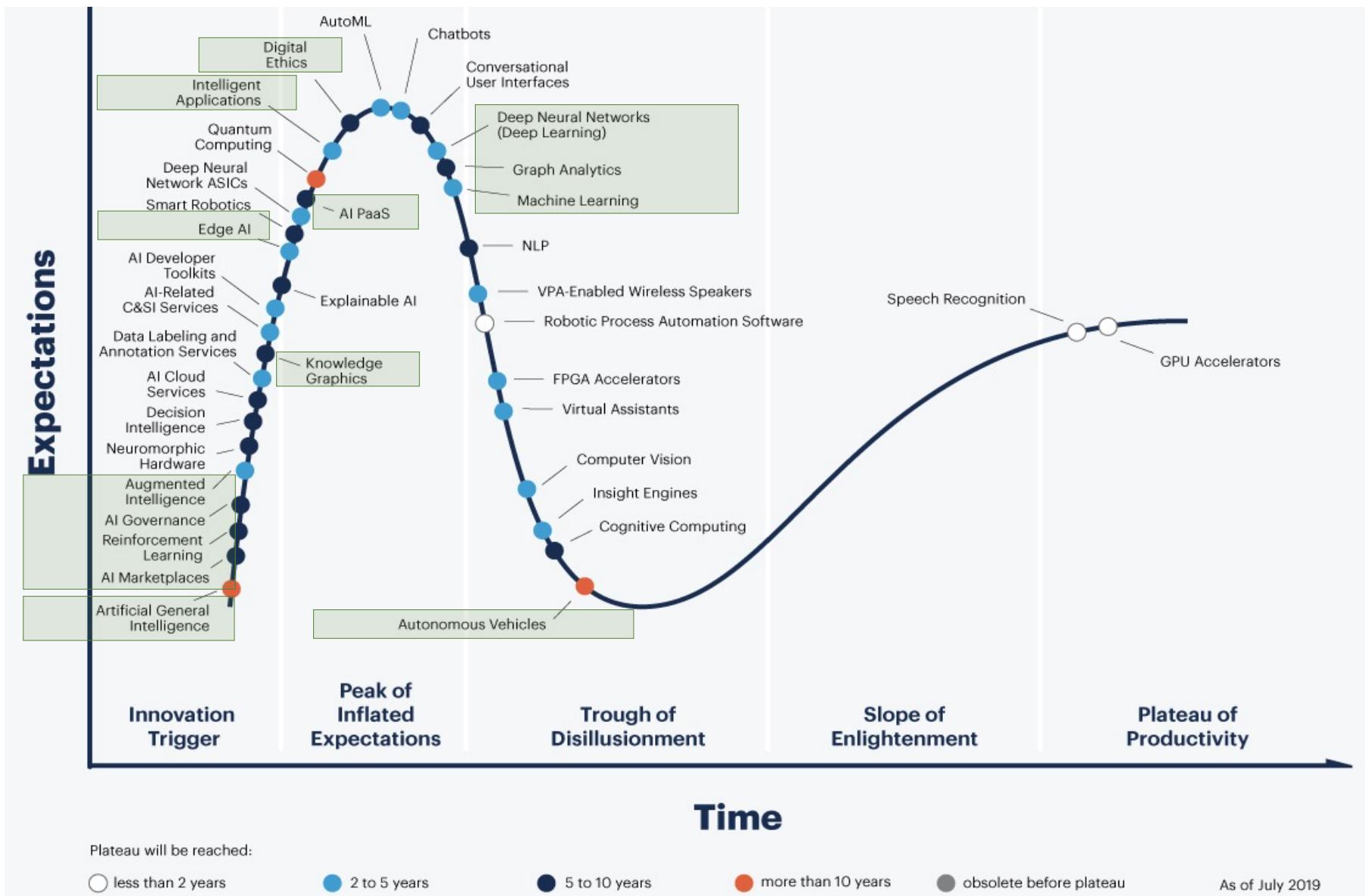
Digital

Combinación del mundo digital y físico para crear un mundo inmersivo.

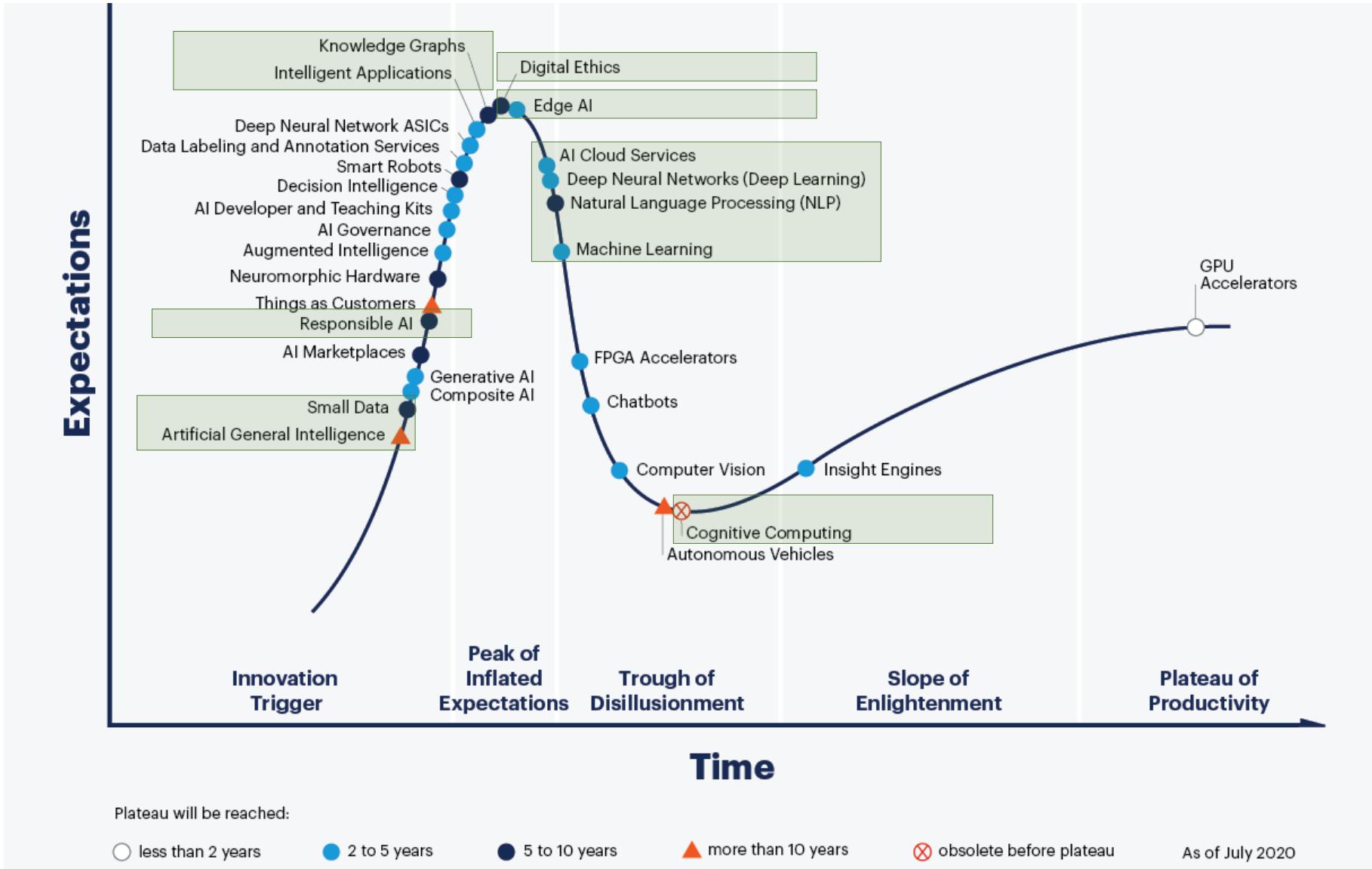
Malla

Exploración de conexiones entre personas, dispositivos, contenido, negocios, y servicios en expansión.

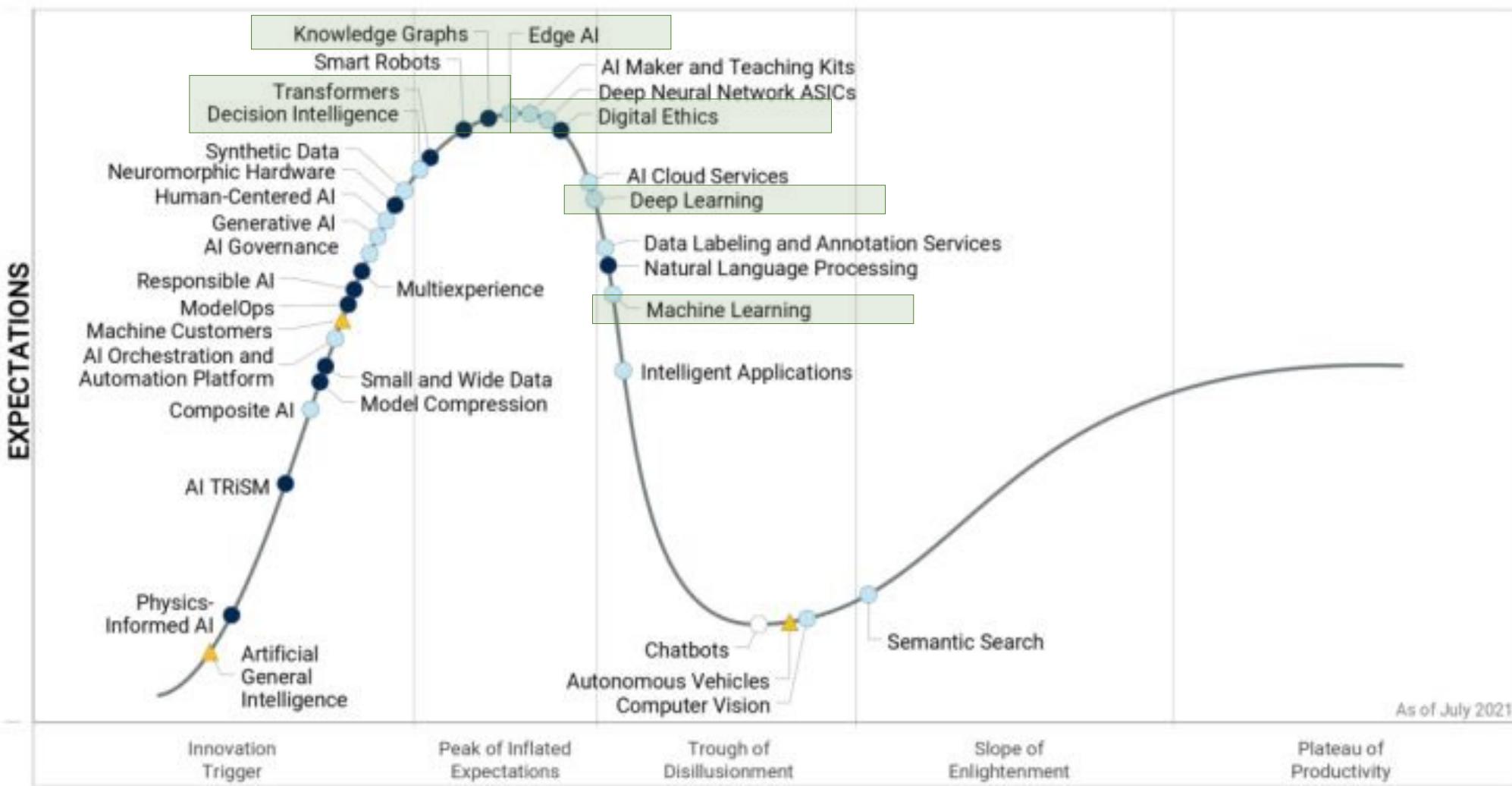
Ciclo para la Inteligencia Artificial, 2019



Ciclo para la Inteligencia Artificial, 2020



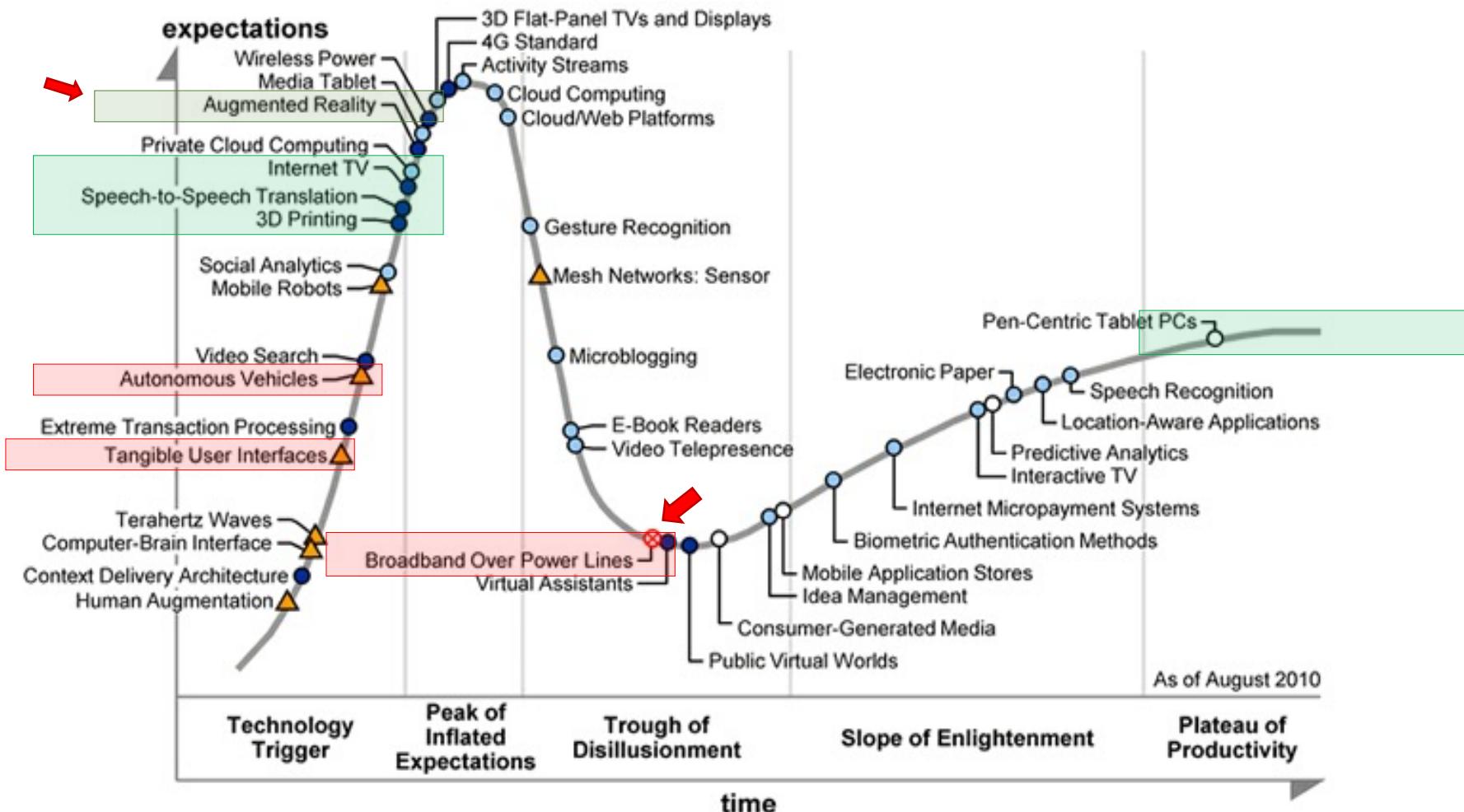
Ciclo para la Inteligencia Artificial, 2021



Tecnologías Emergentes Antecedentes

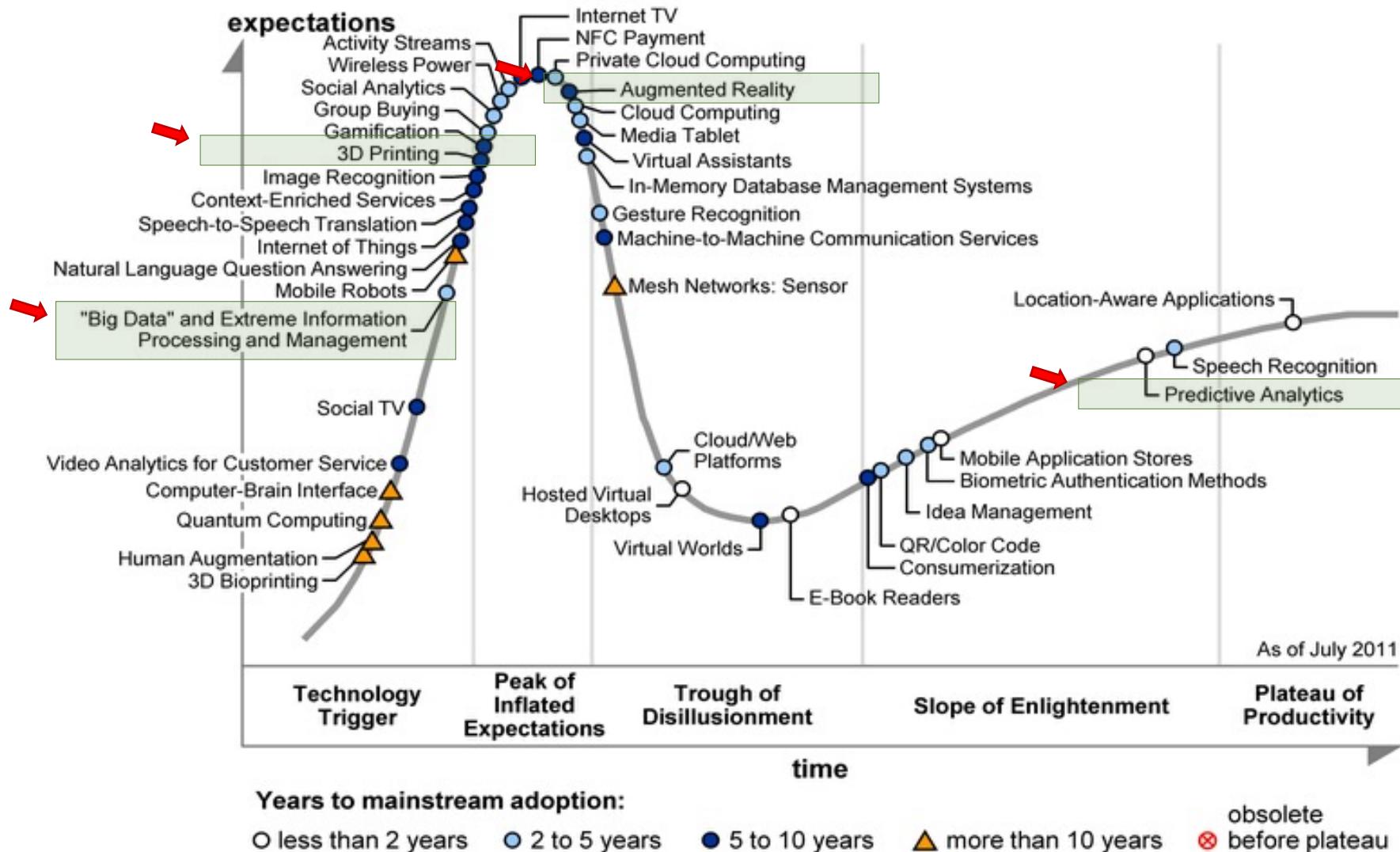
Tendencia (Tecnologías emergentes)

2010



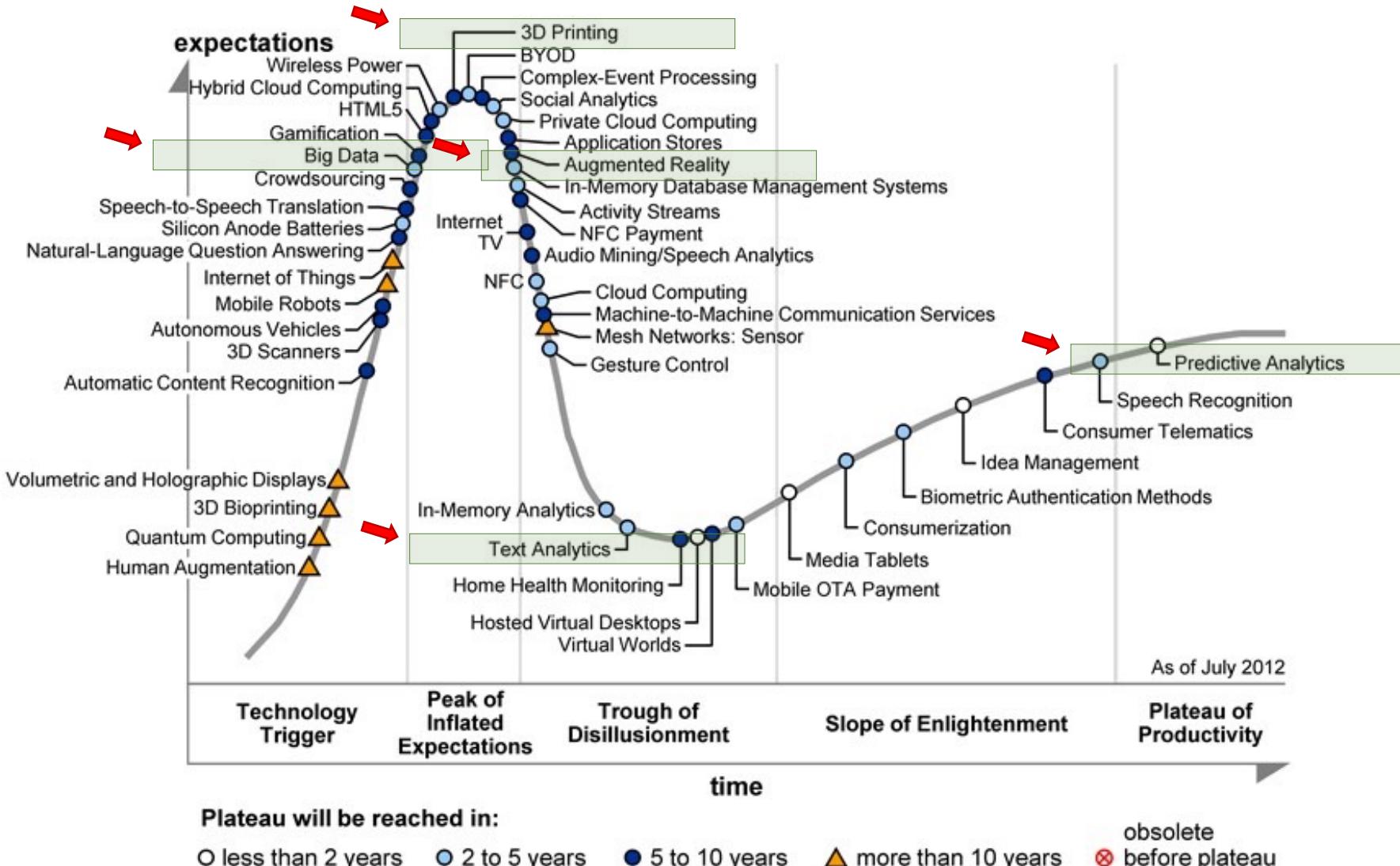
Tendencia (Tecnologías emergentes)

2011



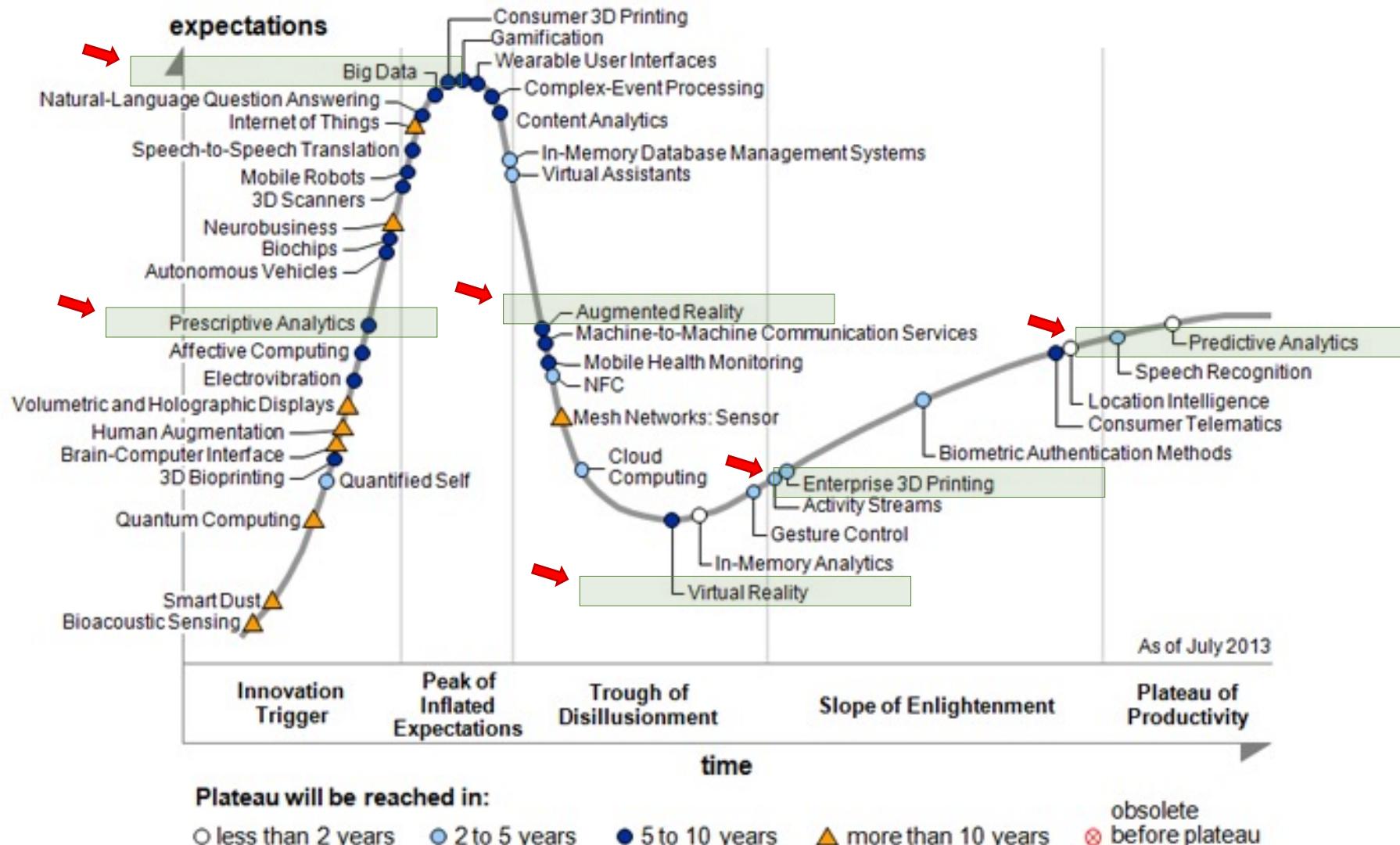
Tendencia (Tecnologías emergentes)

2012



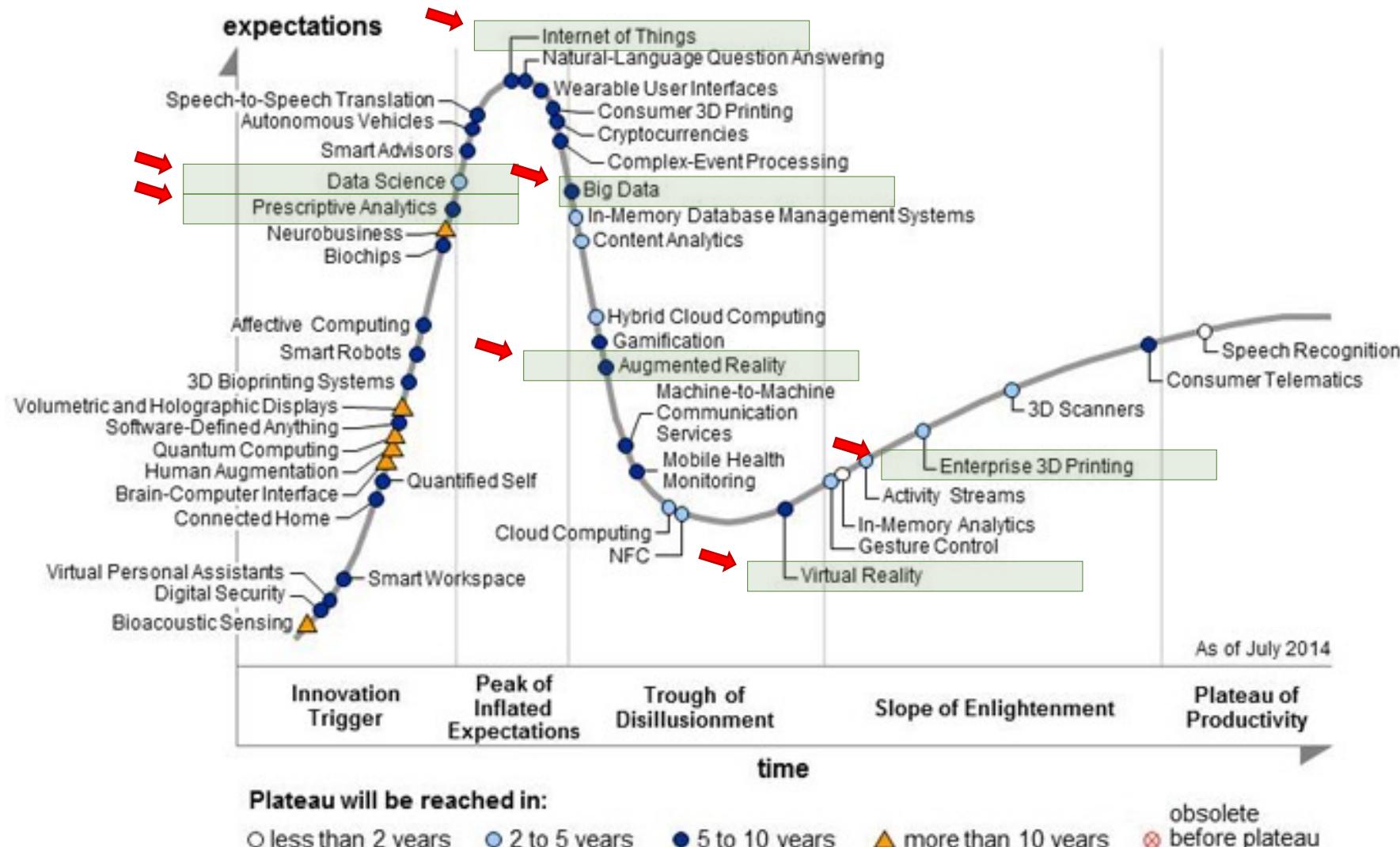
Tendencia (Tecnologías emergentes)

2013



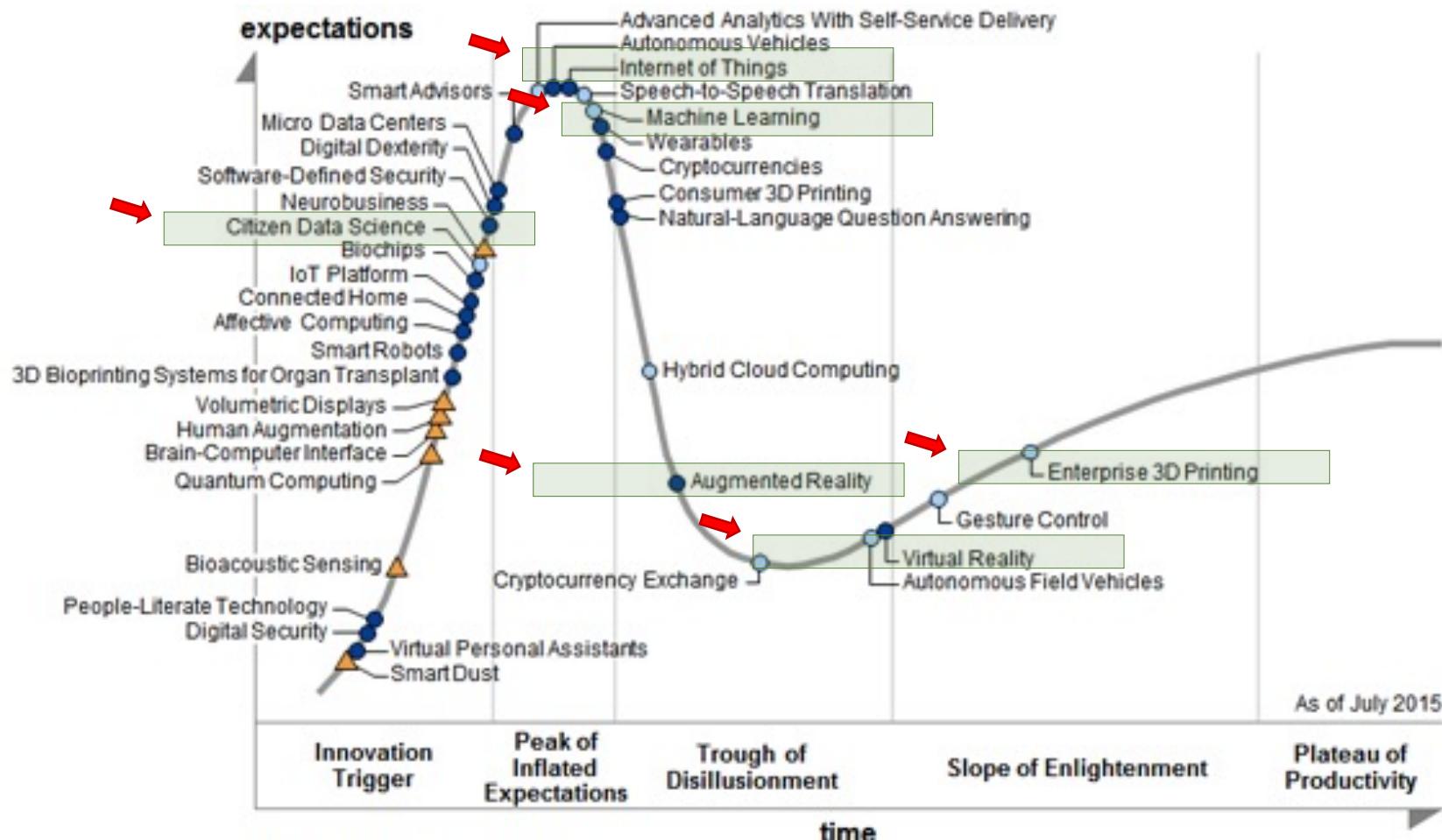
Tendencia (Tecnologías emergentes)

2014



Tendencia (Tecnologías emergentes)

2015



Plateau will be reached in:

○ less than 2 years

● 2 to 5 years

● 5 to 10 years

▲ more than 10 years

obsolete
◎ before plateau

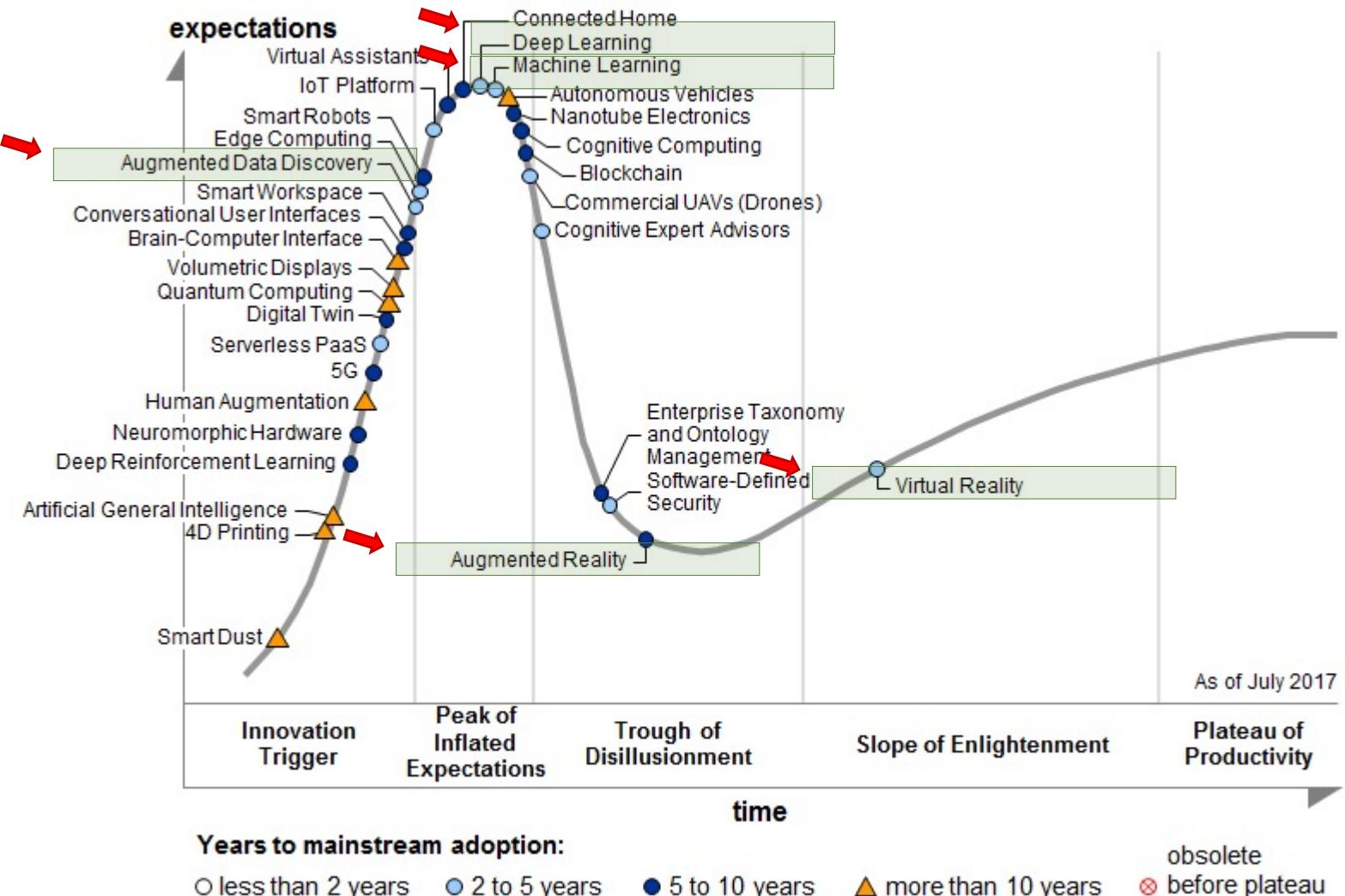
Tendencia (Tecnologías emergentes)

2016



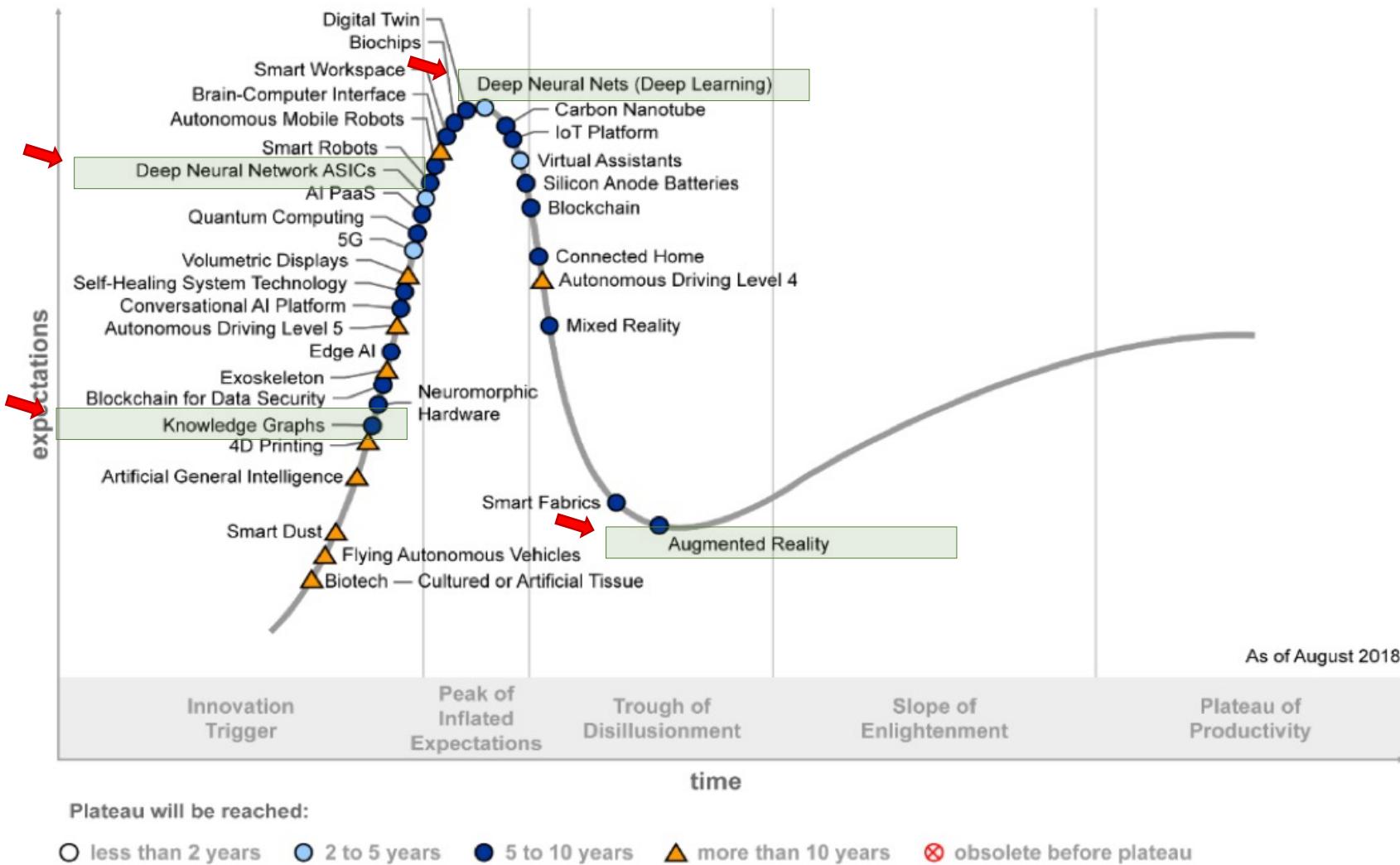
Tendencia (Tecnologías emergentes)

2017



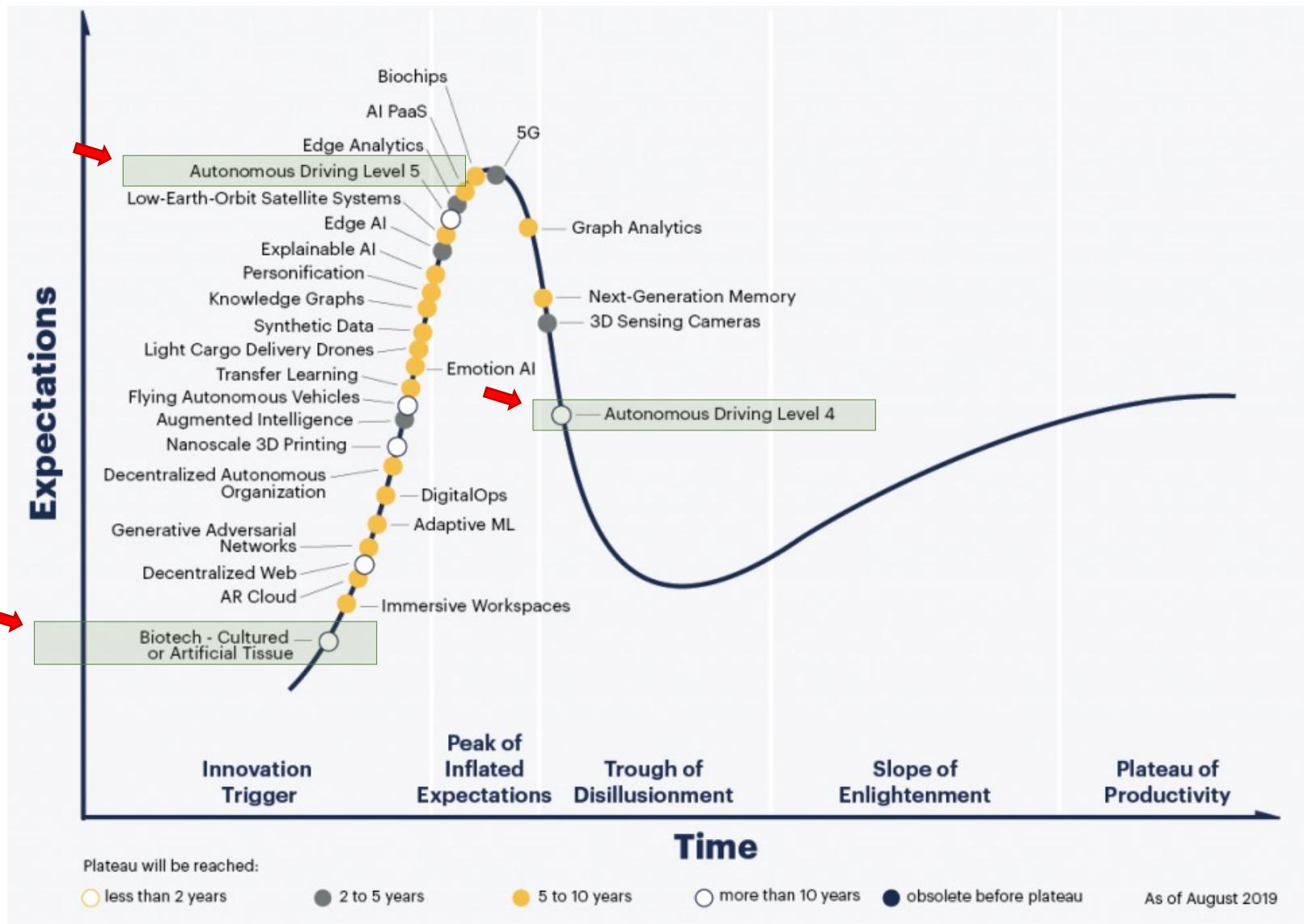
Tendencia (Tecnologías emergentes)

2018



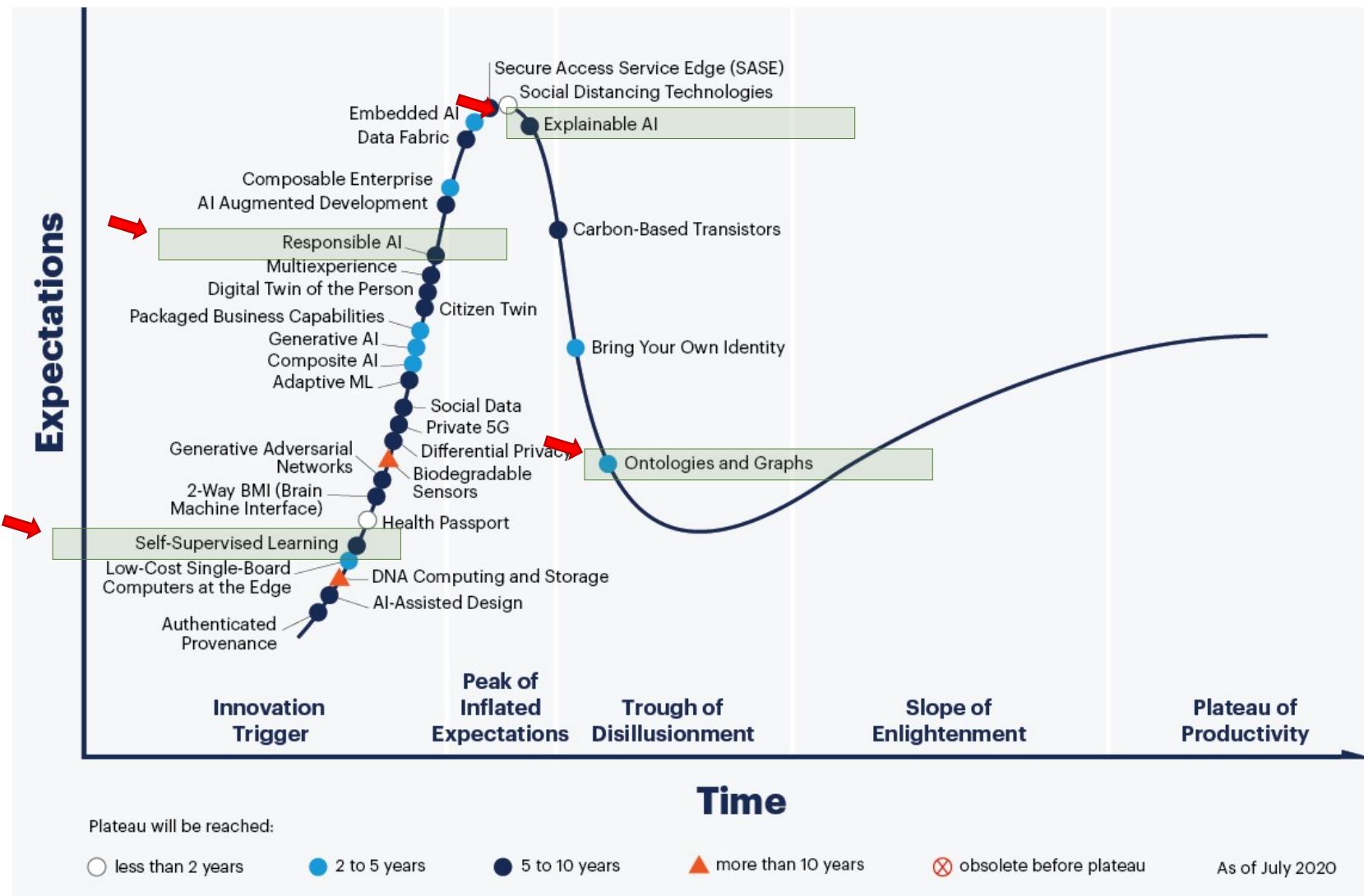
Tendencia (Tecnologías emergentes)

2019



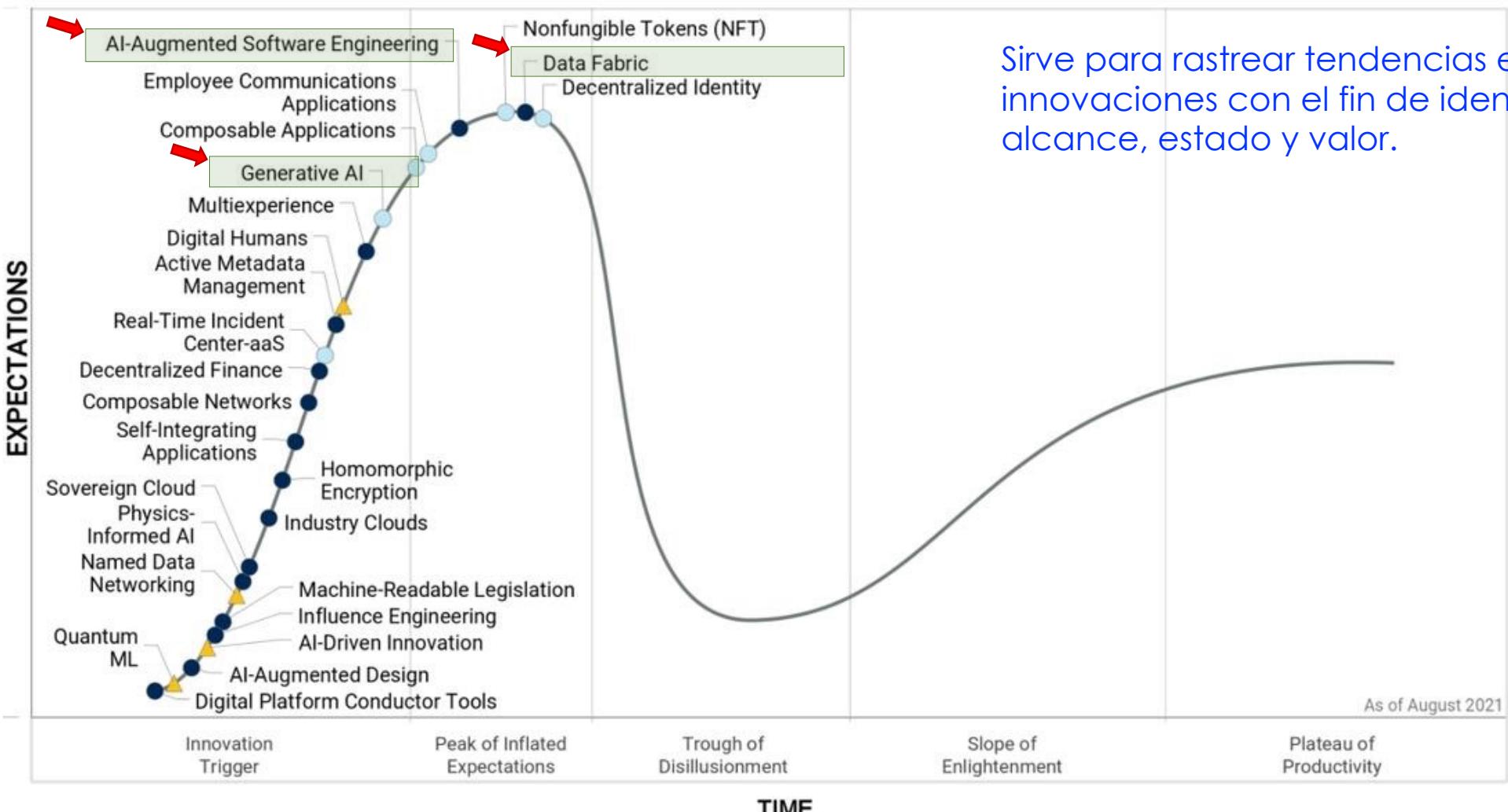
Tendencia (Tecnologías emergentes)

2020



Tendencia (Tecnologías emergentes)

2021



Sirve para rastrear tendencias e innovaciones con el fin de identificar su alcance, estado y valor.

Plateau will be reached: ○ < 2 yrs. ● 2-5 yrs. ■ 5-10 yrs. ▲ >10 yrs. ✕ Obsolete before plateau

Ética Digital

Reporte preliminar de recomendaciones sobre la ética de la Inteligencia Artificial - UNESCO

PRELIMINARY REPORT ON THE FIRST DRAFT OF THE RECOMMENDATION ON THE ETHICS OF ARTIFICIAL INTELLIGENCE

This preliminary report was prepared in accordance with Article 10.1 of the Rules of Procedure concerning recommendations to Member States and international conventions covered by the terms of Article IV, paragraph 4, of the Constitution, on the draft Recommendation on the Ethics of Artificial Intelligence.

INTRODUCTION

1. Artificial intelligence¹ (AI) is one of the central issues of the era of converging technologies with profound implications for humanity, cultures, societies and the environment. AI is already having impact across all sectors, and it is already transforming the future of education, natural and social sciences, culture and communication. These areas, along with the higher goal of promoting the respect of human rights and human dignity, along with a culture of peace, are core to UNESCO's mandate.
2. As with previous technological revolutions, AI has the potential to transform the future of humanity for the better and in favour of sustainable development. However, it can also bring downside risks and challenges, in particular derived from malicious utilization of the technology, that affects human rights, or from the fact that highly complex AI systems can widen substantially the already existing high inequalities and divides. In fact, these technologies, and the digital transformation have a "winner takes all dynamic" that needs to be addressed. The impact of the technology will depend on the way humanity frames it and masters it, and on the way it prioritizes the goal of leaving no one behind. This is where UNESCO's role in promoting social inclusion and fighting inequalities, is of paramount importance at the global level. In order to sketch possible scenarios and unlock AI's potential to grasp development opportunities, while managing risks, it is important to develop a more comprehensive understanding of how societies are transformed by disruptive technologies, such as AI.

Noviembre de 2019: 40a reunión de la Conferencia General de la UNESCO se decide preparar un instrumento normativo internacional sobre la ética de la inteligencia artificial, en forma de recomendación.

Diciembre de 2020: Se invita a los Estados Miembros a enviar sus comentarios y observaciones sobre este texto a la Secretaría de la UNESCO.

Iniciativas de uso responsable de la Inteligencia Artificial - OECD

Primer estándar internacional para una IA confiable



OECD Legal Instruments

Login

[Home](#) [General information](#) [Full list](#) [Advanced search](#) [Adherences](#) [Key figures](#)

→ [FR](#)



Garantizar que la IA se utilice de manera responsable, respetando los derechos humanos y los valores democráticos.

OECD/LEGAL/0449

Adopted on:
21/05/2019

Text

Background information

Related document(s)

Unofficial translations

Committee(s)

Date(s)/Reference(s)

Adherents

Recommendation of the Council on Artificial Intelligence

In force Recommendation Science and Technology

Artificial Intelligence (AI) technologies and tools play a key role in every aspect of the COVID-19 crisis response. This Recommendation provides a set of internationally-agreed principles and recommendations that can promote an AI-powered crisis response that is trustworthy and respects human-centred and democratic values. For further information on this Recommendation and its relevance to COVID-19 response and recovery, see the background information below.

THE COUNCIL,

HAVING REGARD to Article 5 b) of the Convention on the Organisation for Economic Co-operation and Development of 14 December 1960;

Lectura: Principales tendencias tecnológicas para 2021

TENDENCIAS

Nueve tendencias tecnológicas para 2021

CIO Published 4 meses ago on octubre 20, 2020
By Redacción CIO México



ADVERTISEMENT

VIDEOS

PRINCIPAL / 12 meses ago **VIDEO: Gestión de credenciales privilegiadas, la última línea de defensa**

PRINCIPAL / 12 meses ago **VIDEO: La protección de datos y entendimiento de los procesos como pilares en la virtualización**

PRINCIPAL / 12 meses ago **VIDEO: Deception Technology, una opción para enfrentar los peligros digitales**

ADVERTISEMENT

Enlace: <https://cio.com.mx/nueve-tendencias-tecnologicas-para-2021>

Artificial Intelligence Engineering (Ingeniería de IA)

8. Ingeniería de inteligencia artificial

DataOps, ModelOps y DevOps son los pilares de ingeniería de la inteligencia artificial. Una sólida estrategia en esta materia facilitará el rendimiento, la escalabilidad, la interpretabilidad y la fiabilidad de los modelos de inteligencia artificial, al tiempo que proporcionará un mayor valor a las inversiones realizadas en esta tecnología, según Gartner.

Se trata, apuntan desde la firma de investigación, de una tendencia importante teniendo en cuenta que solo el 53% de los proyectos de inteligencia artificial pasan de la fase de prototipado a la de producción. Y, según los analistas de la consultora, “el camino hacia la producción de la inteligencia artificial significa recurrir a la ingeniería de esta tecnología, una disciplina centrada en la gobernanza y la gestión del ciclo de vida de una amplia gama de modelos operativos de estas herramientas, como el aprendizaje automático o los gráficos de conocimiento”.

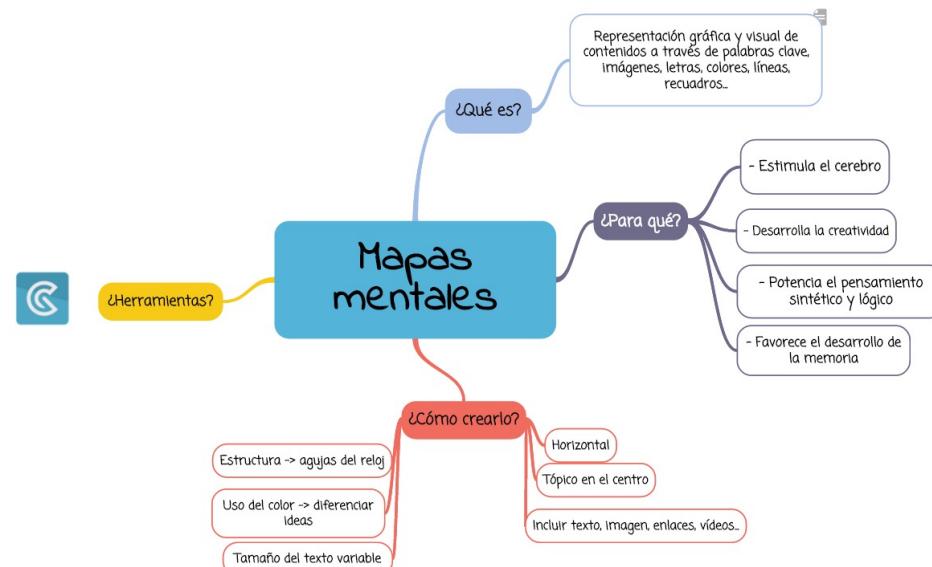
Tarea 1

Elaborar un mapa conceptual sobre Ingeniería de Inteligencia Artificial.

Fecha de entrega: martes 21 de septiembre de 2021

Hora: antes de las 11:00 horas

Formato: libre, subir a la carpeta compartida el archivo en formato 'pdf'.



Fuentes de apoyo:

- <https://algorithmia.com/blog/what-is-artificial-intelligence-engineering>
- <https://www.sei.cmu.edu/our-work/artificial-intelligence-engineering>



Universidad Nacional Autónoma de México
Facultad de Ingeniería

Aprendizaje

Guillermo Molero-Castillo

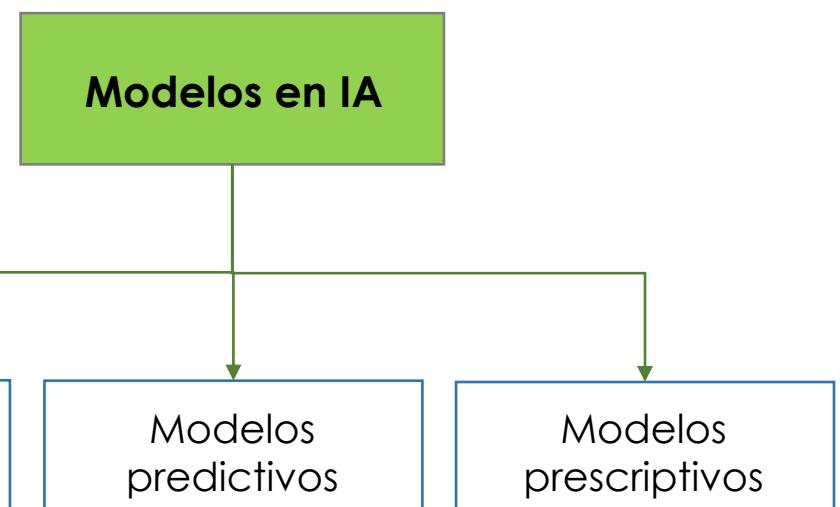
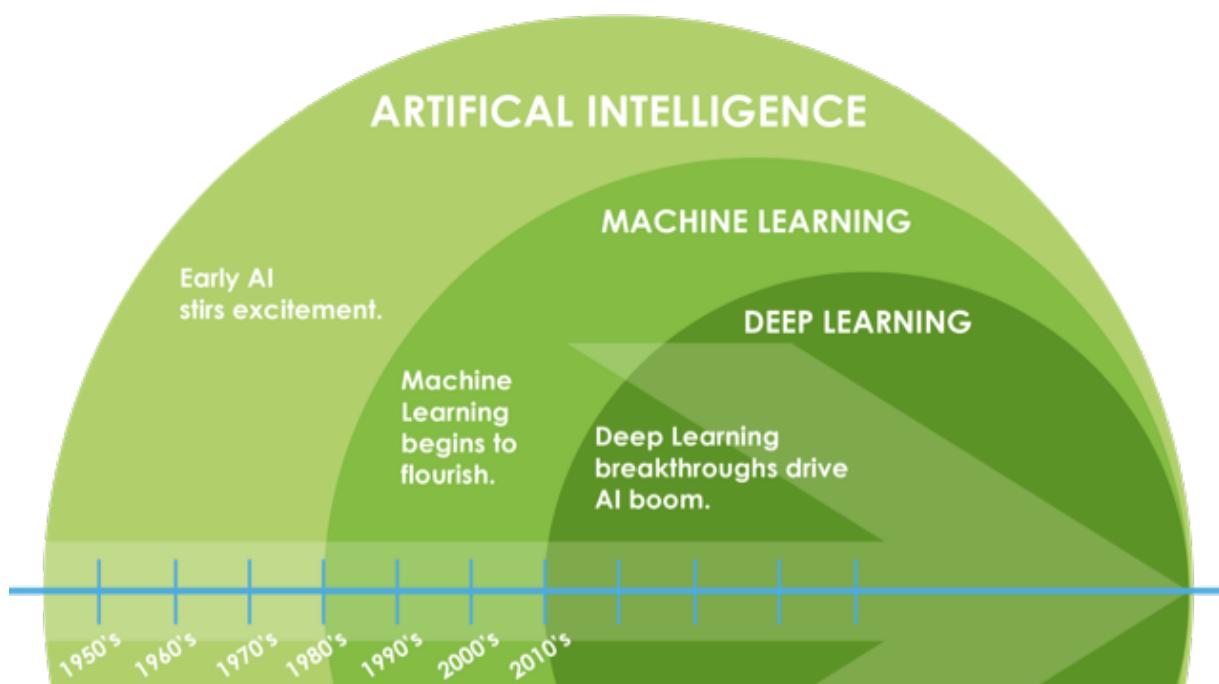
guillermo.molero@ingenieria.unam.edu

Septiembre, 2021

Modelos Inteligentes

Modelar comportamientos inteligentes

Los modelos buscan patrones en los datos para establecer conclusiones, como lo harían las personas.



¿Qué sucede?

¿Qué podría pasar?

¿Qué se debe hacer?

Nueva aproximación

Alexander Wissner-Gross propuso el 2013 una nueva aproximación para describir la inteligencia de manera conceptual: ‘Una nueva ecuación para la inteligencia’.

$$\mathbf{F} = \mathbf{T} \nabla(S\mathbf{T})$$

Donde:

- La inteligencia es una fuerza (\mathbf{F}) que actúa para maximizar una futura ‘acción’ con una fuerza (\mathbf{T}).
- $\nabla(S\mathbf{T})$ es la diversidad de posibilidades (S , entropía), en una determinada dirección (∇ , gradiente) y en un tiempo futuro (\mathbf{T}).

En concreto, cuanta más inteligencia, se tendrá más capacidad de predecir lo que va a suceder en un futuro, y por ende, más posibilidades de elección para actuar de la manera correcta.

Nueva aproximación

Desde punto de vista de la física, esto se explica de la siguiente manera:

$$\mathbf{F} = T \nabla(S\mathbf{T})$$

- La inteligencia es una fuerza (**F**) que es igual a una energía con capacidad de realizar una acción (trabajo), representado por una fuerza (**T**), que apunta en una dirección (**∇**), hacia la mayor cantidad de posibilidades, representado por la entropía (**S**, –grados de libertad–), en un determinado tiempo futuro (**T**).
- Con esta formalización, Wissner-Gross (2013) propone una comprensión de la inteligencia humana para construir una inteligencia artificial, que debe ser vista como un fenómeno físico, que intenta maximizar la futura libertad de acción.

Inteligencia

Alex Wissner-Gross: Una nueva ecuación para la inteligencia

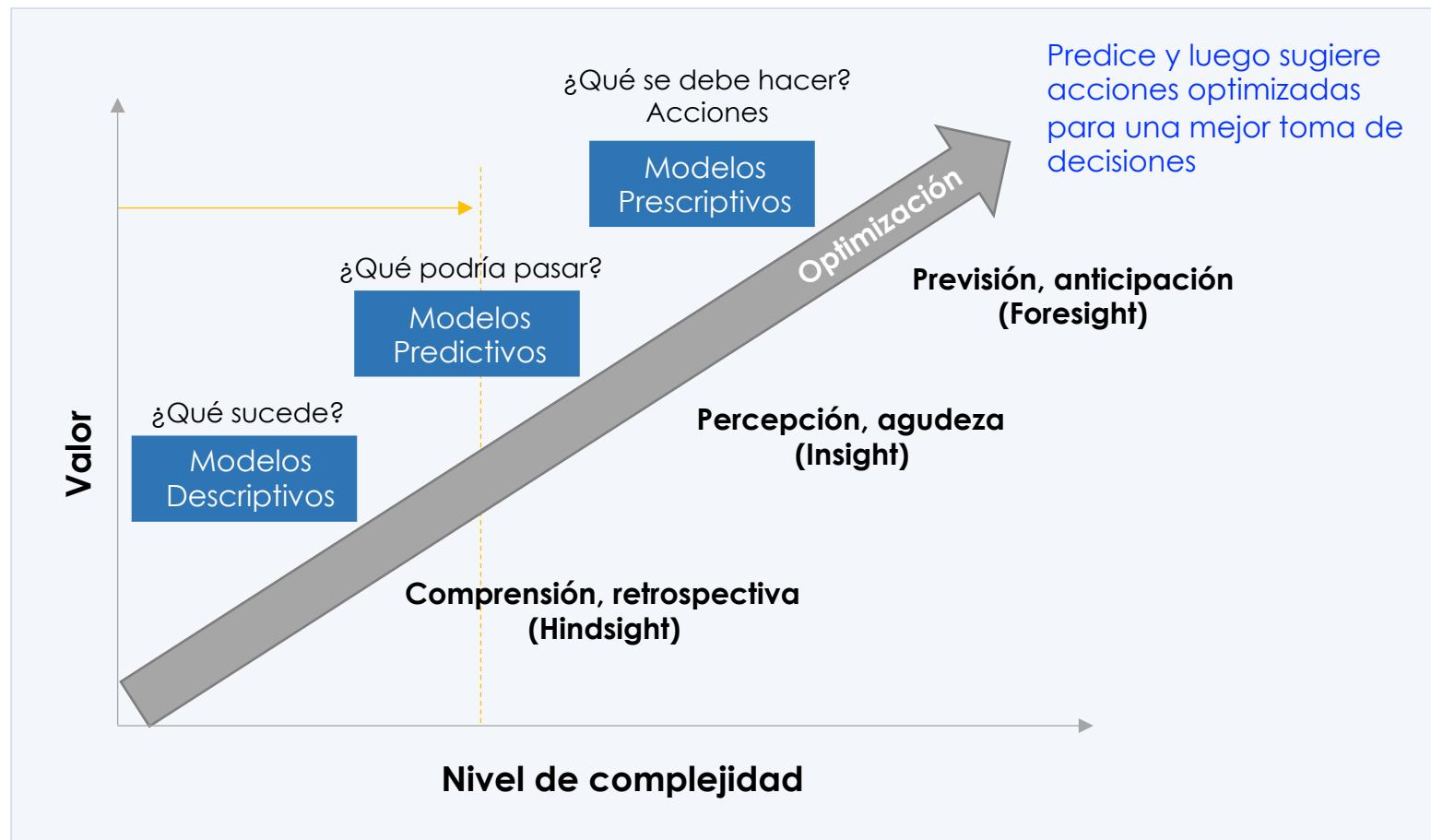


Información ampliada

https://www.youtube.com/watch?v=ue2ZEmTJ_Xo

Inteligencia Artificial

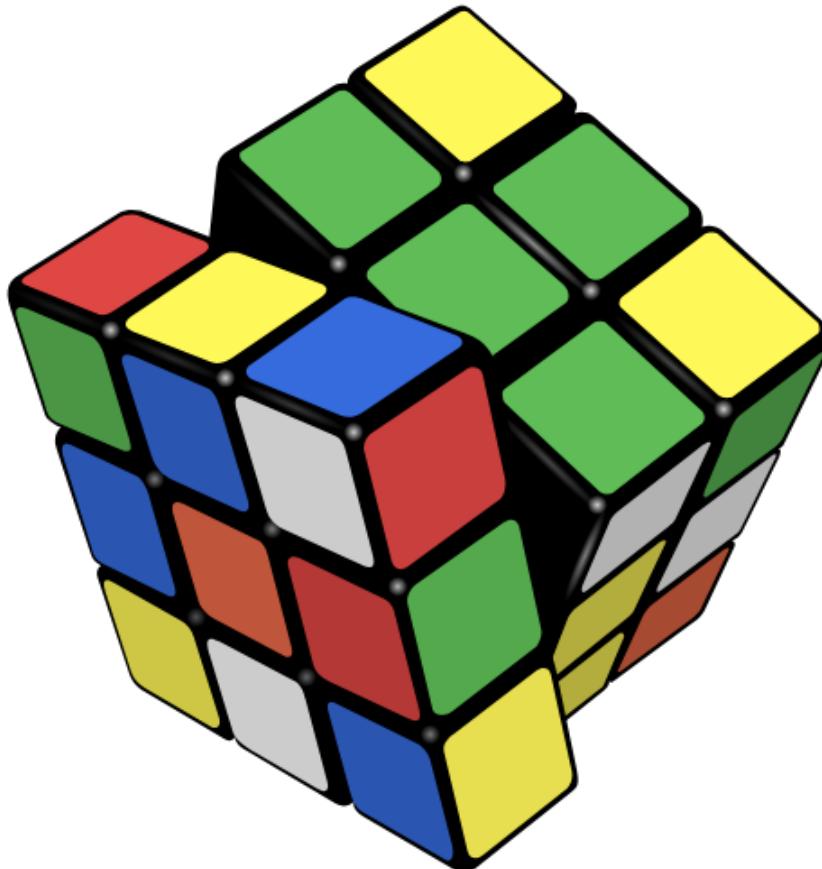
Complejidad



Inteligencia Artificial

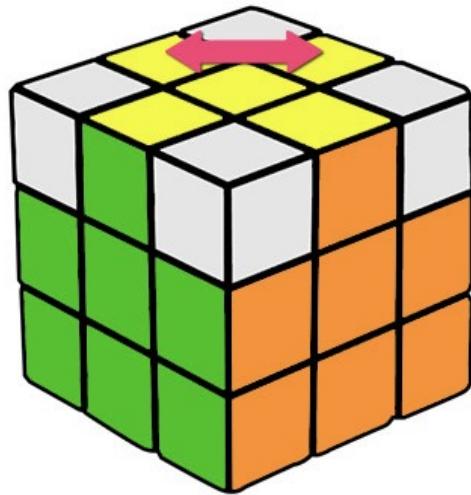
1. Aprendizaje

Aprendizaje

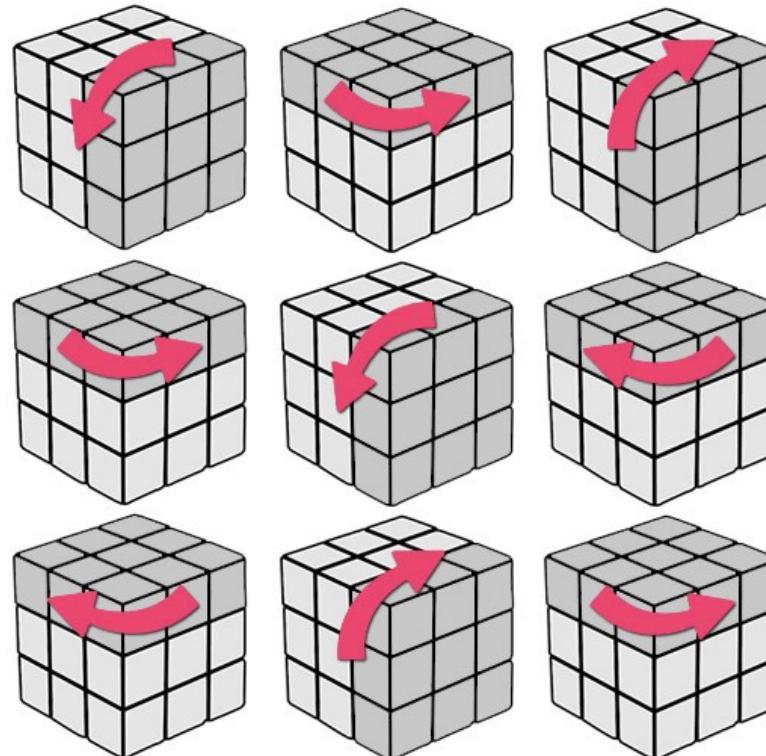


¿Qué hace falta para resolver el cubo de Rubik?

Aprendizaje



Conocimiento



- Conocimiento es algo que se hace.
- El conocimiento se gestiona (muchas veces sin darnos cuenta).

Aprendizaje

El proceso de producir conocimiento es el aprendizaje



Proceso de aprendizaje (horas de aprendizaje)

Conocimiento (se aplica durante un problema)

ANTES

DURANTE

Van de la mano

Aprendizaje

¿Qué pasa con las personas?

- El conocimiento es intangible.
- Sistematizar el conocimiento (experiencia).
- Esta experiencia es reutilizable en el tiempo.



Aprendizaje

¿Qué pasa con las organizaciones, cómo aprenden?

- Sistematizar el conocimiento (experiencia).
- Acumulan.
- Reutilizan.
- ¿Dónde está la memoria de una organización?



Aprendizaje

¿Dónde está el cerebro de una organización?



2. Algoritmos y máquinas

Algoritmos

Don Algoritmo (Erudito Persa)

- ***Al-Khowârizmî, Al-Jwarizmi o Al-Juarismi** fue el sobrenombre del célebre matemático árabe Muhammad ibn Musa (780-850).
- Escribió un libro en 825 ...
- Que se tradujo en el siglo XII como “**Algoritmi de numero Indorum**”.

Aporte

- Introdujo los números a Occidente.
- Las matemáticas no podrían haber avanzado, como lo hicieron, si no se hubiera reemplazado los números romanos (I, II, III, IV, V,...) por los indo-árabicos (1, 2, 3, 4, 5, ..., 0).



Muhammad ibn Musa (780-850)

* Tiene variantes de deletreo

Algoritmos

- Los dígitos del 1 al 9 más el símbolo 0.

Arábigo Occidental	0	1	2	3	4	5	6	7	8	9
Arábico-Índico	.	۱	۲	۳	۴	۵	۶	۷	۸	۹
Arábico-Índico Oriental (Persa y Urdu)	.	۱	۲	۳	۴	۵	۶	۷	۸	۹

Actualidad

- (Conocimiento + Habilidades)*Actitud

El valor del ser humano

- Si tiene **ética**, entonces su valor es 1.
- Además, si es **inteligente** se agrega un cero, entonces su valor será 10.
- Si es **rico**, se agrega otro cero, entonces su valor será 100.
- Si es una **noble persona**, entonces su valor será 1000.
- Pero**, si pierde el 1 (ética), perderá todo su valor, pues solamente le quedarán ceros.
- Sin valores ni principios, no queda nada.**

Algoritmos y máquinas

Se busca entender cómo pensamos, y una vez entendido, reconstruirlo.

Propósito. Formalizar del pensamiento, que es abstracto, donde se entrelazan dos caminos:

- Bases computacionales e inteligencia artificial.
- Bases matemáticas.

Los algoritmos indican **CÓMO** y en **QUÉ** orden se deben ejecutar las instrucciones.
Lenguaje de programación es el **MEDIO** para expresar el algoritmo.
Las computadoras permiten **EJECUTAR** el programa.



Alan Turing (1912-1954)

Algoritmos y máquinas

Arquitectura de von Neumann

John von Neumann, apoyándose en los principios que diseñó Alan Turing, desarrolló una **arquitectura** que propició un salto en el desarrollo de las primeras computadoras.

- Hoy sigue vigente, con modificaciones, que han incrementado la complejidad del modelo de von Neumann.

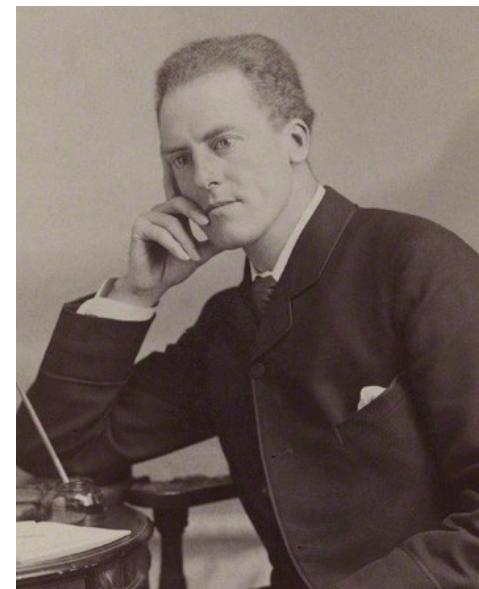


John von Neumann (1903-1957)

Algoritmos y máquinas

Se consolidó una nueva noción en el pensamiento.

- No todo es determinista, la **incertidumbre** siempre está presente.
- Vivimos con incertidumbre.



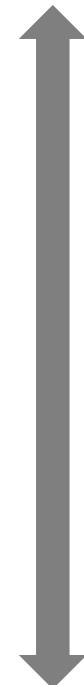
Karl Pearson (1857-1936)



Ronald Fisher (1890-1962)

Algoritmos

- Meta-modelo
- Modelo
- Procedimiento
- Método
- Técnica
- **Algoritmo**



* El **método traza el camino** y la **técnica muestra cómo recorrerlo**.

3. Aprendizaje automático

Aprendizaje Automático

Es un subconjunto de algoritmos de Inteligencia Artificial que **entrena a las máquinas** (computadoras) **para aprender**. Este aprendizaje se logra mediante la extracción de patrones de conjuntos de datos.

- El nombre fue introducido en **1959 por Arthur Samuel**. Es un campo de la ciencia que explora el desarrollo de algoritmos que pueden aprender y hacer predicciones sobre los datos.
- La principal diferencia con otros algoritmos comunes es el "aprendizaje".



Arthur Samuel (1901-1990)

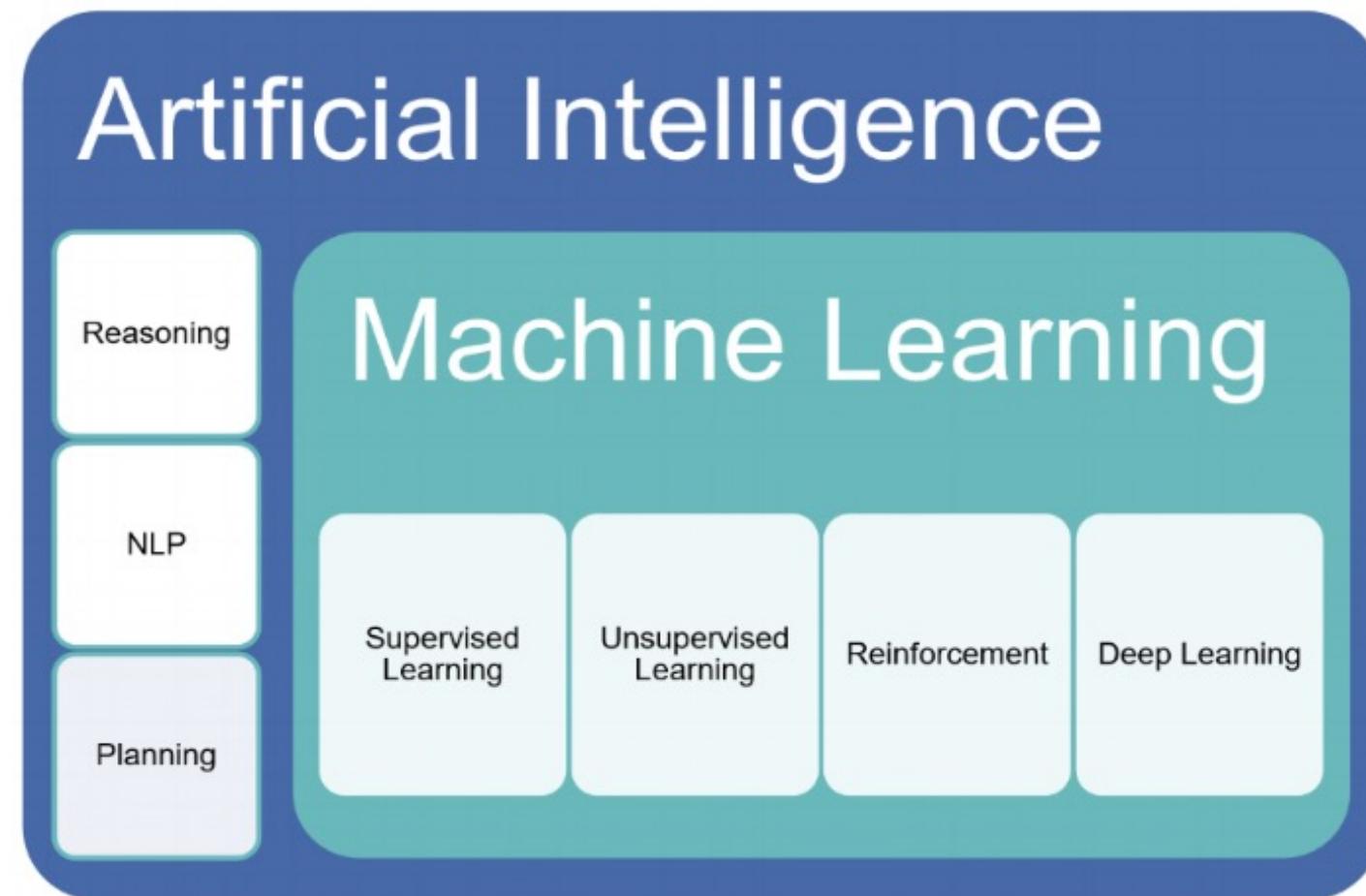
¿Qué necesita para funcionar?

- Conjuntos de datos.
- Datos diversos (heterogeneidad)

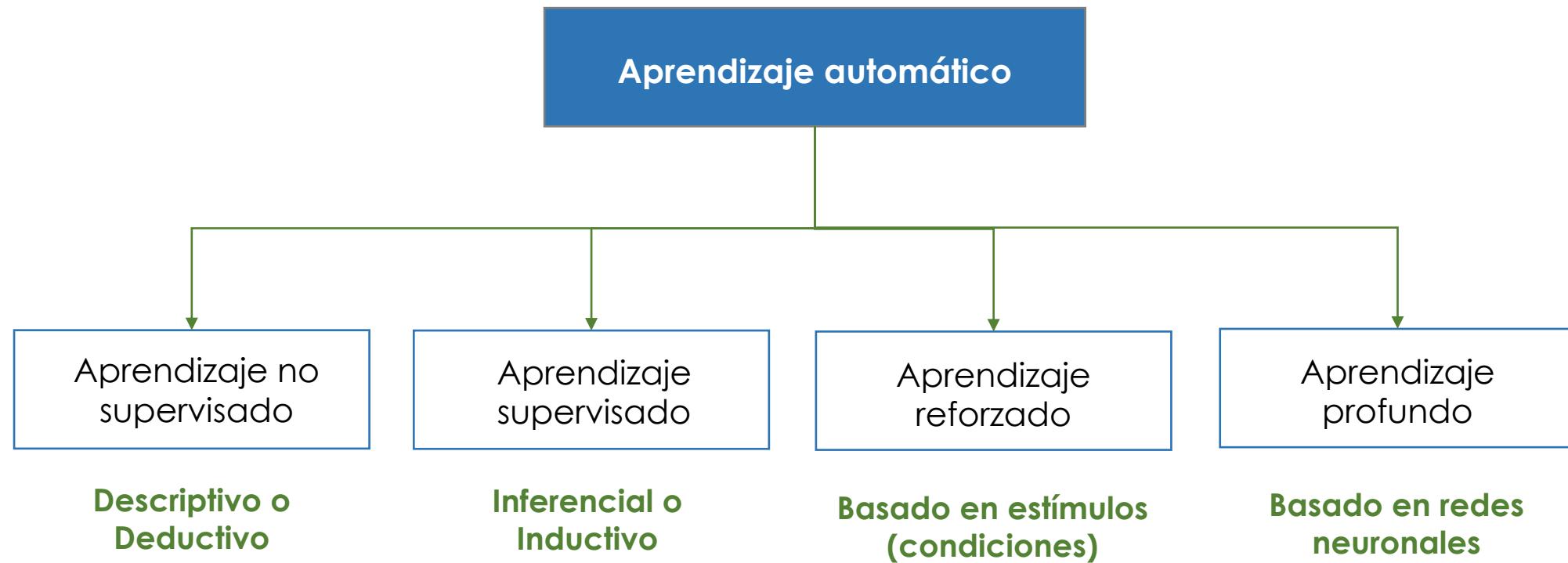
¿Qué hace que el aprendizaje automático sea exitoso?

- La respuesta se encuentra en el concepto central del aprendizaje automático: una máquina (algoritmo) puede aprender de ejemplos y la experiencia.
- Antes del aprendizaje automático, las máquinas se programaban con **instrucciones específicas** y no tenían necesidad de aprender.

Tipos de aprendizaje automático



Tipos de aprendizaje



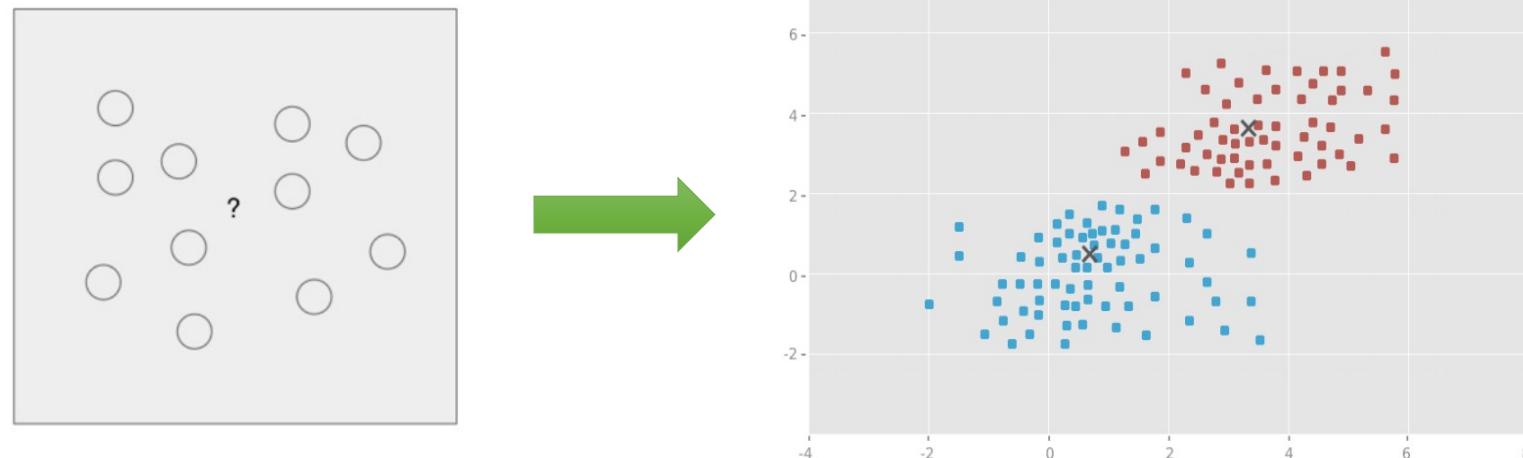
Tipos de aprendizaje

No supervisado	Supervisado	Reforzado	Profundo
<ul style="list-style-type: none">Datos sin etiquetas y resultados desconocidos.Enfocado en encontrar patrones y obtener información a partir de los datos de entrada.	<ul style="list-style-type: none">Los datos tienen etiquetas y resultados conocidos.	<ul style="list-style-type: none">Enfocado en tomar decisiones basadas en experiencias previas.	<ul style="list-style-type: none">Modelo basado en Redes Neuronales Artificiales.
<ul style="list-style-type: none">AgrupamientoReglas de asociación	<ul style="list-style-type: none">RegresionesClasificaciones	<ul style="list-style-type: none">Sistemas de recompensaProblemas complejos de decisión	<ul style="list-style-type: none">Redes de creencias profundasRedes neuronales recurrentesRedes neuronales convolucionales

Tipos de aprendizaje

1) Aprendizaje no supervisado

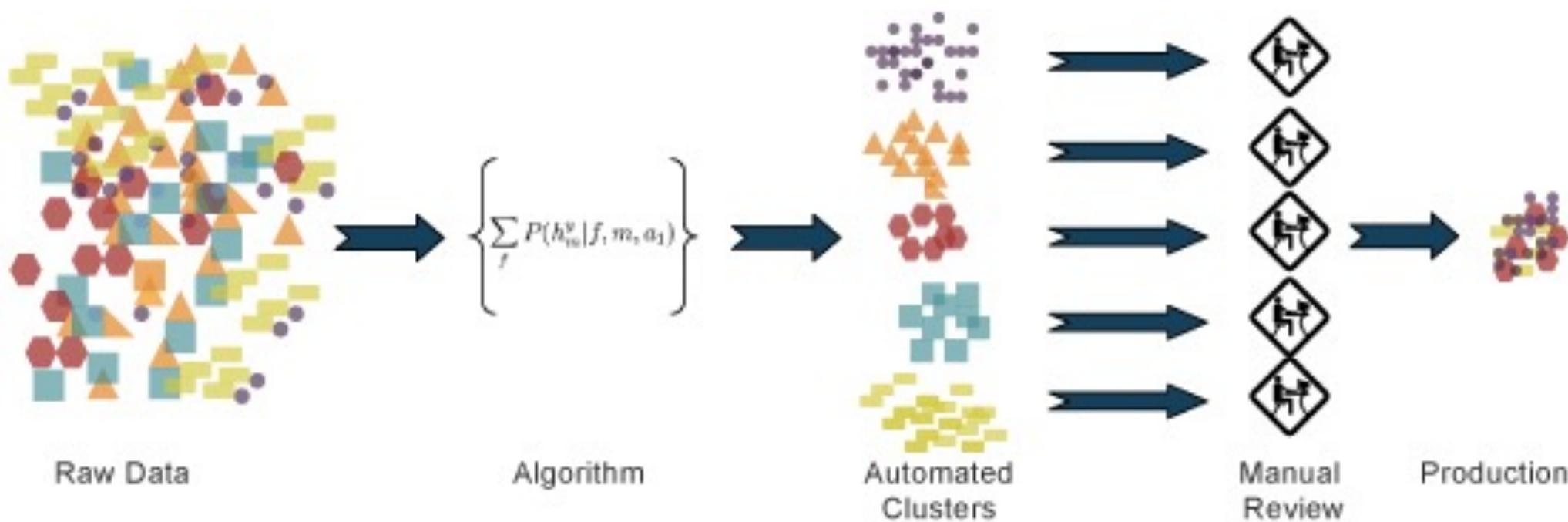
- Se trata de determinar la estructura oculta a partir de los datos.
- Los datos no requieren ser pre-etiquetados.
- Por ejemplo, calcular las distancias de los elementos y luego agruparlos y dividirlos en pequeños subgrupos.
- Otro ejemplo son los sistemas de recomendación del tipo **X** entonces **Y**.



Tipos de aprendizaje

1) Aprendizaje no supervisado

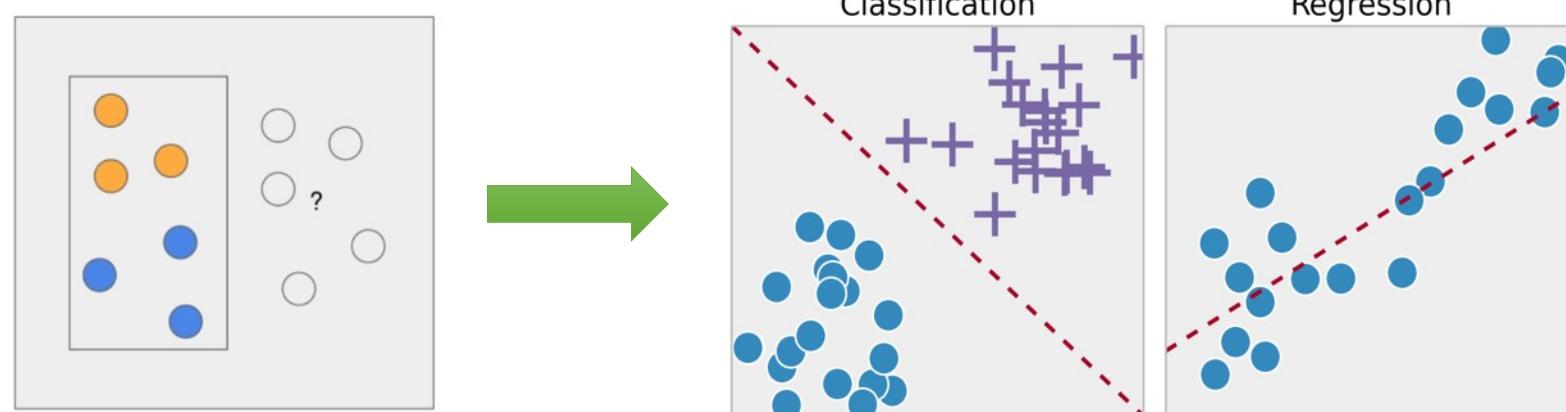
Alta dependencia del algoritmo y alta revisión manual.



Tipos de aprendizaje

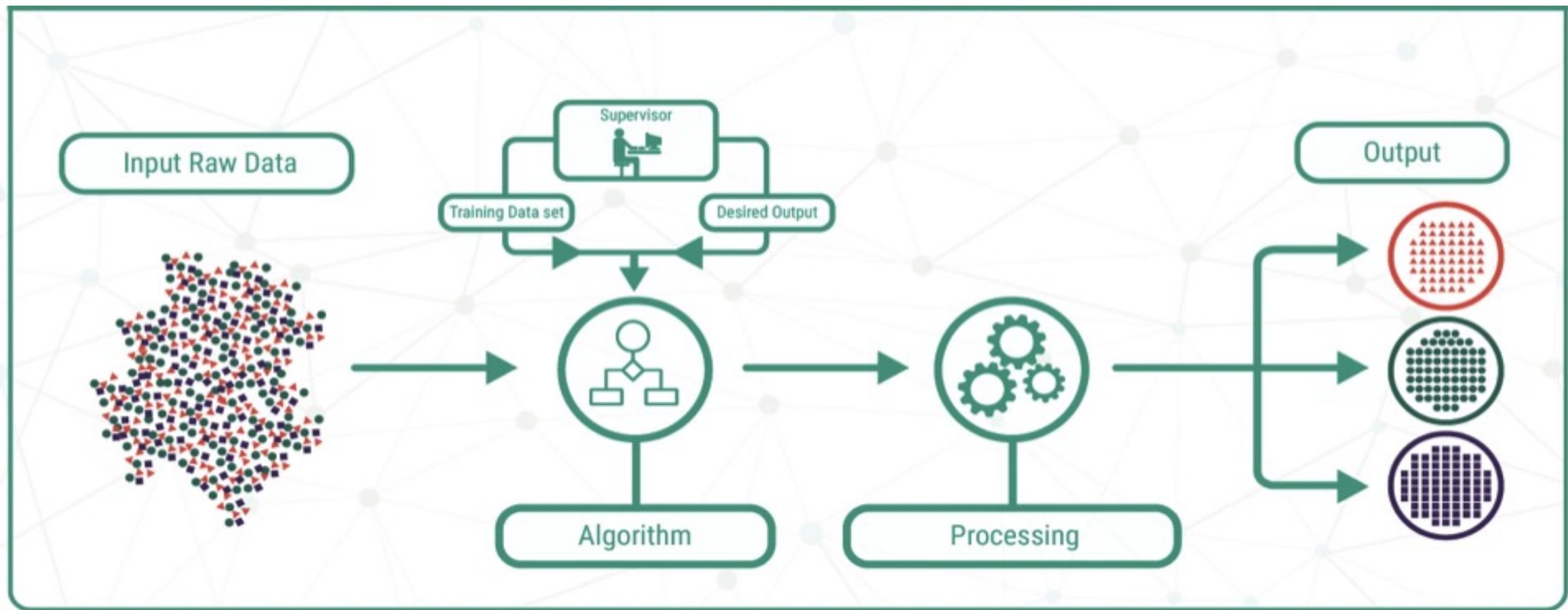
2) Aprendizaje supervisado

- En este tipo de aprendizaje se requiere etiquetar los datos, conocido como variables clase o variables dependientes.
- Basado en las características de los datos etiquetados se pueden hacer **clasificaciones** o **predicciones**.
- Por ejemplo, predicción del número de ventas de un determinado producto, pronóstico del consumo eléctrico, entre otros.



Tipos de aprendizaje

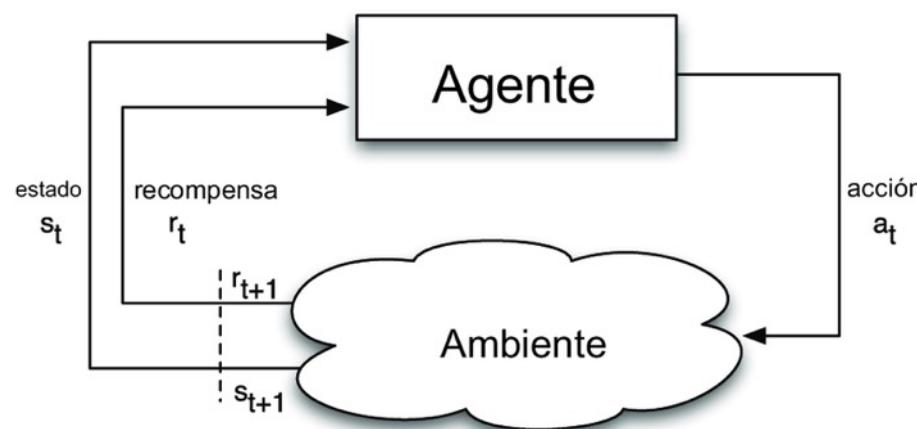
2) Aprendizaje supervisado



Tipos de aprendizaje

3) Aprendizaje reforzado

- Es conocido también como **aprendizaje por reforzamiento** o por refuerzo.
- Aprenden por **ensayo y error**, generalmente en un entorno simulado.
- La información de entrada se obtiene del mundo exterior como respuesta a determinadas acciones.
- Se cuantifica y se busca el **rendimiento máximo** en forma de recompensa.



Tipos de aprendizaje

3) Aprendizaje reforzado

Es un tipo de algoritmo de retroalimentación

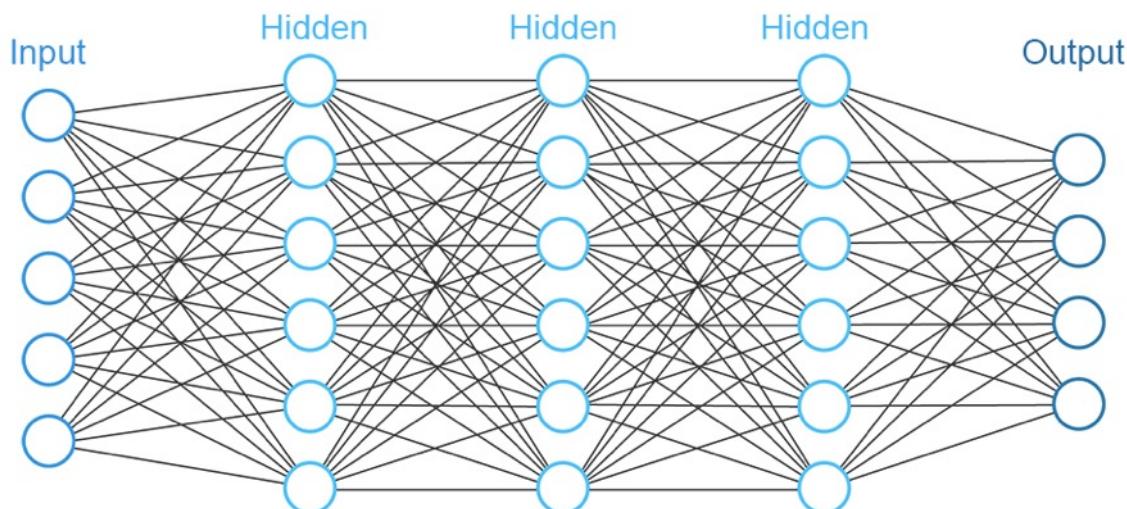


- Programación dinámica.
- Métodos Monte Carlo.
- Métodos Heurísticos.
- Procesos de Markov.
- Q-Learning.

Tipos de aprendizaje

4) Aprendizaje profundo

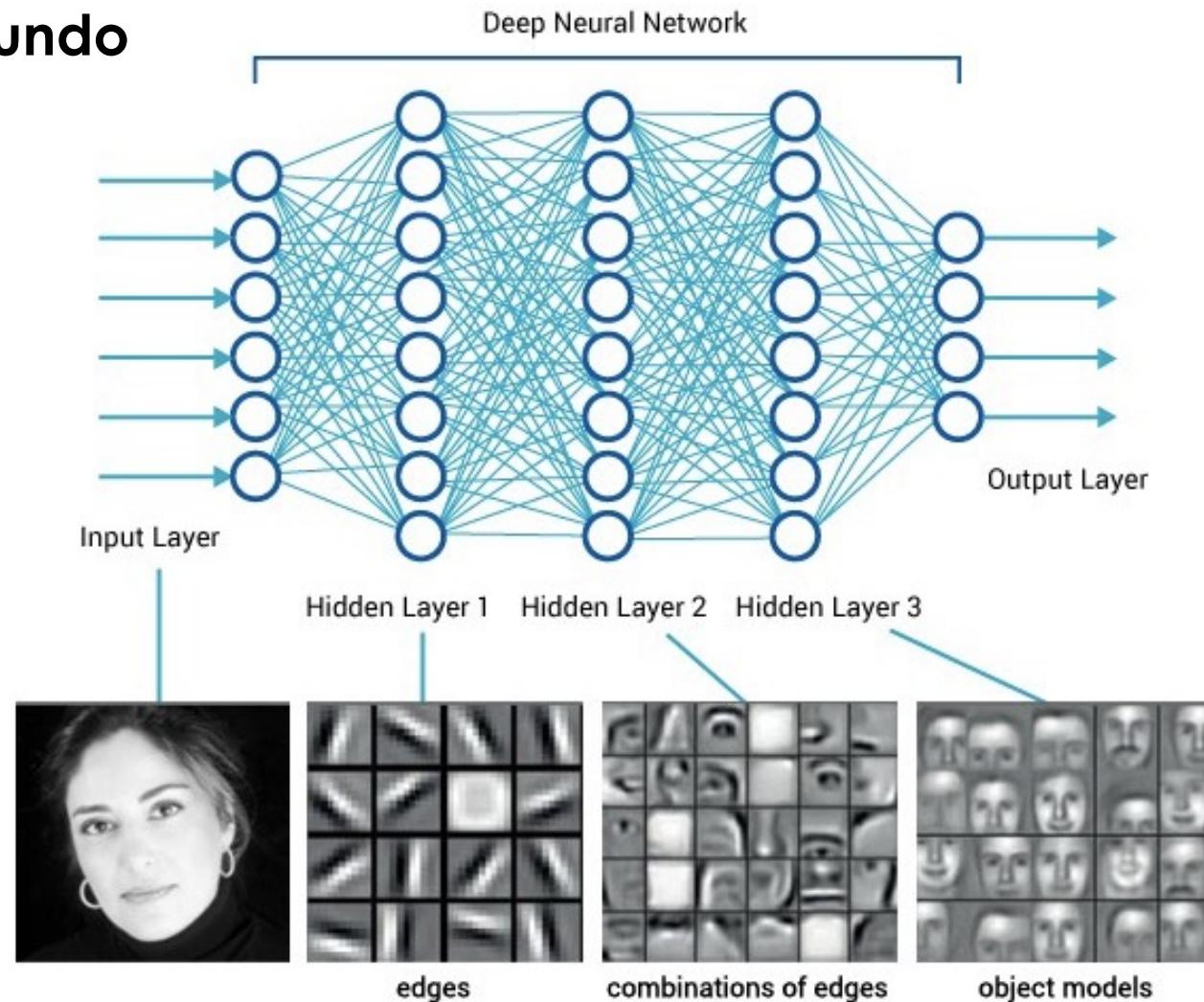
- Este tipo de aprendizaje requiere potencia de procesamiento y datos masivos, que generalmente están disponibles en estos días.
- El modelo está basado en **Redes Neuronales Artificiales** (ANN).
- Existen varias arquitecturas utilizadas, como redes neuronales profundas, redes de creencias profundas, redes neuronales recurrentes y redes neuronales convolucionales.



Se aplican con éxito en problemas de visión por computadora, reconocimiento de voz, procesamiento del lenguaje natural, análisis de imágenes y otras aplicaciones.

Tipos de aprendizaje

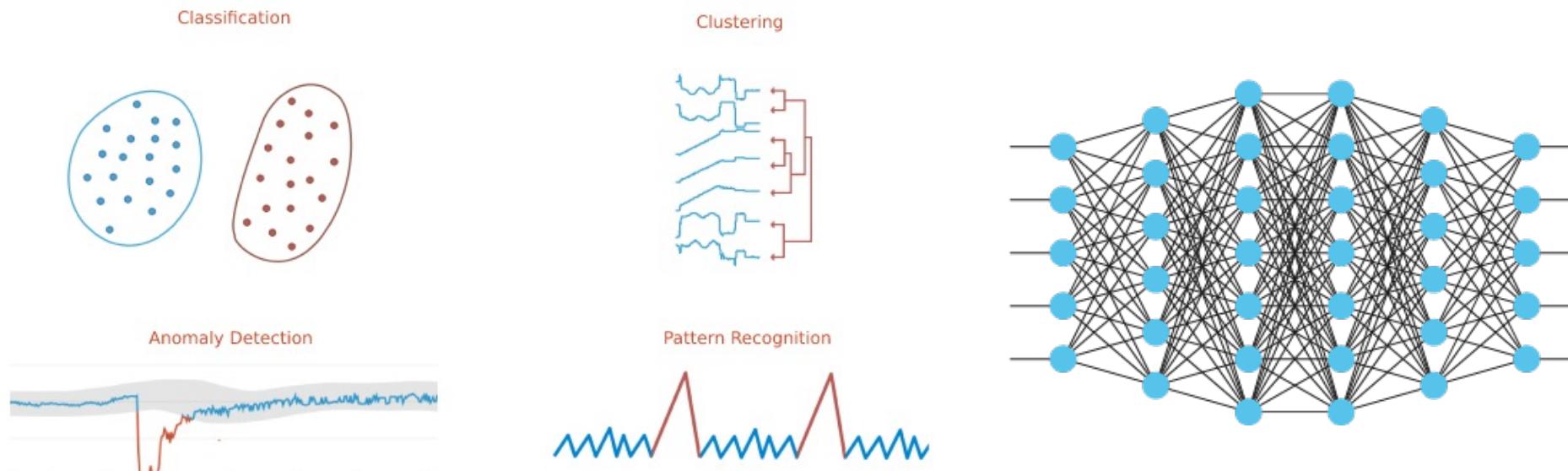
4) Aprendizaje profundo



Tipos de aprendizaje

5) Aprendizaje mixto

- Es la combinación de uno o más algoritmos de aprendizaje no supervisados, supervisados, por refuerzo y profundo.
- Algunos de sus usos generales incluyen clasificación, reconocimiento de patrones, reconocimiento de voz, detección de anomalías, análisis de imágenes, entre otros.



Tarea 2

Hacer una clasificación, mediante un cuadro, mapa mental o diagrama, de los algoritmos comúnmente utilizados en el aprendizaje automático.

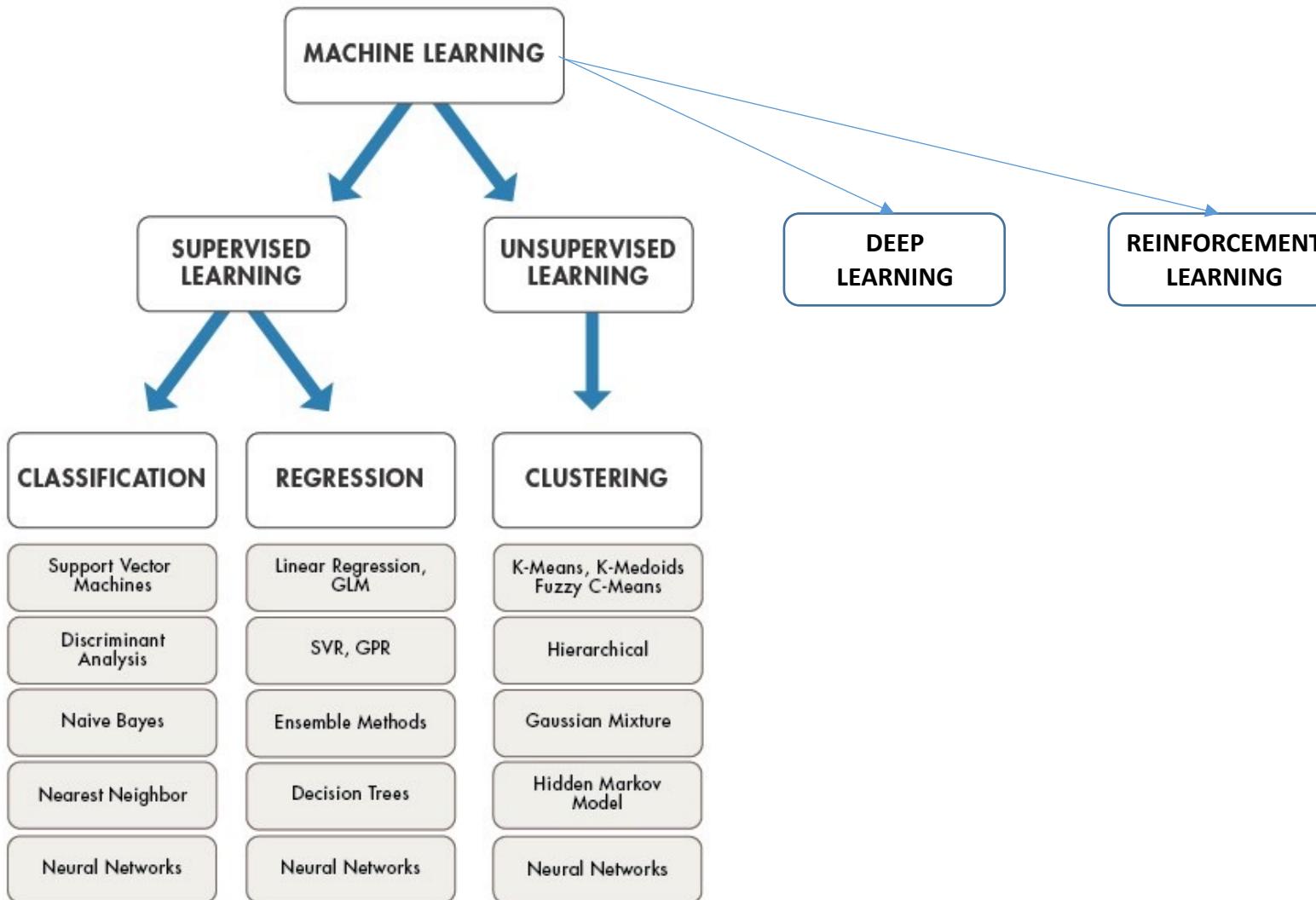
Fecha de entrega: martes 28 de septiembre de 2021

Hora: antes de las 11:00 horas

Formato: libre

Extensión: una hoja (incluir nombre y apellidos en el extremo derecho de la hoja).

Tarea 2 (Ejemplo)





Universidad Nacional Autónoma de México
Facultad de Ingeniería

Datos abiertos en la Inteligencia Artificial

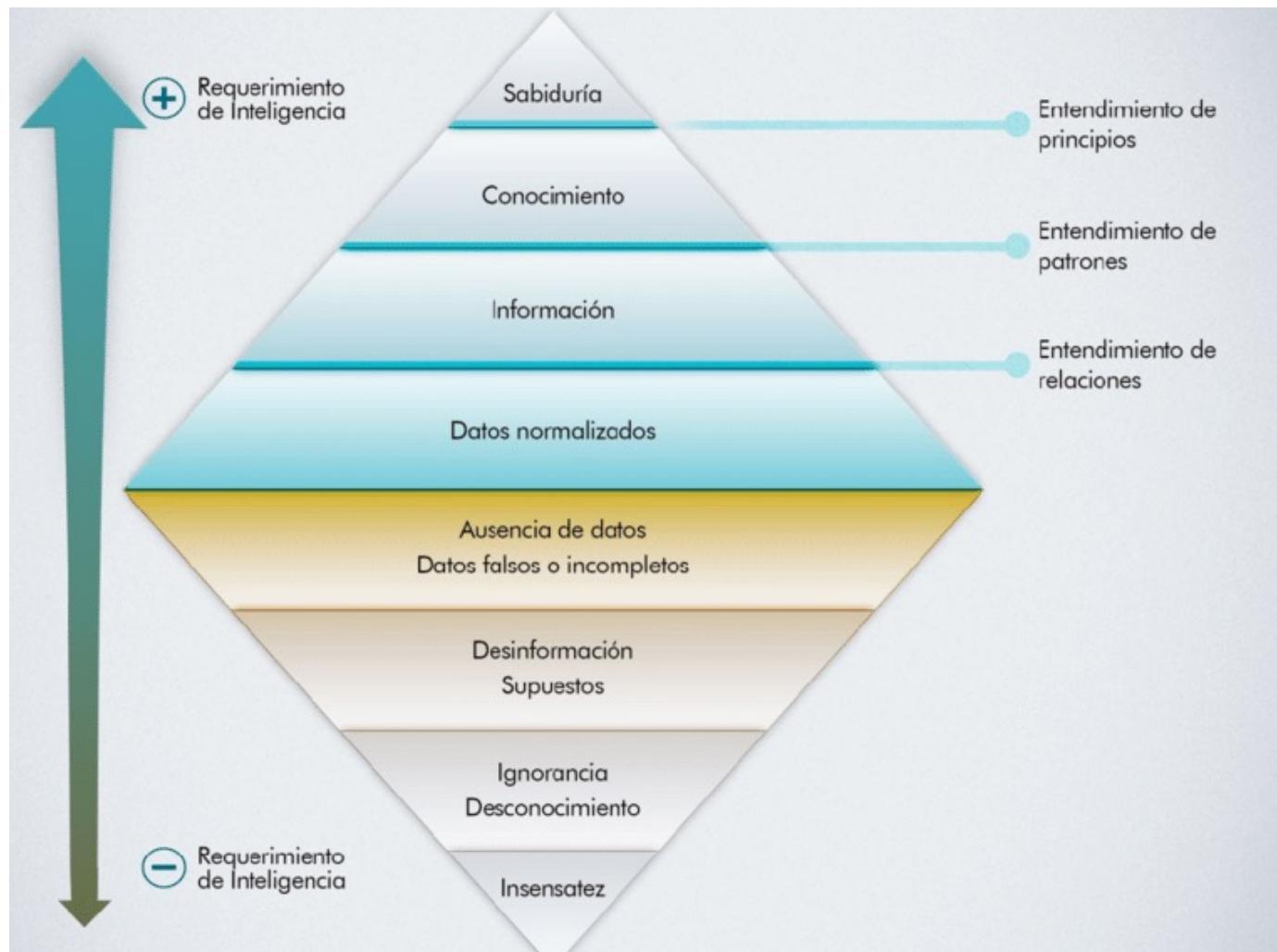
Guillermo Molero-Castillo

guillermo.molero@ingenieria.unam.edu

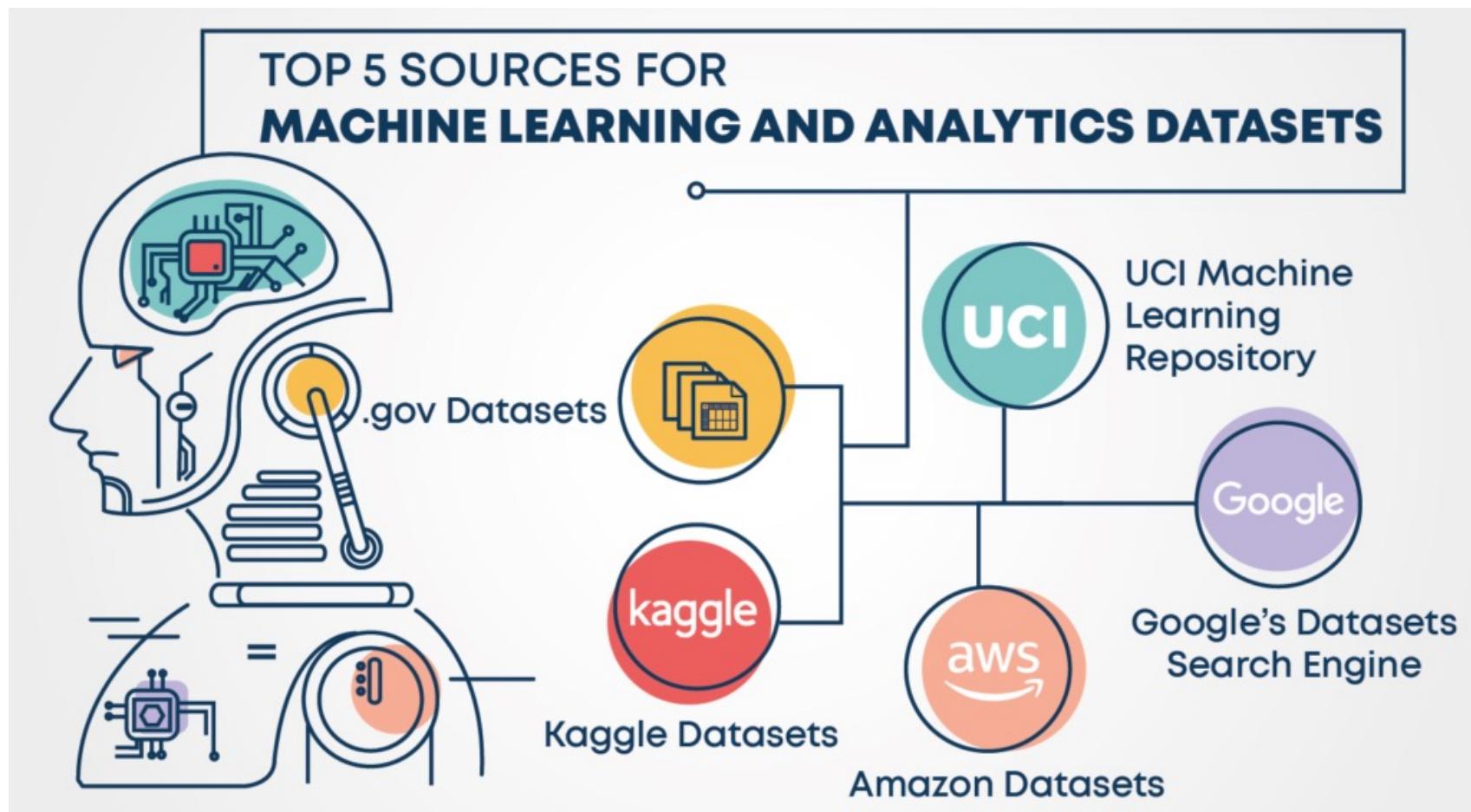
Septiembre, 2021

Datos abiertos

La jerarquía en forma de pirámide



Los **datos** son átomos de conocimiento.



Los **datos** son átomos de conocimiento.

Datos abiertos

Datos abiertos

Open data (OD) son todos aquellos **datos accesibles y reutilizables**, sin restricciones de permisos, como derechos de autor, patentes, u otros mecanismos de control.

Antecedentes:

- 2013: los miembros del G-8 firmaron un acuerdo para adoptar políticas de OD.
- Principios de Open Data (Organización de las Naciones Unidas, 2015).
- En México, Regulación en Materia de Datos Abiertos (DOF 20 de febrero de 2015).



Retos

A través de Open Data también se pueden generar oportunidades de negocios, ya sea para impulsar la creación de empresas y servicios innovadores.



Cuatro desafíos:

- Gobierno centrado en la ciudadanía.
- Presupuesto abierto y participativo.
- Empoderamiento y participación ciudadana.
- Gobernanza de recursos naturales.

Principios

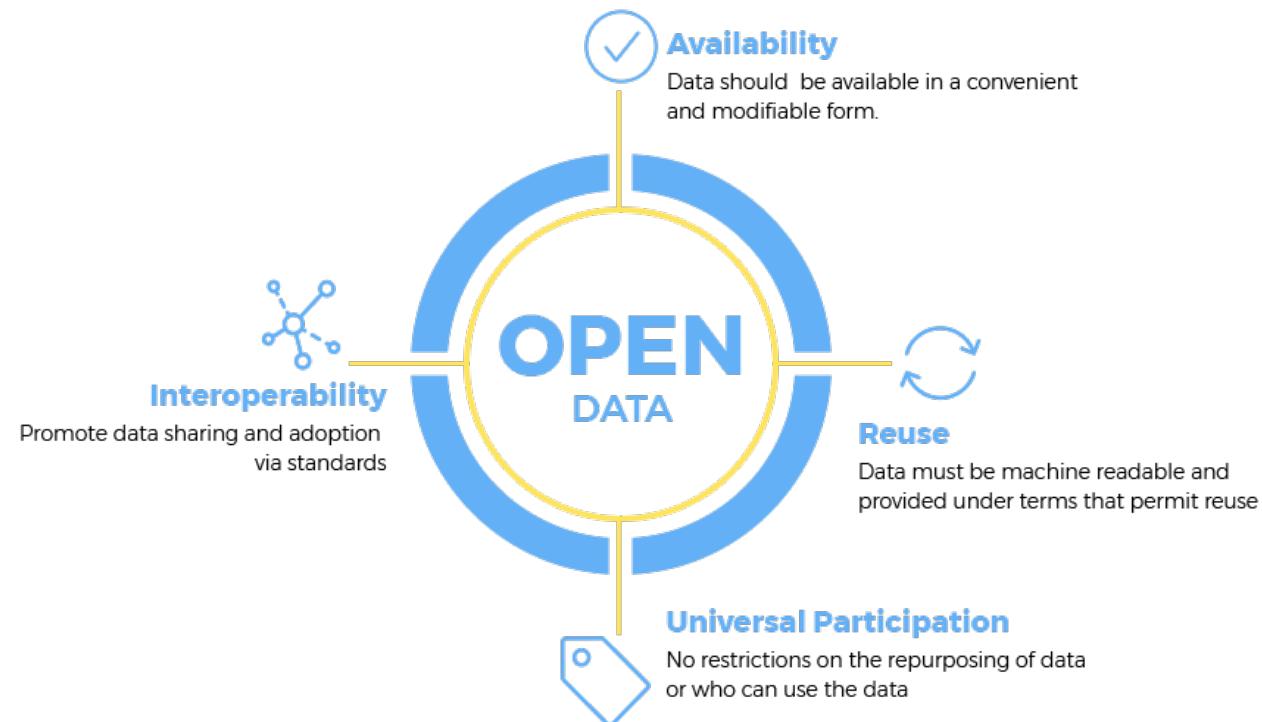
Los **Datos Abiertos** deben satisfacer las siguientes características:

- **Gratuitos.** Se obtendrán sin entregar a cambio contraprestación alguna.
- **No discriminatorios.** Serán accesibles sin restricciones para los usuarios.
- **De libre uso.** Se debe citar la fuente de origen como único requerimiento para ser utilizados.
- **Legibles.** Tienen que estar estructurados, total o parcialmente, para ser procesados.
- **Integrales.** Deben contener, en la medida de lo posible, una descripción y los metadatos necesarios.
- **Primarios.** Provendrán de la fuente de origen con el máximo nivel de desagregación posible.
- **Oportunos.** Serán actualizados periódicamente, conforme se generen.
- **Permanentes.** Deben conservarse en el tiempo, para tal efecto, las versiones históricas deben tener identificadores adecuados.

Datos abiertos

Se considera datos **verdaderamente “abiertos”** si tienen las siguientes características:

- Accesibles a través de Internet.
- En formato digital y legible para la interoperabilidad con otros datos.
- Libre de restricciones de uso o redistribución.



Datos abiertos

- En los últimos años, la **explosión de Open Data** ha evolucionado y representa ahora una influencia global.
- Estos datos son un componente cada vez más importante en el diseño de políticas sociales y económicas, tanto en países desarrollados como en desarrollo.
- Está cada vez **más institucionalizado** por los gobiernos, empresas, organizaciones de la sociedad civil e instituciones internacionales.

Datos Gubernamentales Abiertos (OGD, Open Government Data)

Tiene como finalidad estimular la economía digital, así como contribuir a una prestación más eficiente de servicios públicos y combatir la corrupción.



Formatos

Para permitir que cualquier persona utilice la información de Datos Abiertos, éstos deben estar en archivos fáciles de leer y cargar en bases de datos para su explotación. Los formatos más utilizados son:

- **XML** Extensible Markup Language, o Lenguaje Etiquetado Extensible.
- **CSV** Comma Separated Values, o Valores Separados por Comas.
- **RDF** Resource Description Framework, o Infraestructura para Descripción de Recursos.
- **RSS** Really Simple Syndication, que es un formato XML para distribuir contenido en la Web
- **Odata** Open Data Protocol, o Protocolo para Datos Abiertos.
- **JSON** JavaScript Object Notation, o Notación de Objetos de JavaScript.

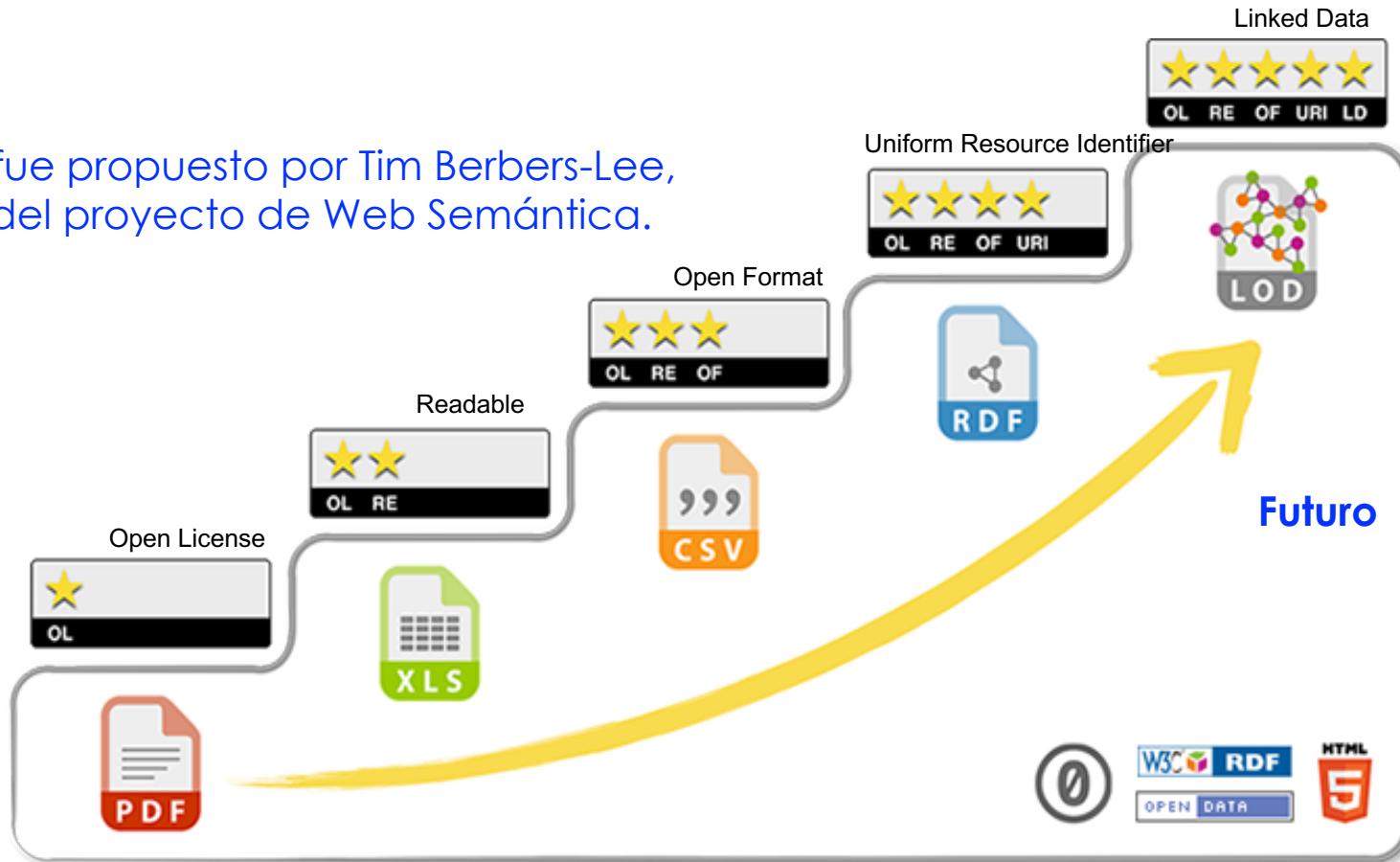


Datos abiertos

Datos enlazados (Linked Data)

Es un método de publicación de datos estructurados para que puedan ser interconectados. Se basa en tecnologías estándar, como HTTP, CSV, RDF y URI, para compartir información.

Este término fue propuesto por Tim Berners-Lee, como parte del proyecto de Web Semántica.



Alfabetización de datos (Data Literacy)

- Capacidad de leer.
- Comprender.
- Crear.
- Comunicar datos como información.

Arquitectura de datos (Data Fabric)

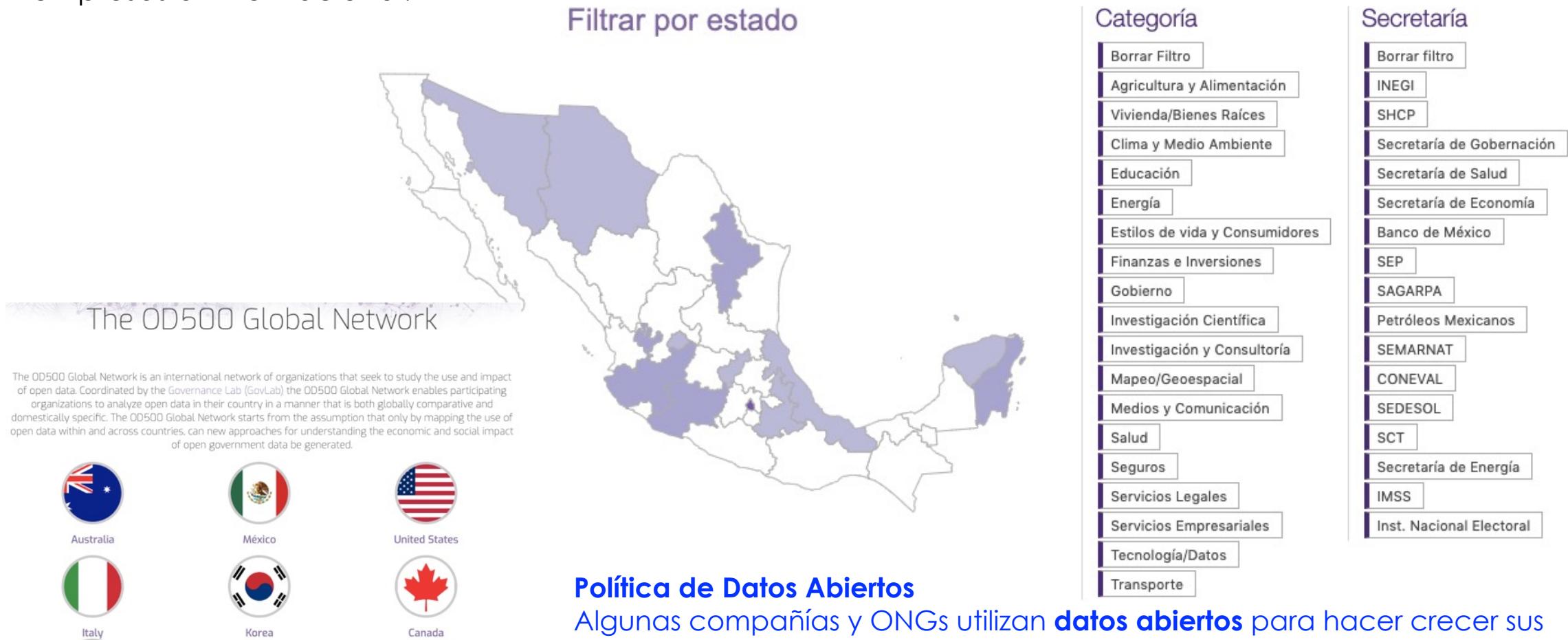
- Combinación de arquitectura y tecnología.
- Datos en todas partes y en todos los formatos.

Iniciativas en Open Data

Datos abiertos

Iniciativas nacionales e internacionales

The GovLab en colaboración de la Universidad de Nueva York analizaron el uso de datos abiertos por las empresas a nivel nacional.



Fuente: www.opendata500.com/mx/

Datos abiertos

Iniciativas nacionales e internacionales

Abison Burke Distrito Federal <i>Investigación y Consult...</i> En Abison Burke contamos con más de 12 años reuniendo talento especializado en diferentes áreas de mercadotecnia e investigación de mercado.	Abt Associates... Distrito Federal <i>Investigación y Consult...</i> Abt Associates implementa el Programa de Política Económica para México, enfocado a mejorar la productividad y competitividad a través de asistencia técnica al sector	Accenture Distrito Federal <i>Investigación y Consult...</i> Accenture es una compañía global de consultoría, tecnología y outsourcing.	Agencia de Est... Chihuahua <i>Investigación y Consult...</i> Somos una empresa de investigación, integrada por economistas y con cobertura en la zona norte del país (6 estados).
Agroclima Aguascalientes <i>Seguros</i> Agroclima Informática Avanzada genera servicios exclusivos a PROAGRO, Protección Agropecuaria Compañía de Seguros.	Akko Group Distrito Federal <i>Vivienda/Bienes Raíces</i> Nuestra compañía se caracteriza por la integración de 3 componentes fundamentales que garantizan nuestros resultados: investigación, pragmatismo y estrategias de	Alternativas y ... Distrito Federal <i>Servicios Empresariales</i> Alternativas y Capacidades es una organización fortalecedora de otras organizaciones de la sociedad civil en México.	AMAI Distrito Federal <i>Investigación y Consult...</i> Agrupación independiente de organizaciones que realizan distintas fases del proceso de generación y transformación de datos para tomar decisiones en ámbitos sociales o de
AMK Technolo... Distrito Federal <i>Tecnología/Datos</i> Nos encargamos del análisis, diseño, construcción, calidad, implementación y soporte tecnológico.	ANTAD Distrito Federal <i>Servicios Empresariales</i> Asociación Nacional de Tiendas de Autoservicio y Departamentales, A. C (ANTAD) ofrece productos y servicios para apoyar el Desarrollo del Comercio Detallista.	Aporta Nuevo León <i>Tecnología/Datos</i> Aporta es un portal que utilizará el poder del Internet y de herramientas tecnológicas, con el objetivo de digitalizar la filantropía en México.	Asociación Me... Distrito Federal <i>Medios y Comunicación</i> AMIPCI promueve el uso generalizado e intensivo de internet en los sectores estratégicos del país y su utilización y apropiación en la vida cotidiana.
Asociación Me... Distrito Federal	Asociación Me... Distrito Federal	Asociación Nac... Distrito Federal	Aspiria Jalisco

Datos abiertos

Organización para la Cooperación y el Desarrollo Económico

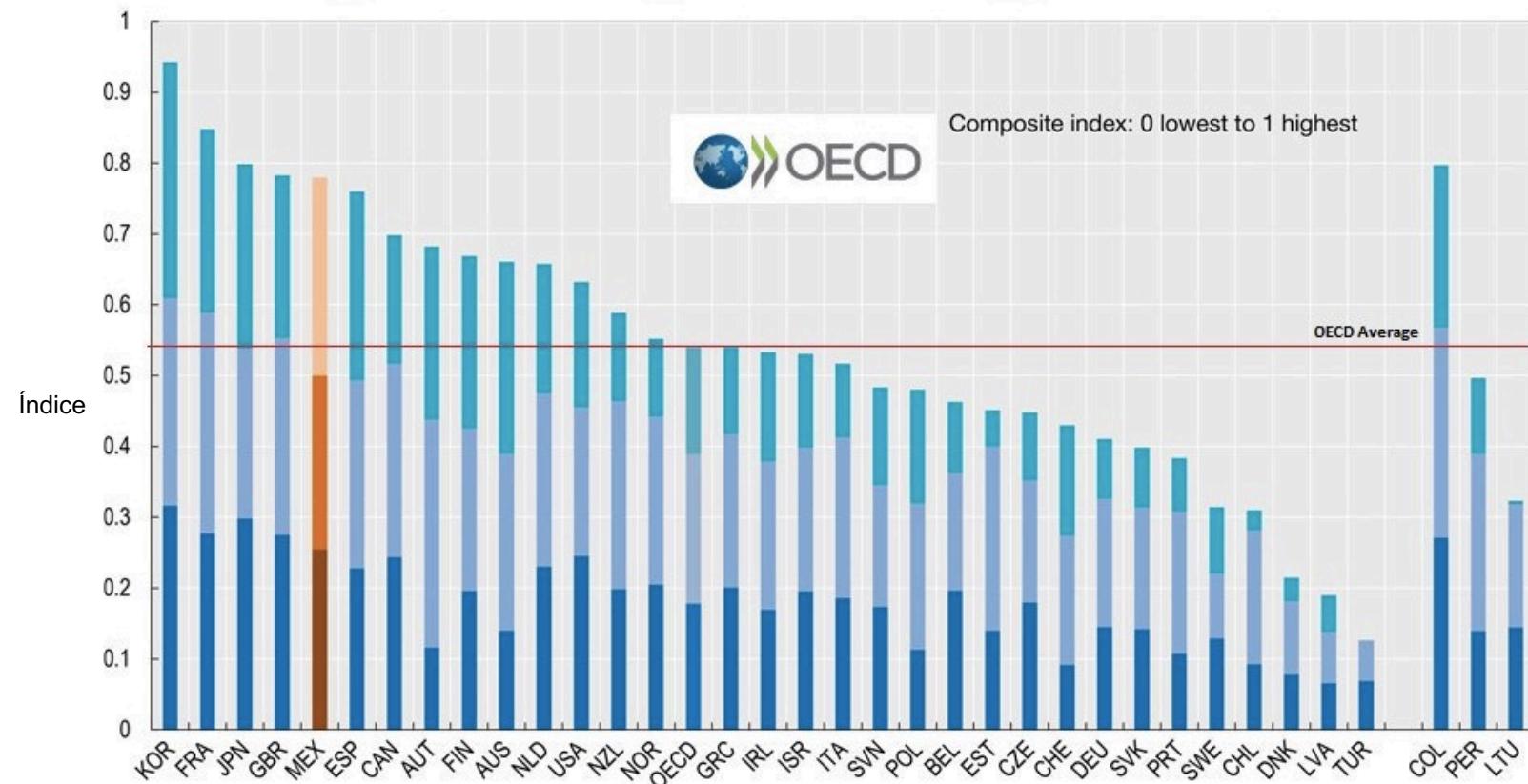
OURData mide los esfuerzos realizados por los gobiernos para aumentar la accesibilidad y disponibilidad de datos del gobierno.

Open-Useful-Reusable Government Data Index (OURdata), 2017

Data availability

Data accessibility

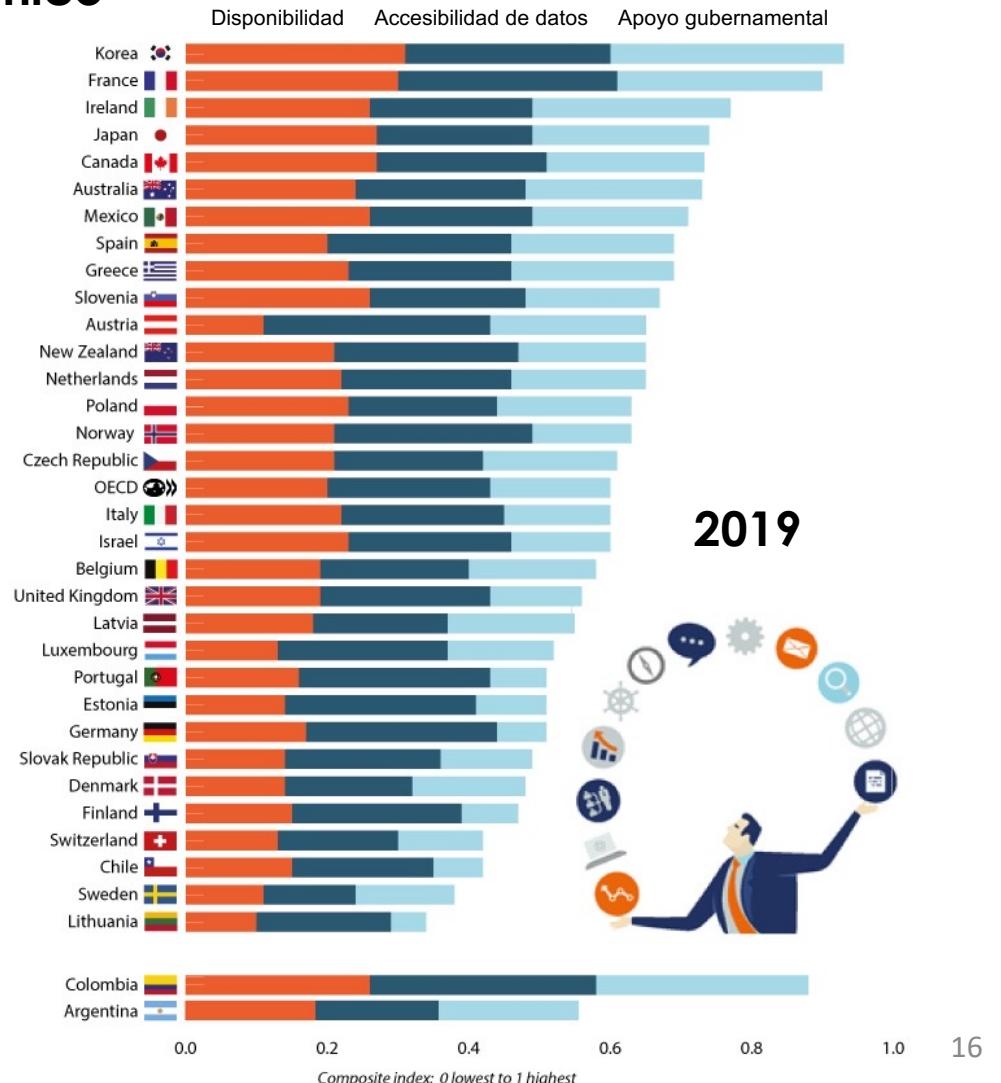
Government support to the re-use



Datos abiertos

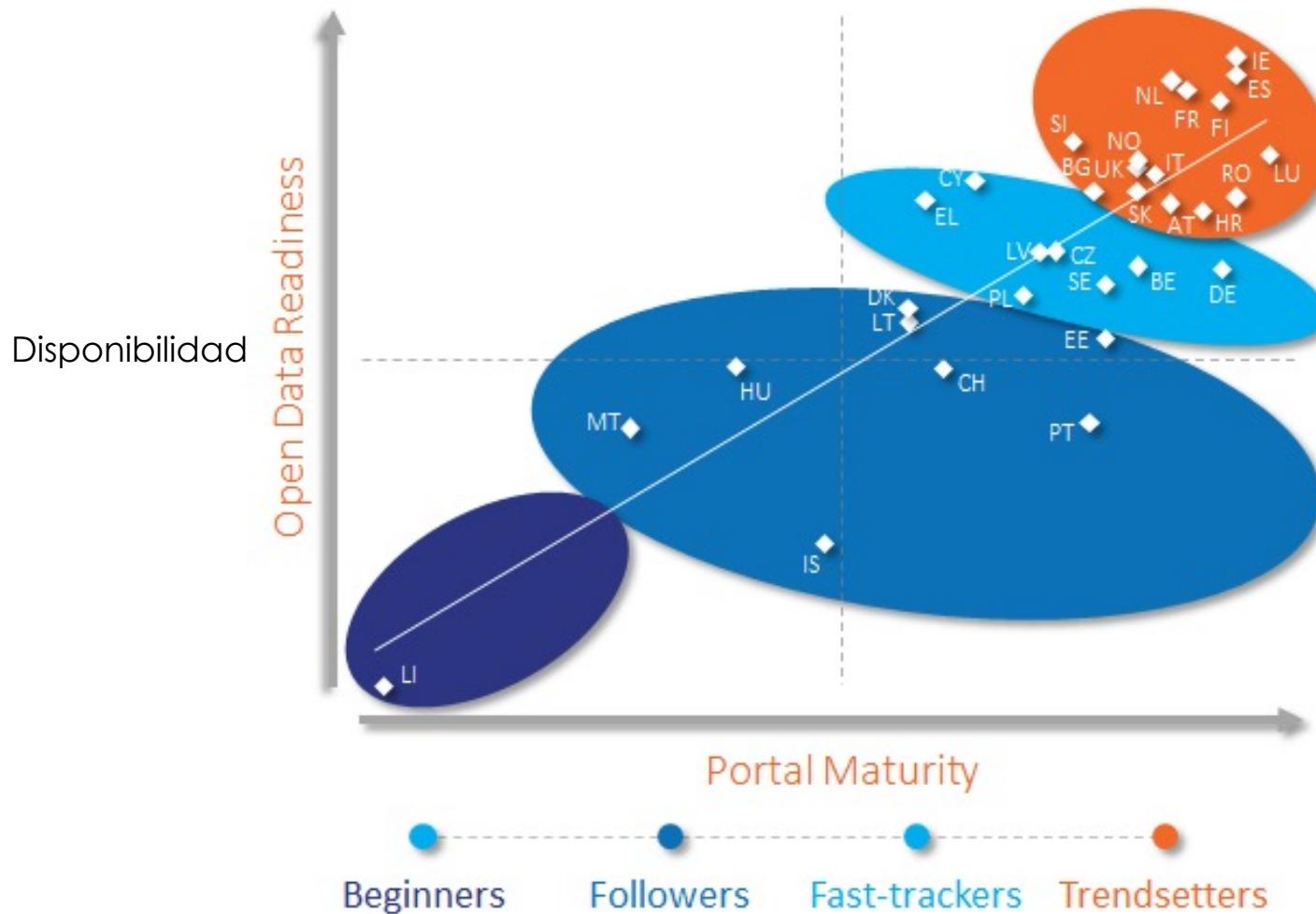
Organización para la Cooperación y el Desarrollo Económico

OURData mide los esfuerzos realizados por los gobiernos para aumentar la accesibilidad y disponibilidad de datos del gobierno.



Datos abiertos

Informe de madurez de datos abiertos en Europa 2017



Muestra los países que han acelerado el **aumento de la cantidad de datos** en sus territorios.

Se establece **cuatro niveles**: principiante, seguidor, acelerador y precursor.

Algunas iniciativas nacionales

Datos abiertos

1) Gobierno de México



Datos

Tiempo real / APIs	Recomendados	Recientes
Nombre	Institución	Formato
Información referente a casos COVID-19 en México	SALUD	CSV
Mapeo Colaborativo daños y derrumbes, centros de acopio y refugios	Otras	XLSX

URL: <https://datos.gob.mx>

Datos abiertos

2) Entidades federativas



Explorar datos Creador de mapas Sugerencia de datos Calendario de apertura

Portal de datos de la Ciudad de México

Construyendo una ciudad de ventanas transparentes

Explora datos por categoría



<https://datos.cdmx.gob.mx/pages/home>

The screenshot shows the main interface of the portal. At the top right is a navigation bar with links to Conjuntos de datos, Organizaciones, Grupos, Acerca de, and a search bar. Below the navigation is a search box with placeholder text "Ej: ambiente" and a magnifying glass icon. Underneath the search box are "Etiquetas populares" (Popular tags) buttons for Turismo, seguridad, and sector público. To the right is a section titled "Nomenclátor de Localidades" (Nomenclature of Localities) which includes a map of the State of Mexico with localities highlighted in green, and a note stating "Es necesario el uso de google chrome para el correcto despliegue de capas" (Google Chrome is required for correct layer display). At the bottom left of the screenshot, there is a summary of data statistics: 65 conjuntos de datos, 16 Organizaciones, 22 grupos, and 0 Elementos relacionados.

<http://datos.edomex.gob.mx>

Datos abiertos

2) Entidades federativas

Jalisco

Nuevo León

Veracruz

Conjuntos de datos Instituciones Temas Contacto

 Datos Abiertos

En esta plataforma encontrarás los datos de carácter público están disponibles en formatos digitales en línea, y puedes

Buscar covid

5

Ciencia y tecnología Desarrollo e integración social

Te invitamos a conocer nuestra versión beta del portal. Danos tú opinión. [Visitar nuevo portal](#)





Jalapa Enriquez 20 °C



Datos Abiertos

La página de Datos Abiertos se encuentra en etapa de elaboración y/o aprobación, una vez esté autorizado, se publicará en este apartado.

UBICACIÓN

Palacio de Gobierno. Av. Enriquez s/n. Col Centro C.P. 91000, Jalapa, Veracruz, México. Tel.

LO MÁS CONSULTADO

Portal del empleo
Transparencia

Datos abiertos

3) Organizaciones públicas

The screenshot shows the INEGI website's "Datos Abiertos" (Open Data) section. At the top, there are links for Inicio, Datos, Servicios, Transparencia, and Investigación. A search bar and language options (English, Otros idiomas, Contacto) are also present. Below the header, there are sections for Temas (Topics), Área geográficas (Geographic Areas), Programas, Microdatos, Sistemas de indicadores, and Datos primarios. The "Temas" section includes icons and labels for: Agricultura, Ganadería y Pesca; Comercio; Comercio Exterior; Empresas y Establecimientos; Gobierno; Hogares y Vivienda; Marco Geodésico; Marco Geoestadístico; and Medio Ambiente. The main content area features a large image of a map of Mexico with various data overlays, and text explaining the purpose of open data. Below this, there are links for Tema: TRANSPARENCIA | DATOS ABIERTOS, Capacitación, Fiscalización, and Organización Electoral.

The screenshot shows the IMSS website's "Bienvenido a datos abiertos IMSS" (Welcome to IMSS Open Data) section. It features a banner with a map of Mexico and text about the initiative. Below the banner, there is a search bar with the placeholder "ej. Derechohabiente". To the right, there is a section titled "IMSS DIGITAL" with links to tutorials: "Paso a paso descarga los archivos de datos abiertos de Asegurados", "Paso a paso descarga los archivos de datos abiertos de Población Derechohabiiente", and "Guía de uso del mapa interactivo". At the bottom, there is a "Busque en sus datos" (Search in your data) section with a search bar and filters for Popular, asegurados, asegurados no trabajadores, and subdelegación.

Datos abiertos

4) Instituciones académicas

The collage illustrates the implementation of open data principles across different Mexican academic entities:

- UNAM (Universidad Nacional Autónoma de México):** Shows the "Portal de Datos Abiertos UNAM" featuring a dashboard with various data visualizations and metrics.
- GOBIERNO DE MÉXICO:** Displays the "Menú principal" and a "Datos Abiertos" section, indicating the national government's role in promoting transparency.
- IPN (Instituto Politécnico Nacional):** Shows the "Transparencia" page under the "CTAG" (Comisión de Transparencia y Acceso a la Información Pública) banner, highlighting the institution's commitment to openness.
- UDG (Universidad de Guadalajara):** Shows the "Datos Abiertos" page, which emphasizes the university's role in generating and sharing knowledge through its open data portal.

DE ESTA FORMA SE FORTALECE
LA TRANSPARENCIA Y LA RENDICIÓN DE CUENTAS UNIVERSITARIA

Algunas iniciativas internacionales

Datos abiertos

1) UCI Machine Learning



The screenshot shows the homepage of the UC Irvine Machine Learning Repository. At the top, there is a logo featuring a yellow 'UCI' monogram next to a white line drawing of an antelope. Below the logo, the text 'Machine Learning Repository' and 'Center for Machine Learning and Intelligent Systems' is displayed. On the right side of the header, there are links for 'About', 'Citation Policy', 'Donate a Data Set', and 'Contact'. A search bar with a 'Search' button is located above a navigation menu with options for 'Repository' and 'Web'. A 'Google' search link is also present. A large blue button labeled 'View ALL Data Sets' is prominently displayed.

Welcome to the UC Irvine Machine Learning Repository!

We currently maintain 481 data sets as a service to the machine learning community. You may [view all data sets](#) through our searchable interface. For a general overview of the Repository, please visit our [About](#) page. For information about citing data sets in publications, please read our [citation policy](#). If you wish to donate a data set, please consult our [donation policy](#). For any other questions, feel free to [contact the Repository librarians](#).

Supported By:  In Collaboration With: 

Latest News:	Newest Data Sets:	Most Popular Data Sets (hits since 2007):
<p>09-24-2018: Welcome to the new Repository admins Dheeru Dua and Efi Karra Taniskidou!</p> <p>04-04-2013: Welcome to the new Repository admins Kevin Bache and Moshe Lichman!</p> <p>03-01-2010: Note from donor regarding Netflix data</p> <p>10-16-2009: Two new data sets have been added.</p> <p>09-14-2009: Several data sets have been added.</p> <p>03-24-2008: New data sets have been added!</p> <p>06-25-2007: Two new data sets have been added: UJI Pen Characters, MAGIC Gamma Telescope</p>	<p>07-30-2019:  PPG-DaLiA</p> <p>07-24-2019:  Divorce Predictors data set</p> <p>07-22-2019:  Alcohol QCM Sensor Dataset</p> <p>07-14-2019:  Incident management process enriched event log</p> <p>06-30-2019:  Wave Energy Converters</p> <p>06-22-2019:  Query Analytics Workloads Dataset</p>	<p>2798896:  Iris</p> <p>1565001:  Adult</p> <p>1214136:  Wine</p> <p>1026608:  Car Evaluation</p> <p>1005775:  Wine Quality</p> <p>994777:  Heart Disease</p>

URL: <http://archive.ics.uci.edu/ml>

Datos abiertos

2) Kaggle (Subsidiaria de Google)

The screenshot shows the Kaggle Datasets homepage. On the left, there's a vertical sidebar with icons for navigation. The main header says "Datasets" and has a search bar, "Sign In", and "Register" buttons. A prominent section titled "Engage With Dataset Tasks" encourages users to create and solve tasks on datasets. Below this, a search bar finds "56,382 datasets". A "Feedback" button and a "Filter" button are also present. The main content area shows a "Public" dataset list. The top dataset is the "COVID-19 Open Research Dataset Challenge (CORD-19)" by the Allen Institute For AI, which has 8420 submissions, was posted 4 hours ago, is 5 GB in size, has a rating of 8.8, and contains 204061 files in JSON, CSV, and other formats. Other visible datasets include "Predict Likelihood of Admission" and "Predict Heart Failure".

Search

Sign In Register

+ New Dataset

Datasets

Find and use datasets or complete tasks. [Learn more.](#)

Engage With Dataset Tasks

You can now actively engage with datasets with thousands of tasks! Help the community by creating and solving Tasks on datasets!

Tackle a new task See Details

Search 56,382 datasets Feedback Filter

Public Sort by: Hottest

 COVID-19 Open Research Dataset Challenge (CORD-19)
Allen Institute For AI Link
4 hours 5 GB 8.8 204061 Files (JSON, CSV, other)

Predict Likelihood of Admission
96 Submissions · In Graduate Admission 2

Predict Heart Failure
80 Submissions · In Heart Failure Prediction

URL: www.kaggle.com/datasets

3) KDnuggets

The screenshot shows the KDnuggets homepage with a yellow header bar. The header includes the KDnuggets logo, social media links (Twitter, Facebook, LinkedIn), a search bar, and navigation links like SOFTWARE, News/Blog, Top stories, Opinions, Tutorials, JOBS, Companies, Courses, Datasets, EDUCATION, Certificates, Meetings, and Webinars.

The main content area has a yellow header "Datasets for Data Mining and Data Science". Below it, there are social sharing buttons (Like 92, Share 92, Tweet, Share, 8.3K). A "See also" section lists links to Government data sites, Data APIs, Data Mining competitions, and Google Dataset Search. Another section, "Data repositories", lists links to Anacode Chinese Web Datastore, AssetMacro, Awesome Public Datasets, AWS Public Data Sets, BigML, and Bioassay data.

The right sidebar features a "Latest News" section with links to various articles. At the bottom, there is an advertisement for SAS Viya, featuring a cloud icon and text: "Want to code in Python, Java or R? We're open to that. Try SAS® Viya® free for 14 days."

URL: <https://www.kdnuggets.com/datasets/index.html>

4) Amazon Web Services (AWS)



About

This registry exists to help people discover and share datasets that are available via AWS resources. [Learn more about sharing data on AWS.](#)

See [all usage examples for datasets listed in this registry](#).

See datasets from [Facebook Data for Good](#), [NOAA Big Data Project](#), and [Space Telescope Science Institute](#).

Search datasets (currently 109 matching datasets)

Add to this registry

If you want to add a dataset or example of how to use a dataset to this registry, please follow the instructions on the [Registry of Open Data on AWS GitHub repository](#).

Unless specifically stated in the applicable dataset documentation, datasets available through the Registry of Open Data on AWS are not provided and maintained by AWS. Datasets are provided and maintained by a variety of third parties under a variety of licenses. Please check dataset licenses and related documentation to determine if a dataset may be used for your application.

Sentinel-2

[disaster response](#) [earth observation](#) [geospatial](#) [natural resource](#)
[satellite imagery](#) [sustainability](#)

The [Sentinel-2 mission](#) is a land monitoring constellation of two satellites that provide high resolution optical imagery and provide continuity for the current SPOT and Landsat missions. The mission provides a global coverage of the Earth's land surface every 5 days, making the data of great use in on-going studies. L1C data are available from June 2015 globally. L2A data are available from April 2017 over wider Europe region and globally since December 2018.

[Details →](#)

Usage examples

- Integrate imagery from the Sentinel-2 archive into your own apps, maps, and analysis with the [Sentinel-2 image service](#) by Esri
- Satellite Search by [Remote Pixel](#) by [Remote Pixel](#)
- Using Vector tiles and AWS Lambda, we can build a really simple API to get Landsat and Sentinel images by [Remote Pixel](#)
- QGIS plugin for Sentinel-2 data by [Sinergise](#)
- Sentinel Playground by [Sinergise](#)

[See 16 usage examples →](#)

Landsat 8

[disaster response](#) [earth observation](#) [geospatial](#) [natural resource](#)
[satellite imagery](#) [sustainability](#)

5) Google Dataset Search

Google Dataset Search Beta

Probar [boston education data](#) o [weather site:noaa.gov](#)

Consulta más información sobre la inclusión de tus conjuntos de datos en Búsqueda de Datasets.

Se han encontrado más de 100 resultados



Education Statistics
www.kaggle.com
Actualizado el May 16, 2019

**kaggle U.S. Education Datasets:
Unification Project**
www.kaggle.com
Actualizado el Mar 2, 2019



Education Statistics
From World Bank Open Data
[Ver en Kaggle](#)

Fecha de actualización del conjunto de datos May 16, 2019
Conjunto de datos proporcionado por
[World Bank](#)

URL: <https://toolbox.google.com/datasetsearch>

Datos abiertos

6) Otros

The screenshot shows the 'About assets' section of the IBM Watson Studio interface. At the top, there's a navigation bar with 'Log In' and 'Sign Up' buttons. Below that is a breadcrumb trail: 'Documentation / Overview / About assets'. On the left, a sidebar titled 'Overview' contains links like 'What's new', 'IBM Watson Studio', 'IBM Watson Knowledge Catalog', 'Offering plans', 'Feature matrix', 'About assets' (which is currently selected), 'Known issues', 'FAQs', 'IBM Watson APIs', 'Security', 'Notices', 'Accessibility', 'Getting started', 'Projects', 'Preparing data', 'Data science', 'Machine learning & AI models', 'Catalogs', and 'Governance'. The main content area has a red header with 'YouTube | 8M' and tabs for 'Dataset', 'Explore', 'Download', 'Workshop', and 'About'. The page features several large, semi-transparent text overlays: 'An asset is an item of data set that's accessed', 'An asset has metadata that require specific IBM Clo...', 'The information that you can edit the properties of, an projects. For data assets', 'For analytical assets, you metadata for the asset.', 'Data assets:', '• Data asset from a file', '• Connected data asset', '• Connection asset', '• Folder asset', 'Analytical assets:', '• June 27th, 2019: Released the YouTube-8M', '• May 14th, 2018: Released an update to the labels, and reduced size / higher-quality video', 'News', 'RESEARCH', 'Research Areas', 'Datasets', 'News', 'Publications', 'About Us', and 'Follow'. There are also various logos and icons for different asset types like 'Food Games', 'Fashion', 'Hair', 'String instrument', 'Football', 'Performance', 'Call of Duty: Black Ops', 'Home improvement', 'Airplane', 'Banana', 'Poker', 'Farm', 'Walt Disney Company', 'YAHOO!', and 'WEBSCOPE'.

UNAM

YouTube-8M Segments Dataset

The YouTube-8M Segments dataset is an extensive collection of video segment annotations. In addition to annotating video segments with labels, it also includes information about the video's context, such as the video's title, description, and thumbnail.

YOUTUBE

The Yahoo Webscope Program is a reference library of interesting and scientifically useful datasets for non-commercial use by academics and other scientists.

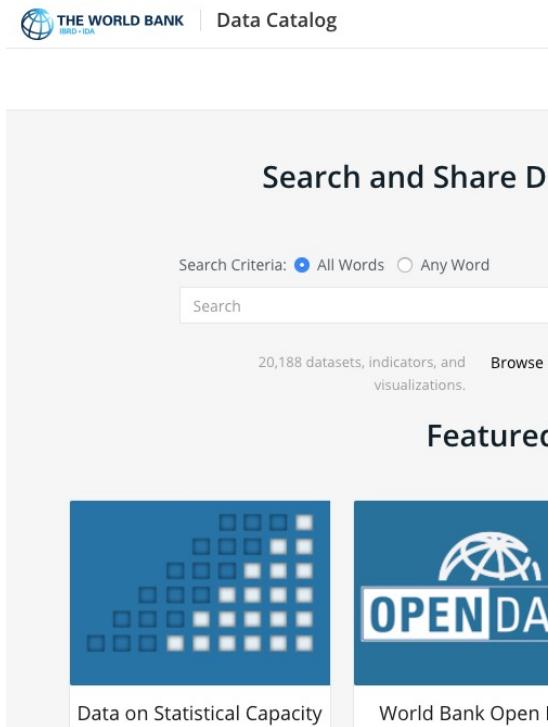
All datasets have been reviewed to conform to Yahoo's data protection standards, including strict controls on privacy. We have a number of datasets that we are excited to share with you.

Yahoo is pleased to make these datasets available to researchers who are advancing the state of knowledge and understanding in web sciences. The datasets are only available for academic use by faculty and university researchers who agree to the Data Sharing Agreement.

YAHOO

Datos abiertos

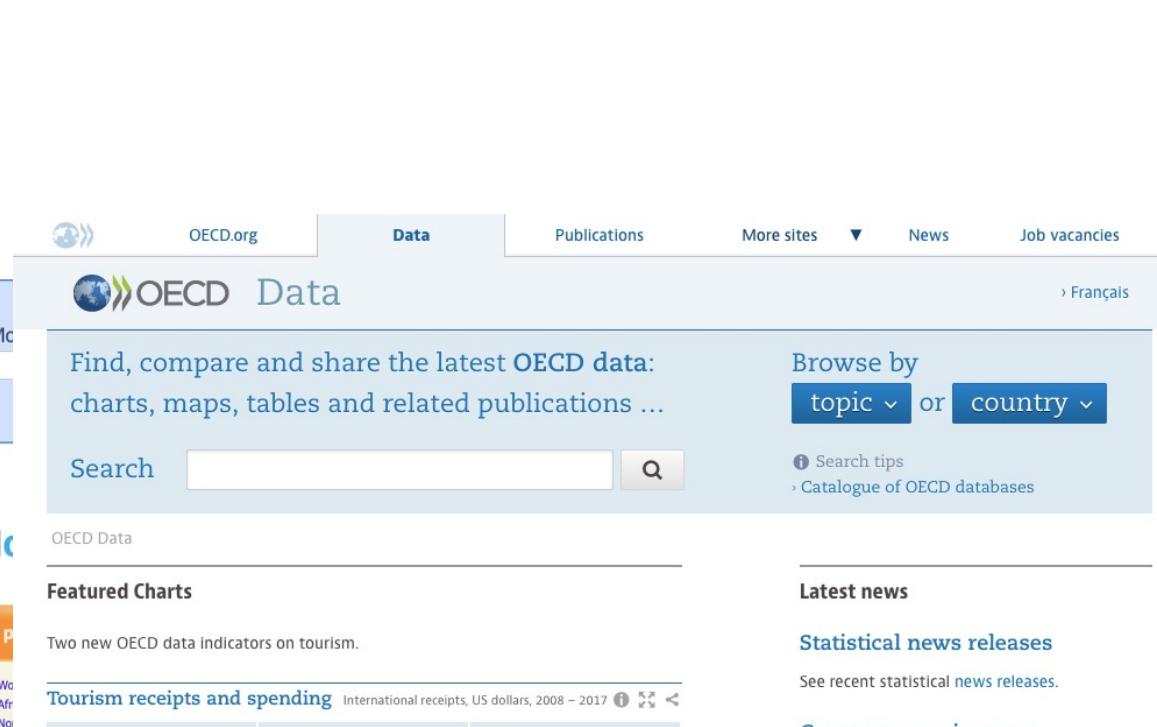
6) Otros



The screenshot shows the World Bank Data Catalog interface. It includes a search bar, a sidebar with 'Search Criteria' (All Words selected), and sections for 'Featured' datasets like 'Data on Statistical Capacity' and 'World Bank Open Data'.



The screenshot shows the UNdata website, which is a hub for various UN statistical databases. It features a search bar, a sidebar with 'Search Criteria' (All Words selected), and sections for 'Other UNSD Databases' like MBS, SDG indicators, and UN Commodity Trade Statistics.



The screenshot shows the OECD Data website. It features a search bar, a sidebar with 'Browse by topic or country', and a main section titled 'Featured Charts' showing tourism receipts and spending for countries like Canada, Greece, Iceland, Israel, Netherlands, and New Zealand.

Datos abiertos

6) Otros

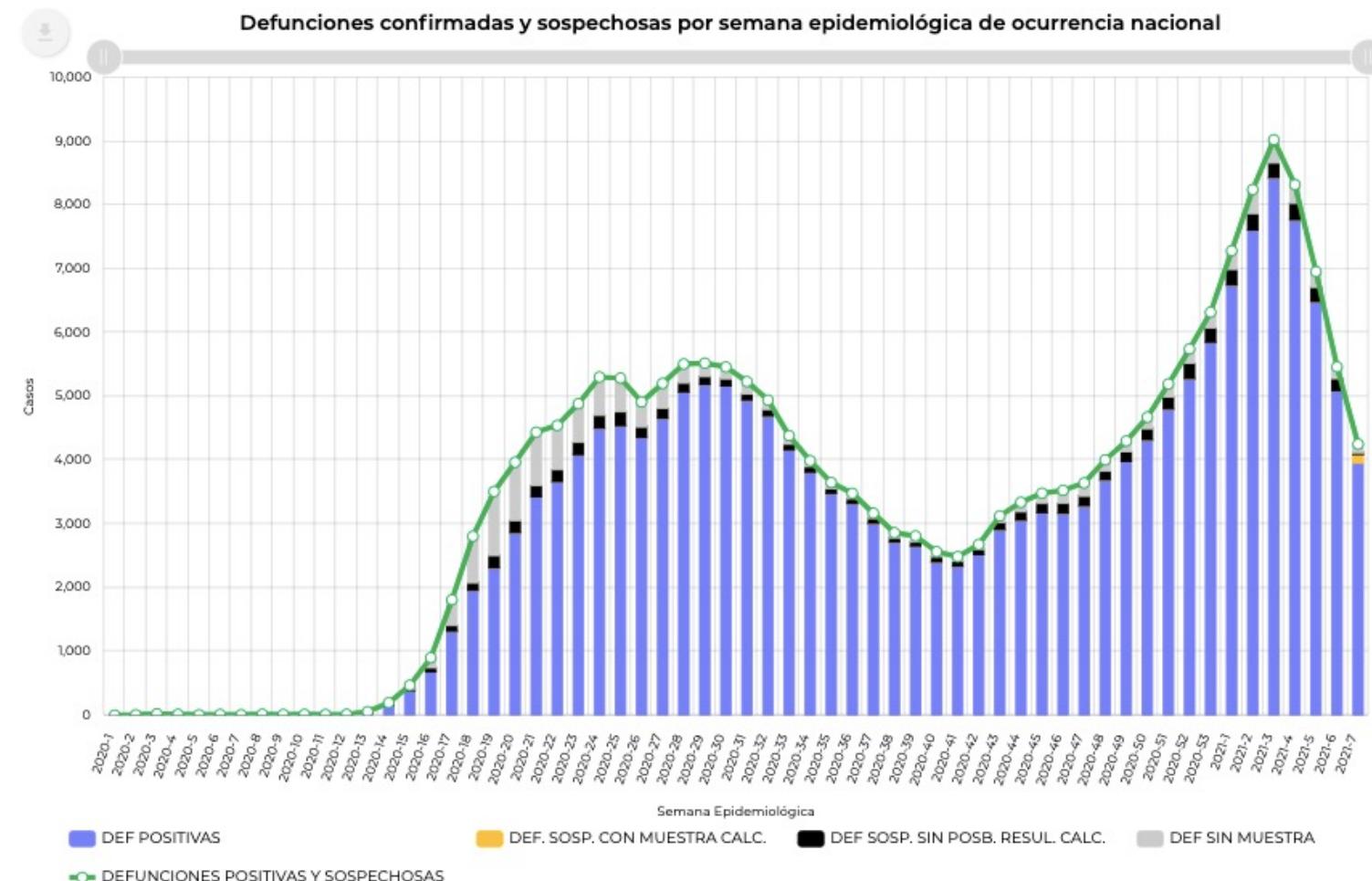
The image displays three side-by-side screenshots of government websites dedicated to open data:

- Canada.ca (Left):** Shows the Canadian Government's open government portal. It features a search bar, navigation links for Jobs, Immigration, Travel, Business, Benefits, Health, Taxes, and More services, and a "Home" link. The main content area includes sections for Open Government (describing it as about making government more accountable), Search data and information (with a search bar), Proactive disclosure (with a dropdown for Government contracts), and Access to Information (with a dropdown for Freedom of Information requests). A large "CANADÁ" watermark is overlaid at the bottom left.
- DATA.GOV (Middle):** Shows the U.S. Government's open data portal. It features a search bar, navigation links for DATA, TOPICS, IMPACT, APPLICATIONS, DEVELOPERS, and CONTACT, and a "Home" link. The main content area includes sections for The home of the U.S. Government (describing it as the home of data, tools, and resources), Credit Card Complaints, and BROWSE TOPICS (with categories like Agriculture, Climate, Consumer, and Ecosystems).
- Contraloría General de la República Costa Rica (Right):** Shows Costa Rica's open data portal. It features a search bar, navigation links for Menú, DATA, TOPICS, IMPACT, APPLICATIONS, DEVELOPERS, and CONTACT, and a "Home" link. The main content area includes sections for Datos abiertos (with a background image of binary code) and a sidebar with a language selection dropdown (Google Translate) and zoom controls (+ - ⌂).

Below the screenshots, the words "ESTADOS UNIDOS" and "COSTA RICA" are centered under their respective images.

COVID-19

Dirección General de Epidemiología



Portal de datos abiertos, Ciudad de México

The screenshot shows the Mexico City Open Data portal (GOBIERNO DE LA CIUDAD DE MÉXICO) displaying the COVID-19 SINAVE dataset. The page includes a sidebar with follower information and logos for the Government of Mexico City and the Secretaría de Salud.

Covid-19 SINAVE Ciudad de México

Seguidores: 0

Dependencia: SECRETARÍA DE SALUD

Conjunto de datos: Covid-19 SINAVE Ciudad de México

Base del Sistema Nacional de Vigilancia Epidemiologica para el seguimiento a posibles casos de COVID-19 en la Ciudad de México.

Recursos:

- base-covid-sinave.csv (CSV file, Explorar)
- diccionario.xlsx (XLSX file, Explorar)

Información Adicional:

URL: <https://datos.cdmx.gob.mx/dataset/base-covid-sinave>

COVID-19

ProgrammableWeb API DIRECTORY API NEWS Search over 23,728 APIs and much more

LEARN ABOUT APIs WHAT IS AN API ? TUTORIALS CORONAVIRUS ADD APIs & MORE

COVID-19 Developer Resource Center

Latest Coronavirus APIs and related coverage



NSW Health Pathology Aided by APIs in Response to COVID-19

Over the past four years, NSW Health Pathology invested heavily in API-led connectivity.

Coronavirus Outbreak Map



Coronavirus WHO Dashboard



URL: <https://www.programmableweb.com/coronavirus-covid-19>

APIs

Principales APIs para IA y Aprendizaje Automático

Las APIs (Application Program Interface) son un conjunto de funciones y procedimientos para crear aplicaciones.

- **BigML** URL: <https://bigml.com/developers>
- **Anaconda** URL: <https://docs.anaconda.com>
- **Blue Yonder Platform** URL: <https://github.com/blue-yonder>
- **MLJAR** URL: <https://mljar.com>
- **Recombee** URL: <https://docs.recombee.com>
- **Indico** URL: <https://indico.io/docs>
- **Animetrics Face Recognition** URL: <http://animetrics.com>
- **Eyedea Recognition** URL: <https://eyedea.ai>
- **Betaface** URL: <https://www.betifaceapi.com/wpa/index.php/documentation>
- **Wit.ai** URL: <https://wit.ai/docs>
- **Geneea** URL: <https://api.geneea.com>
- **Diffbot Analyze** URL: <https://www.diffbot.com/dev/docs>
- **Yactraq Speech2Topics** URL: <https://yactraq.com>
- **nlpTools** URL: <http://php-nlp-tools.com/documentation>



Universidad Nacional Autónoma de México
Facultad de Ingeniería

Aprendizaje no supervisado Reglas de asociación

Guillermo Molero-Castillo
guillermo.molero@ingenieria.unam.edu

Septiembre, 2021

Machine Learning

Classical Machine Learning

Task Driven

Supervised Learning

(Pre Categorized Data)



Classification

(Divide the socks by Color)

Eg. Identity Fraud Detection



Regression

(Divide the Ties by Length)

Eg. Market Forecasting

Data Driven

Unsupervised Learning

(Unlabelled Data)

Clustering

(Divide by Similarity)

Eg. Targeted Marketing

Association

(Identify Sequences)

Eg. Customer Recommendation

Dimensionality Reduction

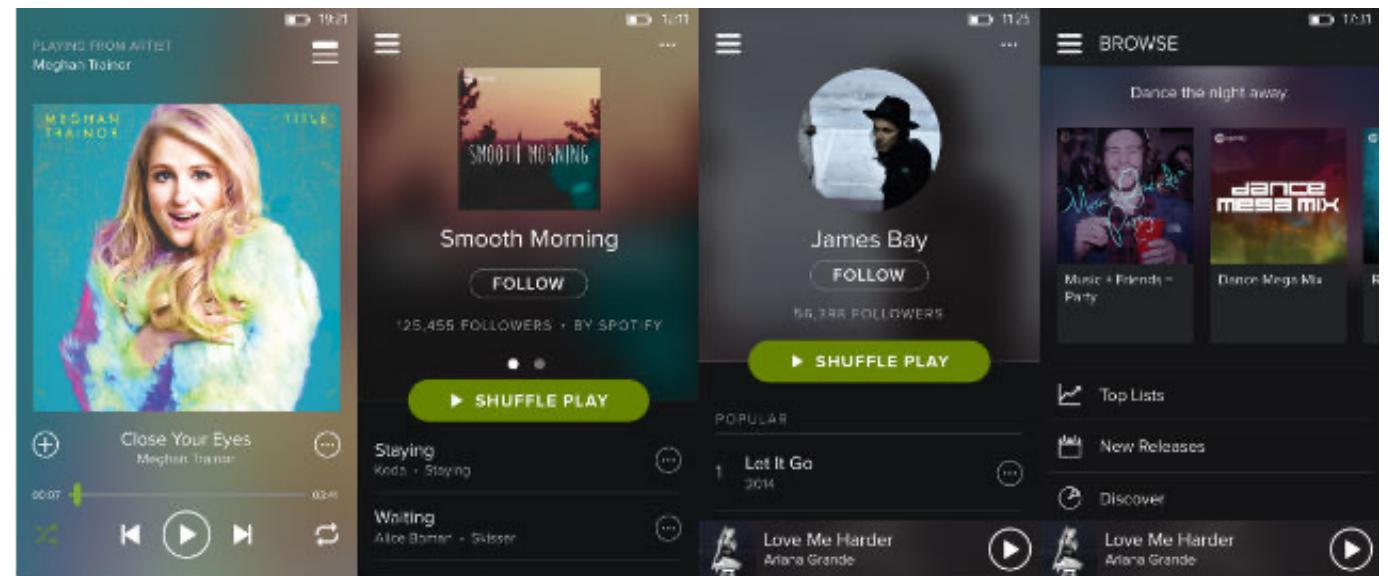
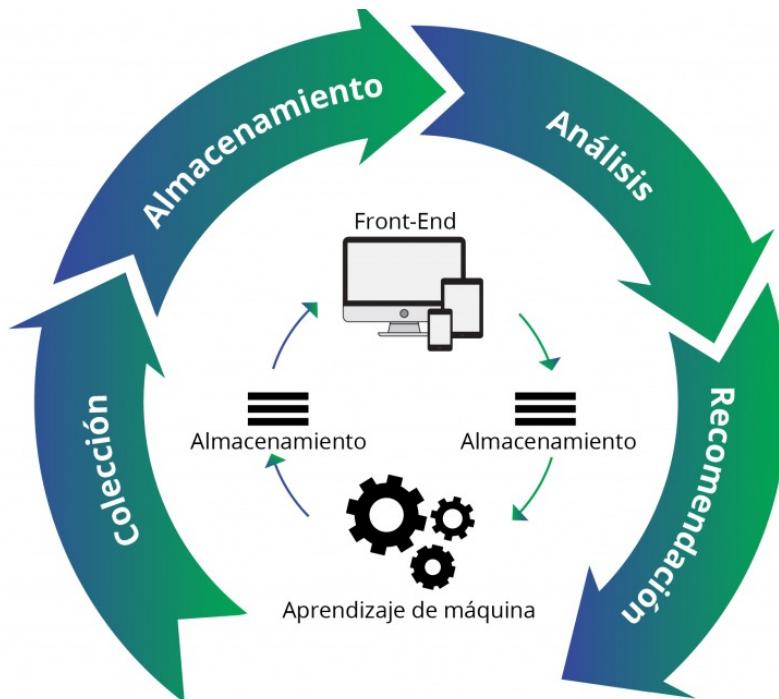
(Wider Dependencies)

Eg. Big Data Visualization

Sistemas de recomendación

Sistemas de recomendación

Un **sistema de recomendación** incluye uno o más algoritmos de Inteligencia Artificial que buscan contenido (información) sobre una fuente de datos, ya sea de clientes o usuarios, para generar recomendaciones en función de los gustos o necesidades de la persona.



* A través de estos sistemas se busca **sacar el máximo valor de los clientes**.

Sistemas de recomendación

Amazon

- Cuando se pretende comprar un artículo, se muestra un apartado de artículos relacionados (recomendados).
- Esto significa la búsqueda de la compra de más artículos.

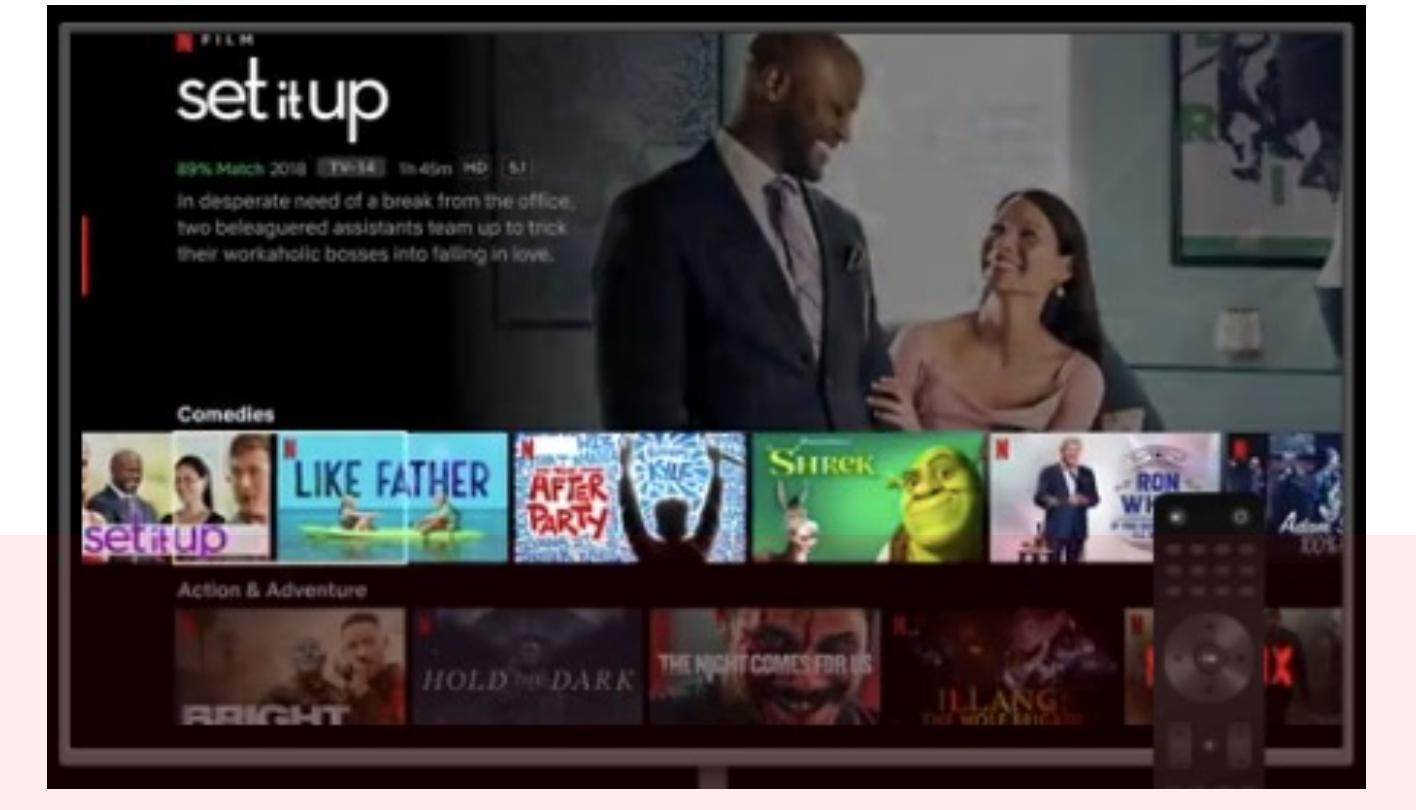
The screenshot shows an Amazon product page for an **Asus Vivobook X413FA-BV105T**. The main image displays the laptop with its screen open, showing the Asus logo. To the left is a sidebar with various thumbnail images of other laptops. The product details section includes the price of **\$16,499.00**, shipping information, and delivery options. On the right, there's a sidebar for adding the item to the cart or wishlist, along with social sharing links. Below the main product image, there's a section for related products labeled "Productos relacionados con este artículo".

This screenshot shows the "Productos relacionados con este artículo" (Related Products) section at the bottom of the Amazon product page. It features a grid of six recommended items, each with a small image and a brief description:

- Apple MacBook Pro 13" - Core i5 8a Gen/Touch
- Turtle Shell Mochila para Laptop Cumberland
- RedLemon Mochila para Laptop Antirrobo con
- RedLemon Mochila para Laptop Antirrobo con
- RedLemon Mochila para Laptop Escolar y
- RedLemon Mochila Antirrobo Impermeable

Netflix

Sistemas de recomendación

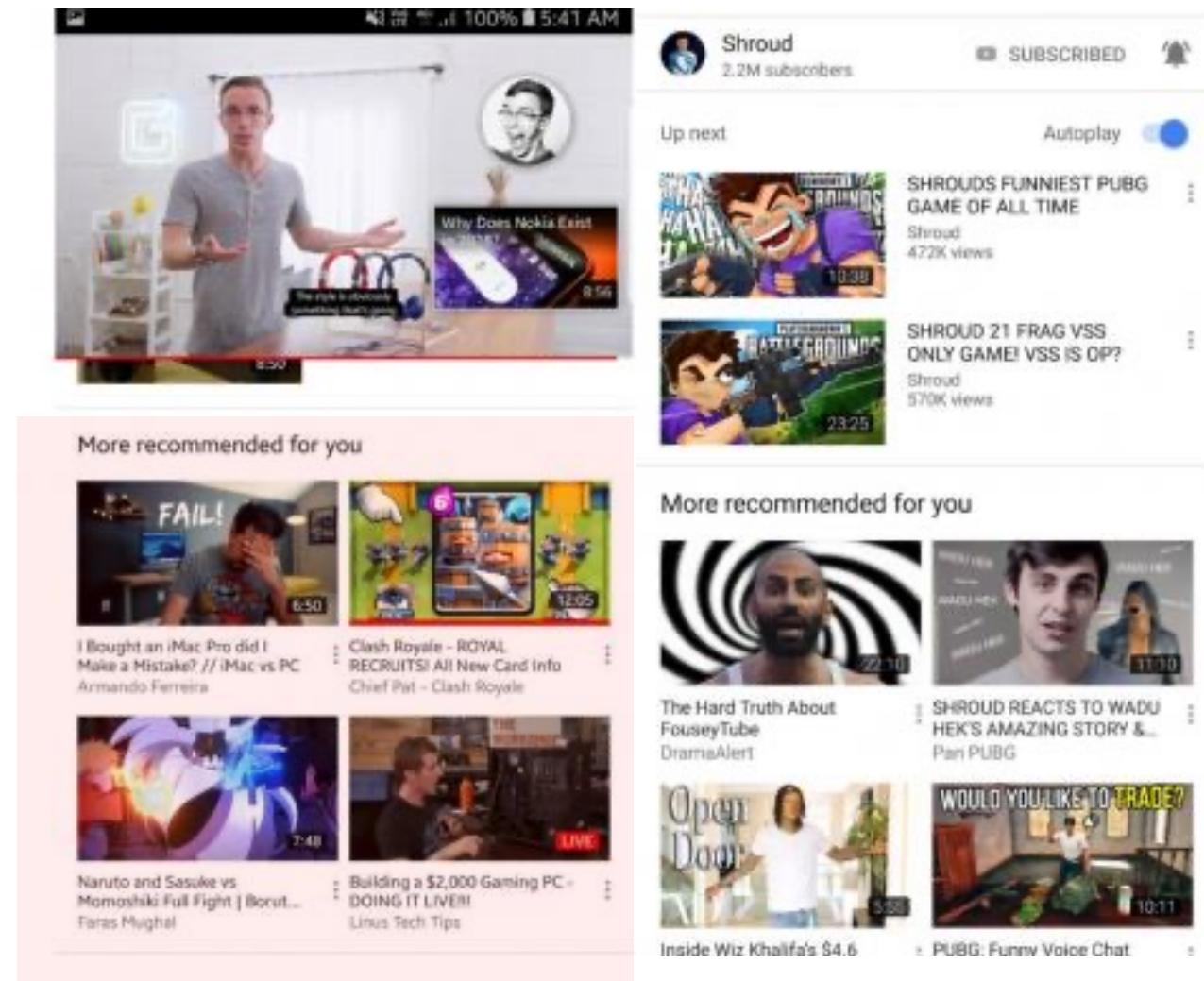


Netflix no busca una retribución directa como Amazon, lo que le interesa es **saber los gustos de los usuarios** para evitar el abandono de la suscripción mensual.

Sistemas de recomendación

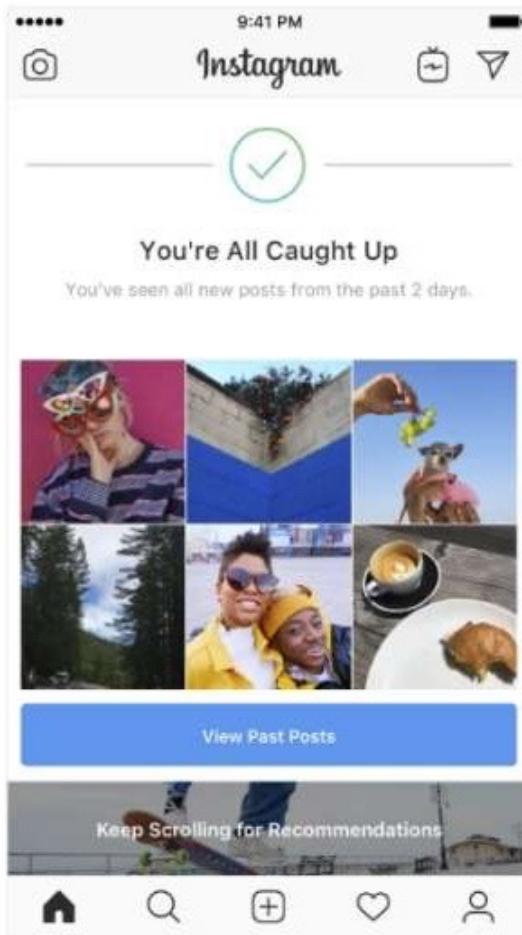
YouTube

- Añade videos relacionados para mantener la atención de los usuarios e incluir más anuncios.
- En la actualidad, separa aún más el vídeo de los comentarios, como si quisieran darle menos importancia.



Sistemas de recomendación

Instagram



- **Instagram** a través del apartado de fotos relacionadas busca mantener la atención del usuario por más tiempo. Esto implica la probabilidad de que algún anuncio capte la atención y se pueda sacar algo de valor.
- **Por otro lado**, en función de los likes, comentarios y el tiempo dedicado, descubre más cosas del usuario.

Reglas de asociación

Reglas de asociación

- Las **reglas de asociación** es un algoritmo de aprendizaje automático basado en reglas, que se utiliza para encontrar relaciones ocultas en los datos.
- Se originó con el estudio de transacciones de clientes para determinar asociaciones entre los artículos comprados. También se conoce como **análisis de afinidad**.

ID Transacciones	Lista de items en la transacción
T1	Artículo 1, Artículo 3, Artículo 4
T2	Artículo 1, Artículo 2
T3	Artículo 5, Artículo 6, Artículo 7
T4	Artículo 1, Artículo 2, Artículo 6, Artículo 10
T5	Artículo 1, Artículo 6, Artículo 8, Artículo 9



Reglas de asociación

Consiste en identificar un conjunto de patrones secuenciales en forma de reglas de tipo (Si/Entonces):



Estos patrones tienen cierta frecuencia (ocurrencia) en los datos.

Por ejemplo:

- Artículos que se compran juntos con frecuencia.
- Síntomas asociados a un diagnóstico.

Usos:

- Colocación de productos.
- Publicidad dirigida.
- Ventas.
- Cupones (compras, descuentos).

Objetivo: Aumentar las ventas y reducir los costos.

Reglas de asociación



Parte "IF" = Antecedente

Parte "THEN" = Consecuente

Itemset = conjunto de elementos. Por ejemplo: productos, artículos, eventos, operaciones, y otros que comprenden el antecedente o consecuente.

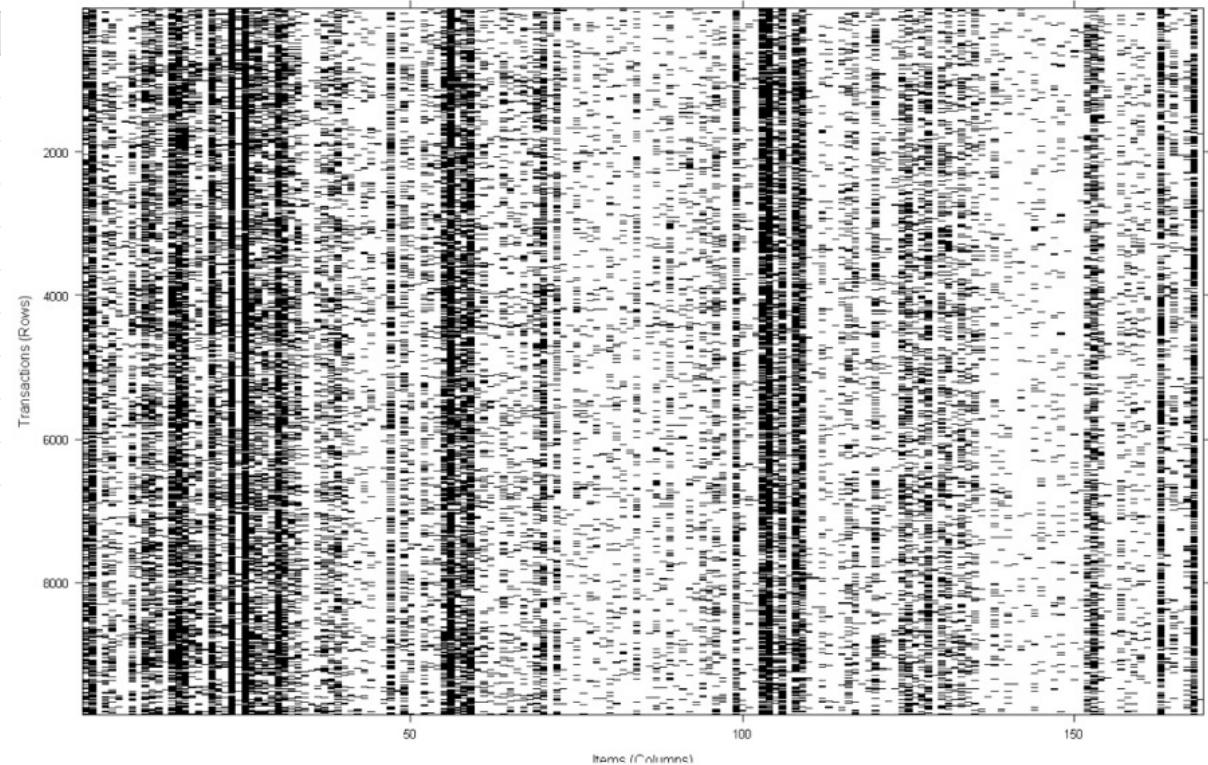
ID Transacciones	Lista de items en la transacción
T1	Artículo 1, Artículo 3, Artículo 4
T2	Artículo 1, Artículo 2
T3	Artículo 5, Artículo 6, Artículo 7
T4	Artículo 1, Artículo 2, Artículo 6, Artículo 10
T5	Artículo 1, Artículo 6, Artículo 8, Artículo 9

Reglas de asociación

Características de los datos transaccionales

- Son datos operativos.
- Se emplean para controlar y ejecutar tareas, como: ventas, transacciones, procesos y otros.
- Están altamente normalizados y se almacenan en tablas.

Row No.	Invoice	Product 10	Product 11	Product 12	Product 13	Product 14	Product 15
1	1306797	true	false	false	false	false	false
2	1306799	true	false	false	false	false	false
3	1306800	false	true	true	false	false	false
4	1306824	false	false	false	true	true	false
5	1306825	false	false	false	false	false	true
6	1306835	false	false	false	false	false	false
7	1306845	false	false	false	false	false	false
8	1306872	false	false	false	false	false	false
9	1306873	false	false	false	false	false	false
10	1306874	false	true	false	false	false	true

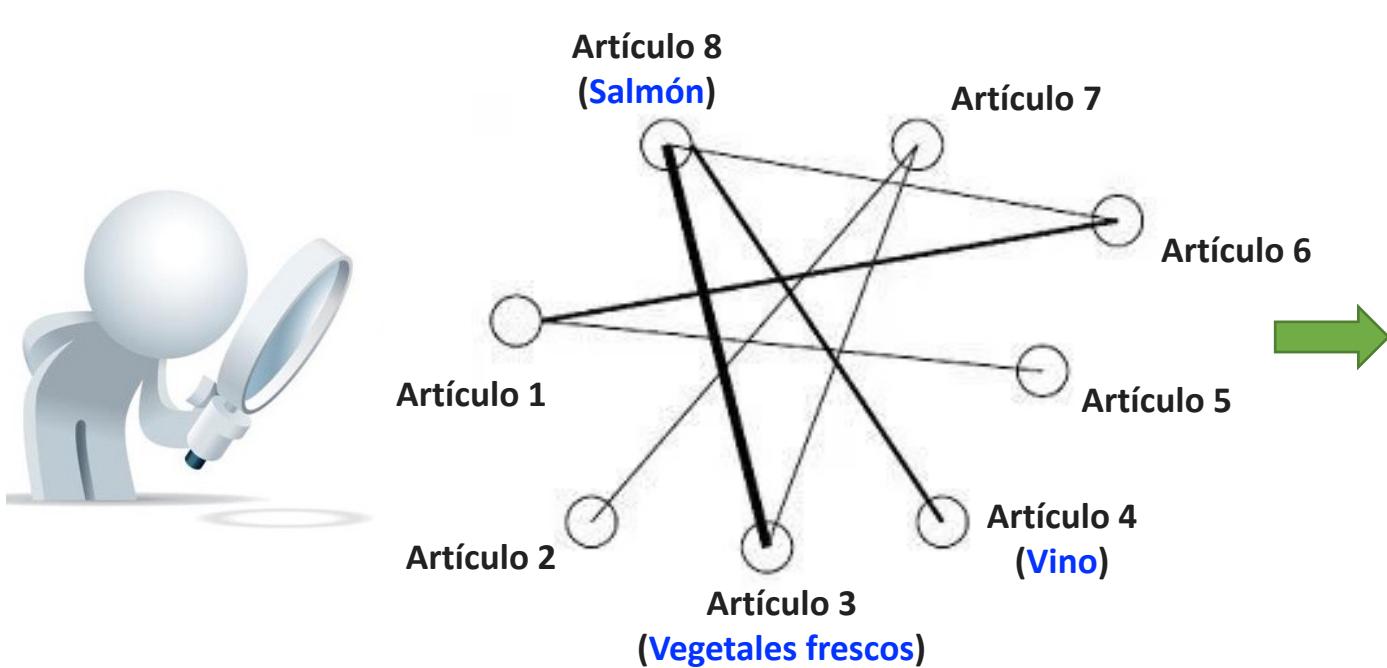


Reglas de asociación

Usos variados

Para el análisis de patrones secuenciales en diferentes áreas. Por ejemplo:

- En e-shop, las compras simultaneas,
- En un supermercado, los artículos que compran los clientes, fijar promociones, mejorar la distribución de los productos en los exhibidores.



Reglas de asociación

Usos variados

- Para el análisis de patrones de búsqueda y tráfico digital en páginas Web.
- En el mercado inmobiliario sobre el filtrado de búsqueda en la compra y renta de inmuebles.
- El análisis de tratamientos médicos de los pacientes.

The collage consists of three main sections:

- Google Analytics Dashboard:** Shows real-time user statistics (1,200 users, 1,400 sessions, 1,500 pageviews) and a chart of a hand interacting with a digital globe.
- Real Estate Platform:** A search interface for buying or renting properties. It features a search bar, a "Buscar" button, and a central image of a couple sitting on the floor surrounded by moving boxes. A blue overlay asks, "¿Te gustaría formar parte de RE/MAX?" with a "Afiliate aquí" link.
- Medical History Template:** A template titled "ised" (Instituto Superior de Estudios) for medical history. It includes sections for "Anamnesis" (Information basic about the person and the process (pathology). Registry: data of identity, motive of the consultation, history of the functional problem, personal antecedents and family, etc.), "Epígrafe" (Final analysis of the process solicited or not (posterior to treatment). Registry: diagnostic, treatment realized, recommended medication and prognosis), and "Exploración" (Physical examination of the patient. Registry: inspection, palpation, percussion, auscultation, tacto ginecológico and rectal, ophthalmoscopy and otoscopy). A QR code at the bottom right provides a link to an example of the template.

Reglas de asociación

Aprendizaje con reglas de asociación

- Una regla de asociación es una **proporción probabilística** sobre la ocurrencia de eventos presentes en el conjunto de datos.

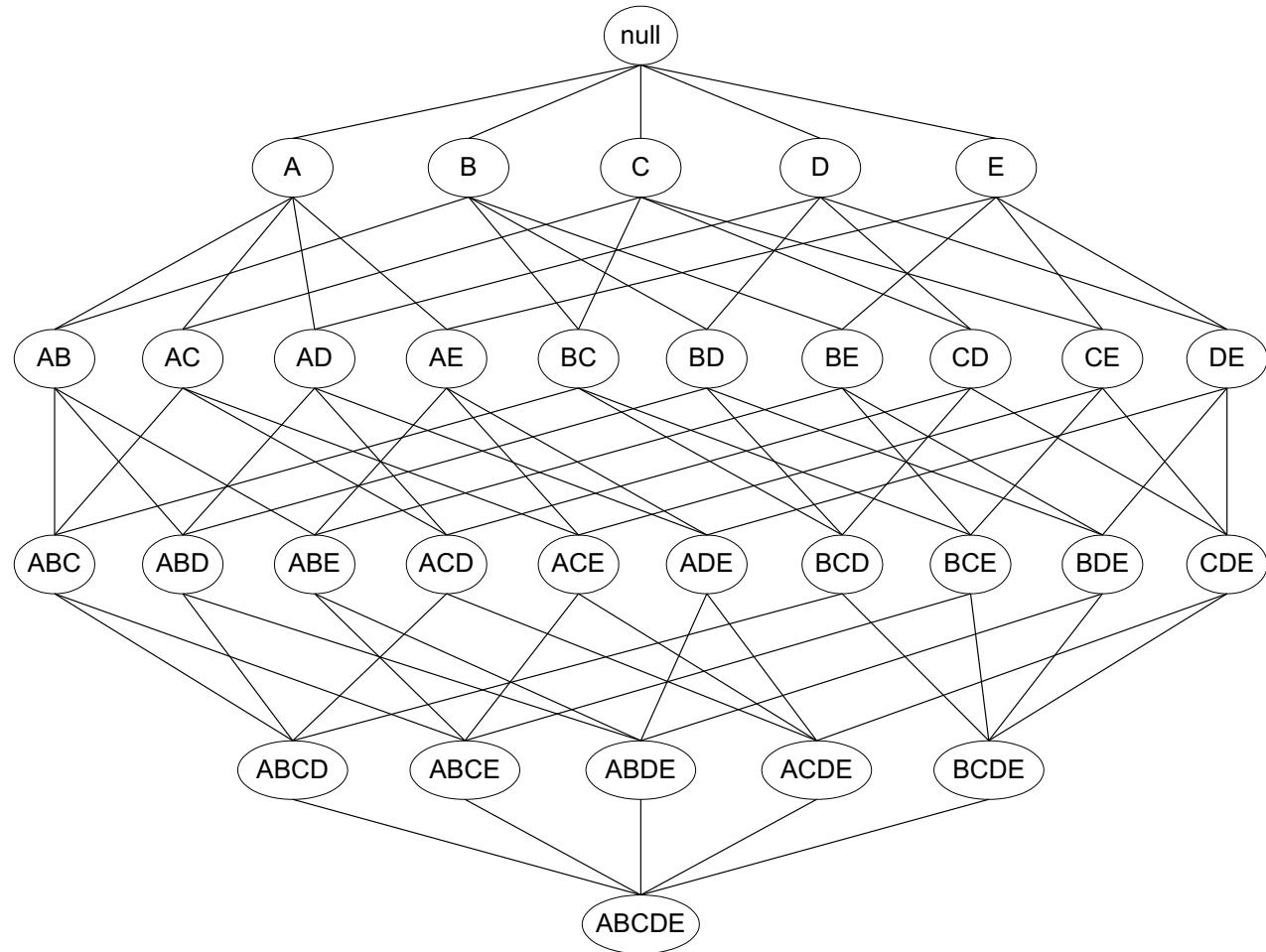
ID	Salmón	Vegetales	Vino	Tortillas
T1	1	1	0	1
T2	0	1	1	0
T3	1	0	0	1
T4	1	1	1	0
T5	0	1	0	1

Implica
Co-ocurrencia

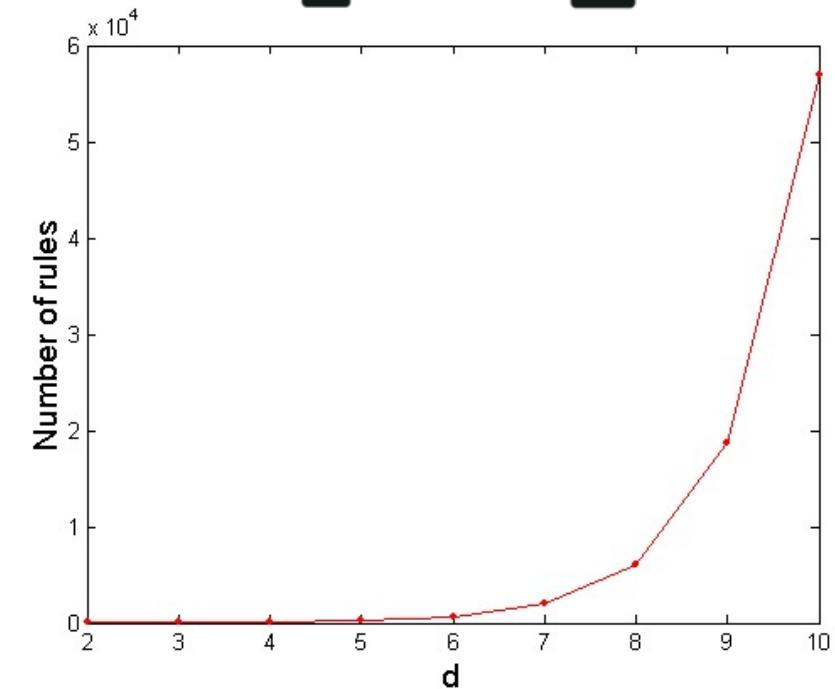


Reglas de asociación

Generación de reglas



A
IF
→
B
THEN

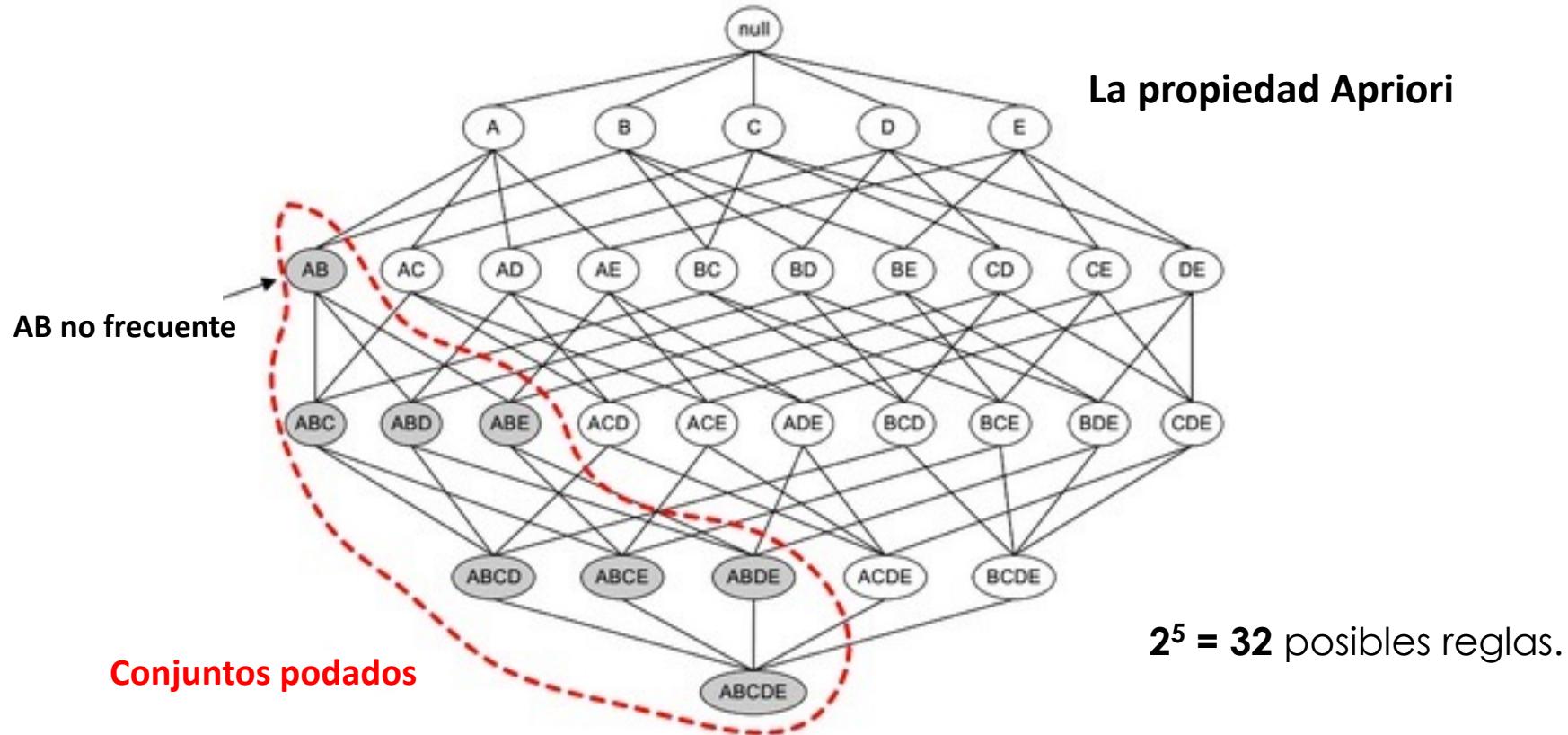


Si se tiene n elementos, hay 2^n posibles elementos candidatos (complejidad exponencial). Por ejemplo:
 $2^{10} = 1024$ posibles reglas (combinaciones).
 $2^{20} = 1'048,576$ posibles combinaciones.

Reglas de asociación

Poda

- Para eliminar las reglas menos importantes se utiliza **la poda**.
- Esto es, si un conjunto es no frecuente, entonces todas las reglas en ese conjunto también serán no frecuentes.



Algoritmo Apriori

Reglas de asociación

Algoritmo Apriori

Pseudocódigo

1. Se establece un soporte mínimo de ocurrencias.
2. Se genera una lista de **un ítem** y se seleccionan los que cumplen con el criterio de soporte mínimo.
3. Se utiliza la lista de un ítem para generar una nueva lista de **dos ítems** que cumplan con el criterio de soporte mínimo.
4. Se utiliza la lista de dos ítems para generar una lista de **tres ítems**.
5. Se continua hasta construir un conjunto con el total de ítems disponibles (k).

C_k : Candidate itemset of size k
 L_k : frequent itemset of size k

```
 $L_1 = \{\text{frequent items}\};$ 
for ( $k = 1; L_k \neq \emptyset; k++$ )
     $C_{k+1} = \text{candidates generated from } L_k;$  // Nuevos candidatos
    for each transaction  $t$  in database
        increment the count of all candidates in  $C_{k+1}$ 
        that are contained in  $t$  // candidatos contenidos en  $t$ 
    end
     $L_{k+1} = \text{candidates in } C_{k+1} \text{ with min\_support}$ 
end
return  $\cup_k L_k;$ 
```

Reglas de asociación

Algoritmo Apriori

ID	Items
T1	A, C, D
T2	B, C, E
T3	A, B, C, E
T4	B, E

Sup_Min = 2

2

Scan BDT

1-Item

Itemset	Sup
A	2
B	3
C	3
D	1
E	3

Frec 1-Item

Itemset	Sup
A	2
B	3
C	3
E	3

3

2-Items

Itemset	Sup
AB	1
AC	2
AE	1
BC	2
BE	3
CE	2

Frec 2-Items

Itemset	Sup
AC	2
BC	2
BE	3
CE	2

1

4

3-Items

Scan BDT

Itemset	Sup
ACB	1
ACE	1
ABE	1
CBE	2

Frec 3-Items

Itemset	Sup
CBE	2

$$\begin{aligned}
 AC \cup BC &= ACB \uparrow ABC \\
 AC \cup BE &= ACB \uparrow ACE \mid ABE \mid CBE \\
 AC \cup CE &= ACE \\
 BC \cup BE &= BCE \\
 BC \cup CE &= BCE \\
 BE \cup CE &= BEC \mid BCE
 \end{aligned}$$

* En cada paso se incrementa el tiempo de forma exponencial.

Reglas de asociación

Algoritmo Apriori

- Retomando la propiedad Apriori.
- Se eliminan los elementos (reglas) menos importantes.

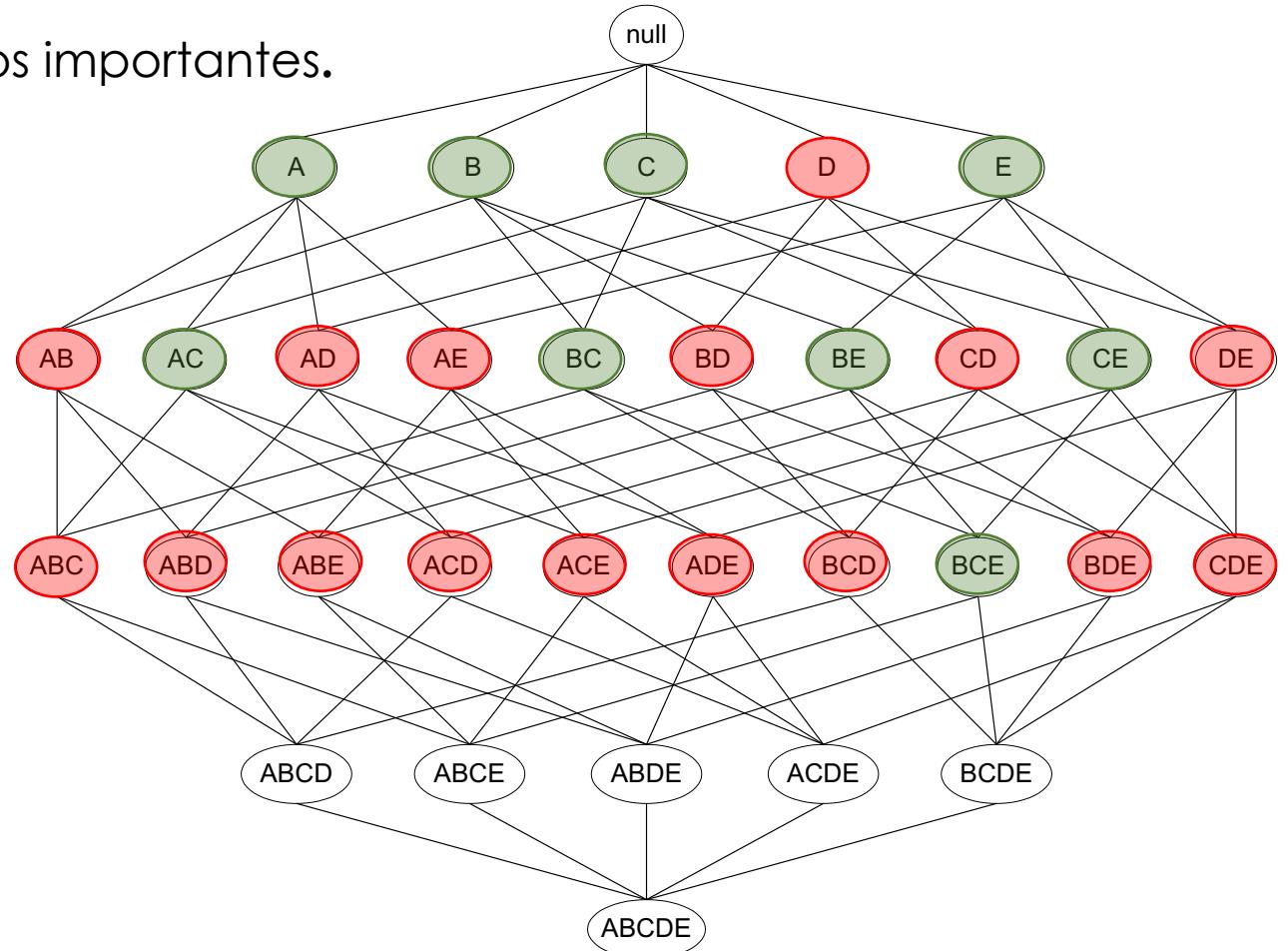
Itemset	Sup
A	2
B	3
C	3
D	1
E	3

Itemset	Sup
AB	1
AC	2
AE	1
ABE	1
BC	2
BE	3
CE	2

Itemset	Sup
ACB	1
ACE	1
ABE	1
CBE	2

Frecuencia:

- {A} {B} {C} {E}
- {AC} {BC} {BE} {CE}
- {CBE}



Reglas de asociación

Ejemplo 1

1

Se establece un soporte mínimo ocurrencias.

Transacción ID	Artículos comprados
1	{Headset-S, Laptop-S, Printer-S, Tablet-S}
2	{Monitor-S, Printer-S, Tablet-S}
3	{Headset-S, Laptop-S, Printer-S, Tablet-S}
4	{Headset-S, Laptop-S, Monitor-S, Tablet-S}
5	{Headset-S, Monitor-S, Printer-S, Tablet-S}
6	{Headset-S, Printer-S, Tablet-S}
7	{Monitor-S, Tablet-S}
8	{Laptop-S, Monitor-S, Printer-S}
9	{Headset-S, Laptop-S, Tablet-S}
10	{Printer-S, Tablet-S}

Soporte mínimo = 30%

Reglas de asociación

Ejemplo 1

2

Se genera una lista de **un ítem** que cumplan con el criterio de soporte mínimo.

Transacción ID	Artículos comprados
1	{Headset-S, Laptop-S, Printer-S, Tablet-S}
2	{Monitor-S, Printer-S, Tablet-S}
3	{Headset-S, Laptop-S, Printer-S, Tablet-S}
4	{Headset-S, Laptop-S, Monitor-S, Tablet-S}
5	{Headset-S, Monitor-S, Printer-S, Tablet-S}
6	{Headset-S, Printer-S, Tablet-S}
7	{Monitor-S, Tablet-S}
8	{Laptop-S, Monitor-S, Printer-S}
9	{Headset-S, Laptop-S, Tablet-S}
10	{Printer-S, Tablet-S}



Artículos	Conteo
Headset-S	6
Laptop-S	5
Monitor-S	5
Printer-S	7
Tablet-S	9

Soporte mínimo = 30% (mínimo de ocurrencias = 3)

Ejemplo 1

3

Se utiliza la lista de un ítem para generar una lista de **dos ítems** que cumplan con el criterio de soporte mínimo.

Artículos	Conteo
Headset-S	6
Laptop-S	5
Monitor-S	5
Printer-S	7
Tablet-S	9

Artículos	Conteo
{Headset-S, Laptop-S}	4
{Headset-S, Monitor-S}	2
{Headset-S, Printer-S}	4
{Headset-S, Tablet-S}	6
{Laptop-S, Monitor-S}	2
{Laptop-S, Printer-S}	3
{Laptop-S, Tablet-S}	4
{Monitor-S, Printer-S}	3
{Monitor-S, Tablet-S}	4
{Printer-S, Tablet-S}	6



Artículos	Conteo
{Headset-S, Laptop-S}	4
{Headset-S, Printer-S}	4
{Headset-S, Tablet-S}	6
{Laptop-S, Printer-S}	3
{Laptop-S, Tablet-S}	4
{Monitor-S, Printer-S}	3
{Monitor-S, Tablet-S}	4
{Printer-S, Tablet-S}	6

Soporte mínimo = 30% (mínimo de ocurrencias = 3)

Ejemplo 1

4

Se utiliza la lista de dos ítems para generar una lista de **tres ítems** que cumplan con el criterio de soporte mínimo.

Artículos	Conteo
{Headset-S, Laptop-S}	4
{Headset-S, Printer-S}	4
{Headset-S, Tablet-S}	6
{Laptop-S, Printer-S}	3
{Laptop-S, Tablet-S}	4
{Monitor-S, Printer-S}	3
{Monitor-S, Tablet-S}	4
{Printer-S, Tablet-S}	6

Artículos	Conteo
{Headset-S, Laptop-S, Printer-S}	2
{Headset-S, Laptop-S, Tablet-S}	4
{Headset-S, Laptop-S, Monitor-S}	1
{Headset-S, Printer-S, Tablet-S}	4
{Headset-S, Printer-S, Monitor-S}	1
{Headset-S, Tablet-S, Monitor-S}	2
{Laptop-S, Printer-S, Tablet-S}	2
{Laptop-S, Printer-S, Monitor-S}	1
{Monitor-S, Tablet-S, Printer-S}	1

Transacción ID	Artículos comprados
1	{Headset-S, Laptop-S, Printer-S, Tablet-S}
2	{Monitor-S, Printer-S, Tablet-S}
3	{Headset-S, Laptop-S, Printer-S, Tablet-S}
4	{Headset-S, Laptop-S, Monitor-S, Tablet-S}
5	{Headset-S, Monitor-S, Printer-S, Tablet-S}
6	{Headset-S, Printer-S, Tablet-S}
7	{Monitor-S, Tablet-S}
8	{Laptop-S, Monitor-S, Printer-S}
9	{Headset-S, Laptop-S, Tablet-S}
10	{Printer-S, Tablet-S}



Artículos	Conteo
{Headset-S, Laptop-S, Tablet-S}	4
{Headset-S, Printer-S, Tablet-S}	4

Soporte mínimo = 30% (mínimo de ocurrencias = 3)

Ejemplo 1

5

Se utiliza la lista de tres ítems para generar una lista de **cuatro ítems** que cumplan con el criterio de soporte mínimo.

Transacción ID	Artículos comprados
1	{Headset-S, Laptop-S, Printer-S, Tablet-S}
2	{Monitor-S, Printer-S, Tablet-S}
3	{Headset-S, Laptop-S, Printer-S, Tablet-S}
4	{Headset-S, Laptop-S, Monitor-S, Tablet-S}
5	{Headset-S, Monitor-S, Printer-S, Tablet-S}
6	{Headset-S, Printer-S, Tablet-S}
7	{Monitor-S, Tablet-S}
8	{Laptop-S, Monitor-S, Printer-S}
9	{Headset-S, Laptop-S, Tablet-S}
10	{Printer-S, Tablet-S}

Artículos	Conteo
{Headset-S, Laptop-S, Tablet-S}	4
{Headset-S, Printer-S, Tablet-S}	4



Artículos	Conteo
{Headset-S, Laptop-S, Tablet-S, Printer-S}	2

Soporte mínimo = 30% (mínimo de ocurrencias = 3)

Reglas de asociación

Obtención de reglas significativas

Mediciones para determinar reglas significativas:

- **Soporte (Cobertura).** Indica cuan importante es una regla dentro del total de transacciones.
- **Confianza.** Indica que tan fiable es una regla.
- **Lift (Elevación, Interés).** Indica el nivel de relación (aumento de probabilidad) entre el antecedente y consecuente de la regla. Lift < 1 (Relación negativa), Lift = 1 (Independientes), **Lift > 1 (Relación positiva)**

$$\begin{array}{c} \text{Rule: } X \Rightarrow Y \\ \swarrow \quad \searrow \\ \text{Support} = \frac{\text{Frequency}(X, Y)}{N} \\ \text{Confidence} = \frac{\text{Frequency}(X, Y)}{\text{Frequency}(X)} \\ \text{Lift} = \frac{\text{Support}}{\text{Support}(X) \times \text{Support}(Y)} \end{array}$$

Reglas de asociación

Retomando el ejemplo anterior

De acuerdo a la tabla determinar el soporte (Sp), la confianza (Cf) y el Lift para algunas reglas:

Transacción ID	Artículos comprados
1	{Headset-S, Laptop-S, Printer-S, Tablet-S}
2	{Monitor-S, Printer-S, Tablet-S}
3	{Headset-S, Laptop-S, Printer-S, Tablet-S}
4	{Headset-S, Laptop-S, Monitor-S, Tablet-S}
5	{Headset-S, Monitor-S, Printer-S, Tablet-S}
6	{Headset-S, Printer-S, Tablet-S}
7	{Monitor-S, Tablet-S}
8	{Laptop-S, Monitor-S, Printer-S}
9	{Headset-S, Laptop-S, Tablet-S}
10	{Printer-S, Tablet-S}

Artículos	Conteo
{Headset-S, Laptop-S, Tablet-S}	4
{Headset-S, Printer-S, Tablet-S}	4

Reglas de asociación

Retomando el ejemplo anterior

De acuerdo a la tabla determinar el soporte (Sp), la confianza (Cf) y el Lift para algunas reglas:

	Headset-S	Laptop-S	Monitor-S	Printer-S	Tablet-S
T1	1	1	0	1	1
T2	0	0	1	1	1
T3	1	1	0	1	1
T4	1	1	1	0	1
T5	1	0	1	1	1
T6	1	0	0	1	1
T7	0	0	1	0	1
T8	0	1	1	1	0
T9	1	1	0	0	1
T10	0	0	0	1	1

Artículos	Conteo
{Headset-S, Laptop-S, Tablet-S}	4
{Headset-S, Printer-S, Tablet-S}	4

Rule: $X \Rightarrow Y$

$$\text{Support} = \frac{\text{Frequency}(X, Y)}{N}$$

$$\text{Confidence} = \frac{\text{Frequency}(X, Y)}{\text{Frequency}(X)}$$

$$\text{Lift} = \frac{\text{Support}}{\text{Support}(X) \times \text{Support}(Y)}$$

- R1: Si Headset-S=1 y Laptop-S=1 Entonces Tablet-S=1
- R2: Si Headset-S=1 y Printer-S=1 Entonces Tablet-S=1
- R3: Si Tablet-S=1 Entonces Headset-S=1
- R4: Si Printer-S=1 Entonces Tablet-S=1

$\text{Sp} = 4/10 (0.4)$	$\text{Cf} = 4/4 (1)$	$\text{Lift} = (4/10)/(4/10 * 9/10) (1.11)$
$\text{Sp} = 4/10 (0.4)$	$\text{Cf} = 4/4 (1)$	$\text{Lift} = (4/10)/(4/10 * 9/10) (1.11)$
$\text{Sp} = 6/10 (0.6)$	$\text{Cf} = 6/9 (0.67)$	$\text{Lift} = (6/10)/(9/10 * 6/10) (1.11)$
$\text{Sp} = 6/10 (0.6)$	$\text{Cf} = 6/7 (0.86)$	$\text{Lift} = (6/10)/(7/10 * 9/10) (0.95)$

Reglas de asociación

Ejemplo 2

De acuerdo a la tabla determinar el soporte (Sp), la confianza (Cf) y el Lift para algunas reglas:

ID	Vino	Refresco	Tortillas	Jugo	Donas	Galletas	Aqua
T1	1	1	0	0	1	1	0
T2	0	1	1	0	0	0	0
T3	0	0	0	1	1	1	0
T4	1	1	1	1	1	1	1
T5	0	0	0	0	0	1	0
T6	1	0	0	0	0	1	1
T7	1	1	1	1	1	1	0
T8	0	0	0	1	1	1	1
T9	1	1	1	0	1	0	1
T10	0	1	0	1	1	1	0

Rule: $X \Rightarrow Y$

$$\text{Support} = \frac{\text{Frequency}(X, Y)}{N}$$

$$\text{Confidence} = \frac{\text{Frequency}(X, Y)}{\text{Frequency}(X)}$$

$$\text{Lift} = \frac{\text{Support}}{\text{Support}(X) \times \text{Support}(Y)}$$

- R1: Si Jugo=1 y Donas=1 Entonces Galletas=1
- R2: Si Donas=1 y Galletas=1 Entonces Aqua=1
- R3: Si Vino=1 y Refresco=1 Entonces Tortillas=1
- R4: Si Aqua=1 Entonces Jugo=1 y Donas=1
- R5: Si Vino=1 Entonces Jugo=1 y Galletas=1

$\text{Sp} = 5/10$ (0.5)	$\text{Cf} = 5/5$ (1)	$\text{Lift} = (5/10) / (5/10 * 8/10)$ (1.25)
$\text{Sp} =$	$\text{Cf} =$	$\text{Lift} =$
$\text{Sp} =$	$\text{Cf} =$	$\text{Lift} =$
$\text{Sp} =$	$\text{Cf} =$	$\text{Lift} =$
$\text{Sp} =$	$\text{Cf} =$	$\text{Lift} =$

Reglas de asociación

Ejemplo 2

De acuerdo a la tabla determinar el soporte (Sp), la confianza (Cf) y el Lift para algunas reglas:

ID	Vino	Refresco	Tortillas	Jugo	Donas	Galletas	Aqua
T1	1	1	0	0	1	1	0
T2	0	1	1	0	0	0	0
T3	0	0	0	1	1	1	0
T4	1	1	1	1	1	1	1
T5	0	0	0	0	0	1	0
T6	1	0	0	0	0	1	1
T7	1	1	1	1	1	1	0
T8	0	0	0	1	1	1	1
T9	1	1	1	0	1	0	1
T10	0	1	0	1	1	1	0

- R1: Si Jugo=1 y Donas=1 Entonces Galletas=1
- R2: Si Donas=1 y Galletas=1 Entonces Aqua=1
- R3: Si Vino=1 y Refresco=1 Entonces Tortillas=1
- R4: Si Aqua=1 Entonces Jugo=1 y Donas=1
- R5: Si Vino=1 Entonces Jugo=1 y Galletas=1

$$\begin{array}{lll}
 \text{Sp} = 5/10 \quad (0.5) & \text{Cf} = 5/5 \quad (1) & \text{Lift} = (5/10) / (5/10 * 8/10) \quad (1.25) \\
 \text{Sp} = 2/10 \quad (0.2) & \text{Cf} = 2/6 \quad (0.33) & \text{Lift} = (2/10) / (6/10 * 4/10) \quad (0.83) \\
 \text{Sp} = 3/10 \quad (0.3) & \text{Cf} = 3/4 \quad (0.75) & \text{Lift} = (3/10) / (4/10 * 4/10) \quad (1.15) \\
 \text{Sp} = 2/10 \quad (0.2) & \text{Cf} = 2/4 \quad (0.50) & \text{Lift} = (2/10) / (4/10 * 5/10) \quad (1) \\
 \text{Sp} = 2/10 \quad (0.2) & \text{Cf} = 2/5 \quad (0.40) & \text{Lift} = (2/10) / (5/10 * 5/10) \quad (0.8)
 \end{array}$$

Rule: $X \Rightarrow Y$

$$\begin{aligned}
 \text{Support} &= \frac{\text{Frequency}(X, Y)}{N} \\
 \text{Confidence} &= \frac{\text{Frequency}(X, Y)}{\text{Frequency}(X)} \\
 \text{Lift} &= \frac{\text{Support}}{\text{Support}(X) \times \text{Support}(Y)}
 \end{aligned}$$

Reglas de asociación

En resumen

- Un **sistema de recomendación (SR)** es el filtrado de contenido (productos, servicios o eventos) con una determinada valoración.
- Este contenido representa información de mayor interés para los usuarios, ignorando toda aquella que no sea realmente útil.
- Por otro lado, un **SR** no es un sistema que predice las compras o consumo de los usuarios, sino una recomendación.
- Si no se implementan correctamente, pueden presentar problemas que afecten a la calidad de sus recomendaciones.
- Sobre todo a aquellos clientes con **gustos atípicos** y también debido a problemas de escasez de datos.



Universidad Nacional Autónoma de México
Facultad de Ingeniería

Reglas de asociación

Práctica 1

Guillermo Molero-Castillo

guillermo.molero@ingenieria.unam.edu

Septiembre, 2021

Netflix

Sistemas de recomendación



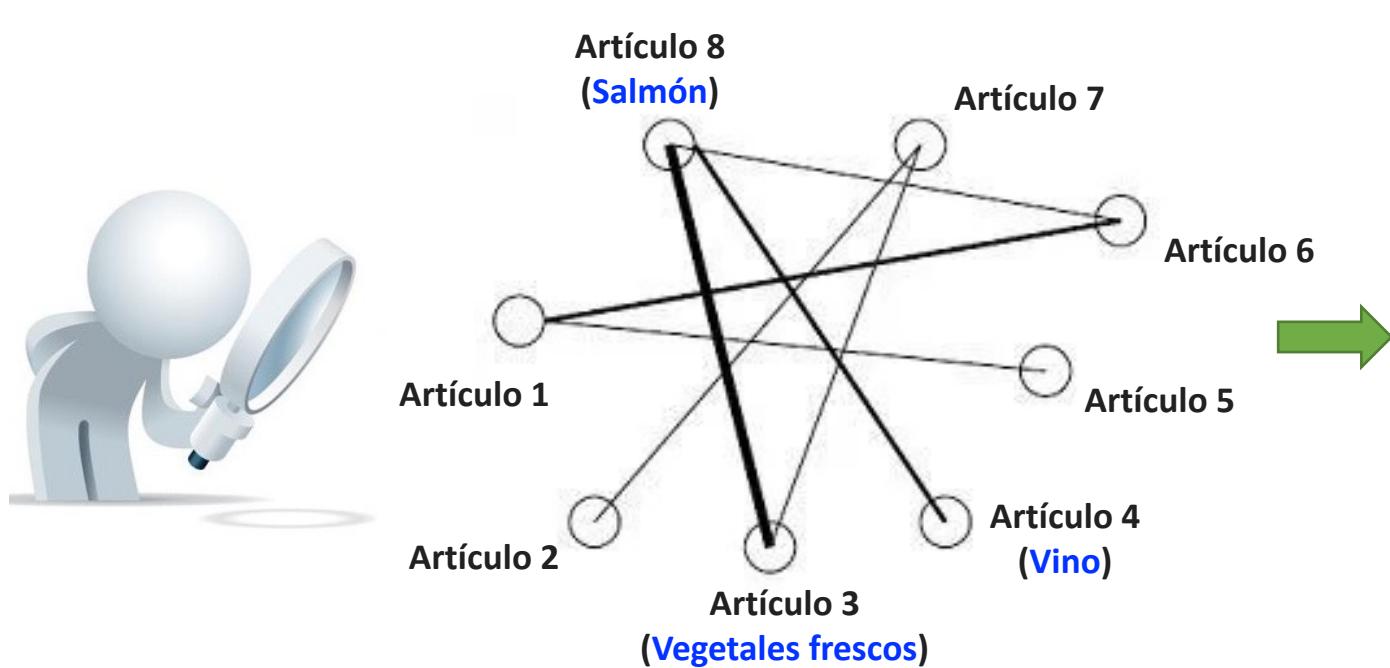
Netflix no busca una retribución directa como Amazon, lo que le interesa es **saber los gustos de los usuarios** para evitar el abandono de la suscripción mensual.

Reglas de asociación

Usos variados

Para el análisis de patrones secuenciales en diferentes áreas. Por ejemplo:

- En e-shop, las compras simultaneas,
- En un supermercado, los artículos que compran los clientes, fijar promociones, mejorar la distribución de los productos en los exhibidores.



Algoritmo Apriori

Reglas de asociación

Algoritmo Apriori

ID	Items
T1	A, C, D
T2	B, C, E
T3	A, B, C, E
T4	B, E

Sup_Min = 2

2

Scan BDT

1-Item

Itemset	Sup
A	2
B	3
C	3
D	1
E	3

Frec 1-Item

Itemset	Sup
A	2
B	3
C	3
E	3

3

2-Items

Itemset	Sup
AB	1
AC	2
AE	1
BC	2
BE	3
CE	2

Frec 2-Items

Itemset	Sup
AC	2
BC	2
BE	3
CE	2

1

4

3-Items

Scan BDT

Itemset	Sup
ACB	1
ACE	1
ABE	1
CBE	2

Frec 3-Items

Itemset	Sup
CBE	2

$$\begin{aligned}
 AC \cup BC &= ACB | ABC \\
 AC \cup BE &= ACB | ACE | ABE | CBE \\
 AC \cup CE &= ACE \\
 BC \cup BE &= BCE \\
 BC \cup CE &= BCE \\
 BE \cup CE &= BEC | BCE
 \end{aligned}$$

* En cada paso se incrementa el tiempo de forma exponencial.

Reglas de asociación

Obtención de reglas significativas

Mediciones para determinar reglas significativas:

- **Soporte (Cobertura).** Indica cuan importante es una regla dentro del total de transacciones.
- **Confianza.** Indica que tan fiable es una regla.
- **Lift (Elevación, Interés).** Indica el nivel de relación (aumento de probabilidad) entre el antecedente y consecuente de la regla. Lift < 1 (Relación negativa), Lift = 1 (Independientes), **Lift > 1 (Relación positiva)**

$$\begin{array}{c} \text{Rule: } X \Rightarrow Y \\ \swarrow \quad \searrow \\ \text{Support} = \frac{\text{Frequency}(X, Y)}{N} \\ \text{Confidence} = \frac{\text{Frequency}(X, Y)}{\text{Frequency}(X)} \\ \text{Lift} = \frac{\text{Support}}{\text{Support}(X) \times \text{Support}(Y)} \end{array}$$

Reglas de asociación

Google Colaboratory



Objetivo

Obtener reglas de asociación a partir de datos obtenidos de una plataforma de películas, donde los clientes pueden rentar o comprar este tipo de contenidos.

Características:

- Por lo general, existe un patrón en lo que ven los clientes. Por ejemplo, superhéroes en la categoría para niños.
- En este sentido, se pueden generar más ganancias, si se puede identificar la relación entre las películas. Esto es, si las películas A y B se rentan juntas, este patrón se puede aprovechar para aumentar las ganancias.
- Las personas que rentan una de estas películas pueden ser empujadas a rentar o comprar la otra, a través de campañas o sugerencias dentro de la plataforma.
- En este sentido, cada vez es común familiarizarse con los motores de recomendación en Netflix, Amazon, por nombrar los más destacados.

1. Importar las bibliotecas necesarias

```
[1] !pip install apyori # pip es un administrador de paquetes de Python. Se instala el paquete Apyori  
Collecting apyori  
  Downloading https://files.pythonhosted.org/packages/5e/62/5ffde5c473ea4b033490617ec5caa80d59804875  
Building wheels for collected packages: apyori  
  Building wheel for apyori (setup.py) ... done  
    Created wheel for apyori: filename=apyori-1.1.2-cp37-none-any.whl size=5975 sha256=0e290dfb5018672  
    Stored in directory: /root/.cache/pip/wheels/5d/92/bb/474bbadbc8c0062b9eb168f69982a0443263f8ab1711  
Successfully built apyori  
Installing collected packages: apyori  
Successfully installed apyori-1.1.2
```



```
import pandas as pd          # Para la manipulación y análisis de los datos  
import numpy as np           # Para crear vectores y matrices n dimensionales  
import matplotlib.pyplot as plt # Para la generación de gráficas a partir de los datos  
from apyori import apriori
```

2. Importar los datos

```
▶ DatosMovies = pd.read_csv('movies.csv')
DatosMovies
```

	The Revenant	13 Hours	Allied	Zootopia	Jigsaw	Achorman	Grinch	Fast and Furious	Ghostbusters	Wolverine	Mad Max	John Wick	La La Land	The Good Dinosaur	Ninja Turtles
0	Beirut	Martian	Get Out		NaN	NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN
1	Deadpool		NaN	NaN		NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN
2	X-Men	Allied		NaN		NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN
3	Ninja Turtles	Moana	Ghost in the Shell	Ralph Breaks the Internet	John Wick		NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN
4	Mad Max		NaN	NaN		NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN
...
7454	Big Sick	Looper	Hulk		NaN	NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN
7455	Beirut	Intern	Get Out	Hotel Transylvania	Mamma Mia	John Wick		NaN		NaN	NaN	NaN	NaN	NaN	NaN
7456	Captain America		NaN	NaN		NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN
7457	Green Lantern	John Wick		NaN	NaN	NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN
7458	Get Out	Inside Out	I Feel Pretty	Mad Max	NaN	NaN	NaN	NaN		NaN	NaN	NaN	NaN	NaN	NaN

7459 rows × 20 columns

2. Importar los datos

- 1) Se observa que el encabezado es la primera transacción.
- 2) 'NaN' indica que esa película no fue rentada o comprada en esa transacción.

```
▶ DatosMovies = pd.read_csv('movies.csv', header=None)
DatosMovies.head(6)
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	The Revenant	13 Hours	Allied	Zootopia	Jigsaw	Achorman	Grinch	Fast and Furious	Ghostbusters	Wolverine	Mad Max	John Wick	La La Land	The Good Dinosaur	Ninja Turtles
1	Beirut	Martian	Get Out	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	Deadpool	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	X-Men	Allied	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	Ninja Turtles	Moana	Ghost in the Shell	Ralph Breaks the Internet	John Wick	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	Mad Max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

3. Procesamiento de los datos

Exploración. Antes de ejecutar el algoritmo, es recomendable observar la distribución de la frecuencia de los elementos.

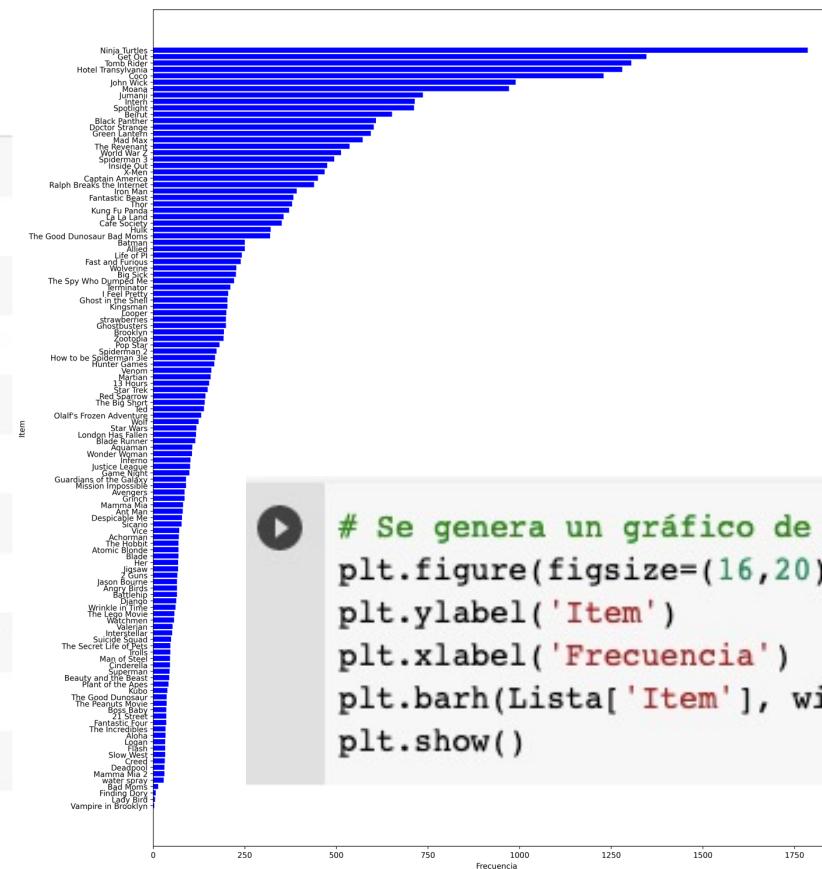
```
▶ #Se incluyen todas las transacciones en una sola lista  
Transacciones = DatosMovies.values.reshape(-1).tolist() #-1 significa 'dimensión desconocida'  
  
#Se crea una matriz (dataframe) usando la lista y se incluye una columna 'Frecuencia'  
Lista = pd.DataFrame(Transacciones)  
Lista[ 'Frecuencia' ] = 1  
  
#Se agrupa los elementos  
Lista = Lista.groupby(by=[0], as_index=False).count().sort_values(by=[ 'Frecuencia' ], ascending=True) #Conteo  
Lista[ 'Porcentaje' ] = (Lista[ 'Frecuencia' ] / Lista[ 'Frecuencia' ].sum()) #Porcentaje  
Lista = Lista.rename(columns={0 : 'Item'})  
  
#Se muestra la lista  
Lista
```

3. Procesamiento de los datos

Exploración. Antes de ejecutar el algoritmo, es recomendable observar la distribución de la frecuencia de los elementos.

	Item	Frecuencia	Porcentaje
106	Vampire in Brooklyn	3	0.000102
63	Lady Bird	5	0.000171
34	Finding Dory	7	0.000239
11	Bad Moms	14	0.000477
118	water spray	29	0.000989
...
25	Coco	1229	0.041915
44	Hotel Transylvania	1280	0.043655
103	Tomb Rider	1305	0.044507
37	Get Out	1346	0.045906
75	Ninja Turtles	1786	0.060912

119 rows × 3 columns



```
# Se genera un gráfico de barras
plt.figure(figsize=(16,20), dpi=300)
plt.ylabel('Item')
plt.xlabel('Frecuencia')
plt.barh(Lista['Item'], width=Lista['Frecuencia'], color='blue')
plt.show()
```

3. Procesamiento de los datos

La función Apriori de Python requiere que el conjunto de datos tenga la forma de una lista de listas, donde cada transacción es una lista interna dentro de una gran lista.
Los datos actuales están en un dataframe de Pandas, por lo que, se requiere convertir en una lista.

```
▶ #Se crea una lista de listas a partir del dataframe y se remueven los 'NaN'  
#level=0 especifica desde el primer índice  
MoviesLista = DatosMovies.stack().groupby(level=0).apply(list).tolist()  
MoviesLista  
  
↳ ['Fast and Furious',  
     'Spotlight'],  
    ['Jumanji', 'Ninja Turtles', 'Moana', 'Get Out', "Olalf's Frozen Adventure"],  
    ['The Revenant', 'Avengers', 'John Wick'],  
    ['Inside Out'],  
    ['Kung Fu Panda', 'Intern', 'Ninja Turtles', 'Pop Star', 'Brooklyn'],  
    ['X-Men',  
     'World War Z',  
     'Tomb Rider',  
     'Moana',
```

4. Aplicación del algoritmo Apriori

Configuración 1

Obtener reglas para aquellas películas que se hayan rentado al menos 10 veces en un día (70 veces en la semana):

- i) El soporte mínimo se calcula de $70/7460 = 0.00938$ (1%).
- ii) La confianza mínima para las reglas de 30%
- iii) La elevación de 2.

Observación. Estos valores se eligen arbitrariamente, por lo que, se recomienda probar valores y analizar la diferencia en las reglas.

Algoritmo

```
ReglasC1 = apriori(MoviesLista,  
                     min_support=0.01,  
                     min_confidence=0.3,  
                     min_lift=2)
```

4. Aplicación del algoritmo Apriori

```
▶ ResultadosC1 = list(ReglasC1)
    print(len(ResultadosC1)) #Total de reglas encontradas
```

⇨ 9

```
▶ ResultadosC1
```

```
[RelationRecord(items=frozenset({'Kung Fu Panda', 'Jumanji'}), support=0.0160857908847185, ordered_statistics=[{'support': 0.0160857908847185}], ordered_apriori_support=0.0160857908847185), RelationRecord(items=frozenset({'Tomb Rider', 'Jumanji'}), support=0.03941018766756032, ordered_statistics=[{'support': 0.03941018766756032}], ordered_apriori_support=0.03941018766756032), RelationRecord(items=frozenset({'Thor', 'Moana'}), support=0.015281501340482574, ordered_statistics=[{'support': 0.015281501340482574}], ordered_apriori_support=0.015281501340482574), RelationRecord(items=frozenset({'Terminator', 'Tomb Rider'}), support=0.01032171581769437, ordered_statistics=[{'support': 0.01032171581769437}], ordered_apriori_support=0.01032171581769437), RelationRecord(items=frozenset({'Get Out', 'Ninja Turtles', 'Jumanji'}), support=0.01018766756032171, ordered_statistics=[{'support': 0.01018766756032171}], ordered_apriori_support=0.01018766756032171), RelationRecord(items=frozenset({'Intern', 'Ninja Turtles', 'Moana'}), support=0.011126005361930294, ordered_statistics=[{'support': 0.011126005361930294}], ordered_apriori_support=0.011126005361930294), RelationRecord(items=frozenset({'Ninja Turtles', 'Moana', 'Jumanji'}), support=0.011126005361930294, ordered_statistics=[{'support': 0.011126005361930294}], ordered_apriori_support=0.011126005361930294), RelationRecord(items=frozenset({'Tomb Rider', 'Ninja Turtles', 'Jumanji'}), support=0.01715817694369, ordered_statistics=[{'support': 0.01715817694369}], ordered_apriori_support=0.01715817694369), RelationRecord(items=frozenset({'Tomb Rider', 'Spiderman 3', 'Ninja Turtles'}), support=0.010321715817694369, ordered_statistics=[{'support': 0.010321715817694369}], ordered_apriori_support=0.010321715817694369)]
```

4. Aplicación del algoritmo Apriori

pd.DataFrame(ResultadosC1)

		items	support	ordered_statistics
0	(Kung Fu Panda, Jumanji)	0.016086	[((Kung Fu Panda), (Jumanji), 0.32345013477088...	
1	(Tomb Rider, Jumanji)	0.039410	[((Jumanji), (Tomb Rider), 0.3994565217391304,...	
2	(Thor, Moana)	0.015282	[((Thor), (Moana), 0.3007915567282322, 2.31092...	
3	(Terminator, Tomb Rider)	0.010322	[((Terminator), (Tomb Rider), 0.36492890995260...	
4	(Get Out, Ninja Turtles, Jumanji)	0.010188	[((Get Out, Jumanji), (Ninja Turtles), 0.50666...	
5	(Intern, Ninja Turtles, Moana)	0.011126	[((Intern, Ninja Turtles), (Moana), 0.30970149...	
6	(Ninja Turtles, Moana, Jumanji)	0.011126	[((Moana, Jumanji), (Ninja Turtles), 0.5030303...	
7	(Tomb Rider, Ninja Turtles, Jumanji)	0.017158	[((Ninja Turtles, Jumanji), (Tomb Rider), 0.41...	
8	(Tomb Rider, Spiderman 3, Ninja Turtles)	0.010322	[((Spiderman 3, Ninja Turtles), (Tomb Rider), ...	

4. Aplicación del algoritmo Apriori

```
▶ print(ResultadosC1[0])  
  
RelationRecord(items=frozenset({'Kung Fu Panda', 'Jumanji'}), support=0.0160857908847185,
```

La primera regla contiene dos elementos: '**Kung Fu Panda**' y '**Jumanji**' que se vieron juntos.

- Esto tiene sentido, las personas que ven películas familiares, en este caso de corte infantil, suelen ver también más películas del mismo tipo, como Kung Fu Panda (2016) y Jumanji (2017).
- El soporte es de 0.016 (1.6%), la confianza de 0.32 (32%) y la elevación de **3.27**, esto representa que existe 3 veces más probabilidades de que los que vean Kung Fu Panda miren también Jumanji, o viceversa.

```
▶ print(ResultadosC1[1])  
print(ResultadosC1[2])  
  
RelationRecord(items=frozenset({'Tomb Rider', 'Jumanji'}), support=0.03941018766756032, ordered_=True  
RelationRecord(items=frozenset({'Thor', 'Moana'}), support=0.015281501340482574, ordered_statist
```

4. Aplicación del algoritmo Apriori



```
for item in ResultadosC1:  
    #El primer índice de la lista  
    Emparejar = item[0]  
    items = [x for x in Emparejar]  
    print("Regla: " + str(item[0]))  
  
    #El segundo índice de la lista  
    print("Soporte: " + str(item[1]))  
  
    #El tercer índice de la lista  
    print("Confianza: " + str(item[2][0][2]))  
    print("Lift: " + str(item[2][0][3]))  
    print("=====")
```

```
Regla: frozenset({'Kung Fu Panda', 'Jumanji'})  
Soporte: 0.0160857908847185  
Confianza: 0.3234501347708895  
Lift: 3.2784483768897226  
=====  
Regla: frozenset({'Tomb Rider', 'Jumanji'})  
Soporte: 0.03941018766756032  
Confianza: 0.3994565217391304  
Lift: 2.283483258370814  
=====  
Regla: frozenset({'Thor', 'Moana'})  
Soporte: 0.015281501340482574  
Confianza: 0.3007915567282322  
Lift: 2.3109217437617016  
=====  
Regla: frozenset({'Terminator', 'Tomb Rider'})  
Soporte: 0.01032171581769437  
Confianza: 0.36492890995260663  
Lift: 2.0861070254762035  
=====  
Regla: frozenset({'Get Out', 'Ninja Turtles', 'Jumanji'})  
Soporte: 0.010187667560321715  
Confianza: 0.5066666666666666  
Lift: 2.1163120567375886  
=====
```

4. Aplicación del algoritmo Apriori

Configuración 2

Obtener reglas para aquellas películas que se hayan visto al menos 210 veces a la semana (30 por día):

- i) El soporte mínimo se calcula de $210/7460 = 0.028$ (2.8%).
- ii) La confianza mínima para las reglas de 30%.
- iii) La elevación mayor a 1.

Algoritmo

```
▶ ReglasC2 = apriori(MoviesLista,
                      min_support=0.028,
                      min_confidence=0.3,
                      min_lift = 1.01)
```

4. Aplicación del algoritmo Apriori



```
ResultadosC2 = list(ReglasC2)
print(len(ResultadosC2))
```

8



```
ResultadosC2
```

```
[RelationRecord(items=frozenset({'Get Out', 'Beirut'}), support=0.028954423592493297, ordered_statistics
RelationRecord(items=frozenset({'Ninja Turtles', 'Coco'}), support=0.05294906166219839, ordered_statist
RelationRecord(items=frozenset({'Intern', 'Ninja Turtles'}), support=0.035924932975871314, ordered_stat
RelationRecord(items=frozenset({'Ninja Turtles', 'Jumanji'}), support=0.04115281501340483, ordered_stat
RelationRecord(items=frozenset({'Tomb Rider', 'Jumanji'}), support=0.03941018766756032, ordered_statist
RelationRecord(items=frozenset({'Ninja Turtles', 'Moana'}), support=0.04825737265415549, ordered_statis
RelationRecord(items=frozenset({'Spotlight', 'Ninja Turtles'}), support=0.0339142091152815, ordered_st
RelationRecord(items=frozenset({'Tomb Rider', 'Ninja Turtles'}), support=0.060053619302949064, ordered_
```

4. Aplicación del algoritmo Apriori



```
pd.DataFrame(ResultadosC2)
```



items support

ordered_statistics

0	(Get Out, Beirut)	0.028954	[((Beirut), (Get Out), 0.3312883435582822, 1.8...
1	(Ninja Turtles, Coco)	0.052949	[((Coco), (Ninja Turtles), 0.32166123778501626...
2	(Intern, Ninja Turtles)	0.035925	[((Intern), (Ninja Turtles), 0.375350140056022...
3	(Ninja Turtles, Jumanji)	0.041153	[((Jumanji), (Ninja Turtles), 0.41711956521739...
4	(Tomb Rider, Jumanji)	0.039410	[((Jumanji), (Tomb Rider), 0.3994565217391304,...
5	(Ninja Turtles, Moana)	0.048257	[((Moana), (Ninja Turtles), 0.3707518022657054...
6	(Spotlight, Ninja Turtles)	0.033914	[((Spotlight), (Ninja Turtles), 0.355337078651...
7	(Tomb Rider, Ninja Turtles)	0.060054	[((Tomb Rider), (Ninja Turtles), 0.34329501915...

4. Aplicación del algoritmo Apriori

```
▶ print(ResultadosC2[0])  
  
RelationRecord(items=frozenset({'Get Out', 'Beirut'}), support=0.028954423592493297,
```

La primera regla contiene dos elementos: '**Beirut**' y '**Get Out**' que se vieron juntos.

- Esto también tiene sentido, las personas que ven películas de espionaje, como Beirut (2018), podrían tener gustos afines con películas de terror, como Get Out (2017).
- El soporte es de 0.028 (2.8%), la confianza de 0.33 (33%) y una elevación de 1.83, esto representa que existe casi 2 veces más probabilidades de que los que vean Beirut miren también Get Out, o viceversa.

```
▶ print(ResultadosC2[1])  
print(ResultadosC2[2])  
  
RelationRecord(items=frozenset({'Ninja Turtles', 'Coco'}), support=0.05294906166219839, ord  
RelationRecord(items=frozenset({'Intern', 'Ninja Turtles'}), support=0.035924932975871314,
```

4. Aplicación del algoritmo Apriori

```
▶ for item in ResultadosC2:  
    #El primer índice de la lista  
    Emparejar = item[0]  
    items = [x for x in Emparejar]  
    print("Regla: " + str(item[0]))  
  
    #El segundo índice de la lista  
    print("Soporte: " + str(item[1]))  
  
    #El tercer índice de la lista  
    print("Confianza: " + str(item[2][0][2]))  
    print("Lift: " + str(item[2][0][3]))  
    print("=====")
```

```
Regla: frozenset({'Get Out', 'Beirut'})  
Soporte: 0.028954423592493297  
Confianza: 0.3312883435582822  
Lift: 1.8361151879233173  
=====  
Regla: frozenset({'Ninja Turtles', 'Coco'})  
Soporte: 0.05294906166219839  
Confianza: 0.32166123778501626  
Lift: 1.3435570178478284  
=====  
Regla: frozenset({'Intern', 'Ninja Turtles'})  
Soporte: 0.035924932975871314  
Confianza: 0.3753501400560224  
Lift: 1.5678118951948081  
=====  
Regla: frozenset({'Ninja Turtles', 'Jumanji'})  
Soporte: 0.04115281501340483  
Confianza: 0.4171195652173913  
Lift: 1.742279930863236  
=====  
Regla: frozenset({'Tomb Rider', 'Jumanji'})  
Soporte: 0.03941018766756032  
Confianza: 0.3994565217391304  
Lift: 2.283483258370814  
=====
```



Universidad Nacional Autónoma de México
Facultad de Ingeniería

Reglas de asociación Práctica 2

Guillermo Molero-Castillo
guillermo.molero@ingenieria.unam.edu

Septiembre, 2021

Objetivo

Analizar las transacciones y obtener reglas significativas (patrones) de los productos vendidos en un comercio minorista en Francia. Los datos son transacciones de un comercio de un periodo de una semana (7 días).

Características:

- Ítems (20 productos)
- 7500 transacciones

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole wheat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	chutney	Nan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

1. Importar las bibliotecas necesarias

```
[1] !pip install apyori # pip es un administrador de paquetes de Python. Se instala el paquete Apyori  
Collecting apyori  
  Downloading https://files.pythonhosted.org/packages/5e/62/5ffde5c473ea4b033490617ec5caa80d59804875  
Building wheels for collected packages: apyori  
  Building wheel for apyori (setup.py) ... done  
    Created wheel for apyori: filename=apyori-1.1.2-cp37-none-any.whl size=5975 sha256=0e290dfb5018672  
    Stored in directory: /root/.cache/pip/wheels/5d/92/bb/474bbadbc8c0062b9eb168f69982a0443263f8ab1711  
Successfully built apyori  
Installing collected packages: apyori  
Successfully installed apyori-1.1.2
```



```
import pandas as pd          # Para la manipulación y análisis de los datos  
import numpy as np           # Para crear vectores y matrices n dimensionales  
import matplotlib.pyplot as plt # Para la generación de gráficas a partir de los datos  
from apyori import apriori
```

2. Importar los datos

```
▶ from google.colab import files  
files.upload()
```

```
▶ DatosTransacciones = pd.read_csv('store_data.csv')  
DatosTransacciones
```

▶

	shrimp	almonds	avocado	vegetables mix	green grapes	whole wheat flour	yams	cottage cheese	energy drink
0	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN
1	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN
4	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...

2. Importar los datos

Observaciones:

- 1) Se observa que el encabezado es la primera transacción.
- 2) NaN indica que esa película no fue rentada o comprada en esa transacción.

```
▶ DatosTransacciones = pd.read_csv('store_data.csv', header=None)  
DatosTransacciones.head(5)
```

	0	1	2	3	4	5	6	7	8	9	10
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole wheat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN

3. Procesamiento de los datos

Exploración. Antes de ejecutar el algoritmo, es recomendable observar la distribución de la frecuencia de los elementos.

```
#Se incluyen todas las transacciones en una sola lista
Transacciones = DatosTransacciones.values.reshape(-1).tolist() #-1 significa 'dimensión desconocida'

#Se crea una matriz (dataframe) usando la lista y se incluye una columna 'Frecuencia'
Lista = pd.DataFrame(Transacciones)
Lista[ 'Frecuencia' ] = 1

#Se agrupa los elementos
Lista = Lista.groupby(by=[0], as_index=False).count().sort_values(by=[ 'Frecuencia' ], ascending=True) #Conteo
Lista[ 'Porcentaje' ] = (Lista[ 'Frecuencia' ] / Lista[ 'Frecuencia' ].sum()) #Porcentaje
Lista = Lista.rename(columns={0 : 'Item'})

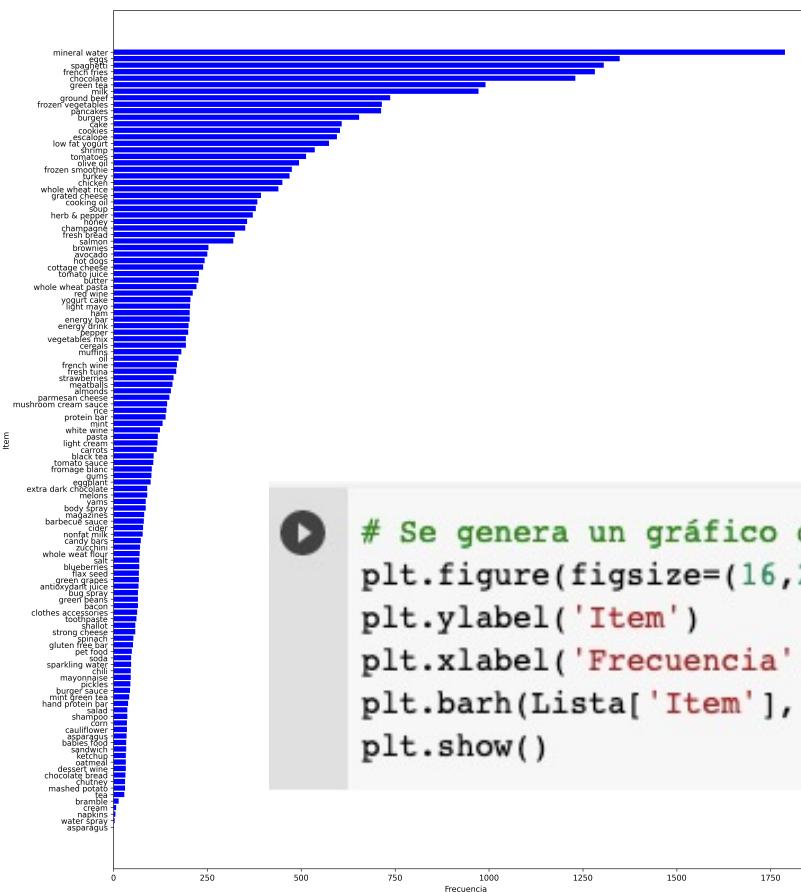
#Se muestra la lista
Lista
```

3. Procesamiento de los datos

Exploración. Antes de ejecutar el algoritmo, es recomendable observar la distribución de la frecuencia de los elementos.

	Item	Frecuencia	Porcentaje
0	asparagus	1	0.000034
112	water spray	3	0.000102
77	napkins	5	0.000170
34	cream	7	0.000238
11	bramble	14	0.000477
...
25	chocolate	1230	0.041889
43	french fries	1282	0.043660
100	spaghetti	1306	0.044478
37	eggs	1348	0.045908
72	mineral water	1788	0.060893

120 rows x 3 columns



```
# Se genera un gráfico de barras
plt.figure(figsize=(16,20), dpi=300)
plt.ylabel('Item')
plt.xlabel('Frecuencia')
plt.barh(Lista['Item'], width=Lista['Frecuencia'], color='blue')
plt.show()
```

3. Procesamiento de los datos

La función Apriori de Python requiere que el conjunto de datos tenga la forma de una lista de listas, donde cada transacción es una lista interna dentro de una gran lista. Los datos actuales están en un dataframe de Pandas, por lo que, se requiere convertir en una lista.

```
#Se crea una lista de listas a partir del dataframe y se remueven los 'NaN'  
#level=0 especifica desde el primer índice  
TransaccionesLista = DatosTransacciones.stack().groupby(level=0).apply(list).tolist()  
TransaccionesLista
```

['green tea'],
['cookies'],
['french fries', 'cookies'],
['milk', 'butter', 'eggs'],
['eggs', 'mushroom cream sauce', 'low fat yogurt'],
['eggs', 'green tea'],
['mineral water', 'whole wheat rice'],
['shrimp',
'frozen vegetables',
'parmesan cheese',
'mineral water']

4. Aplicación del algoritmo Apriori

Configuración 1

Obtener reglas para aquellos artículos que se compran al menos 5 veces al día, entonces, $5 \times 7 = 35$ veces en una semana, entonces:

- i) El soporte mínimo se calcula de $35/7500 = 0.0045$ (0.45%).
- ii) La confianza mínima para las reglas de 20%.
- iii) La elevación de 3.

Algoritmo



```
ReglasC1 = apriori(TransaccionesLista,
                     min_support=0.0045,
                     min_confidence=0.2,
                     min_lift=3)
```

4. Aplicación del algoritmo Apriori



```
ResultadosC1 = list(ReglasC1)
print(len(ResultadosC1)) #Total de reglas encontradas
```

24



ResultadosC1

```
[RelationRecord(items=frozenset({'chicken', 'light cream'}), support=0.004532728969470737, ordered_statistics=[0.004532728969470737], ordered_statistics_min=0.004532728969470737, ordered_statistics_max=0.004532728969470737, ordered_statistics_mean=0.004532728969470737, ordered_statistics_stdev=0.0), RelationRecord(items=frozenset({'escalope', 'mushroom cream sauce'}), support=0.005732568990801226, ordered_statistics=[0.005732568990801226], ordered_statistics_min=0.005732568990801226, ordered_statistics_max=0.005732568990801226, ordered_statistics_mean=0.005732568990801226, ordered_statistics_stdev=0.0), RelationRecord(items=frozenset({'pasta', 'escalope'}), support=0.005865884548726837, ordered_statistics=[0.005865884548726837], ordered_statistics_min=0.005865884548726837, ordered_statistics_max=0.005865884548726837, ordered_statistics_mean=0.005865884548726837, ordered_statistics_stdev=0.0), RelationRecord(items=frozenset({'herb & pepper', 'ground beef'}), support=0.015997866951073192, ordered_statistics=[0.015997866951073192], ordered_statistics_min=0.015997866951073192, ordered_statistics_max=0.015997866951073192, ordered_statistics_mean=0.015997866951073192, ordered_statistics_stdev=0.0), RelationRecord(items=frozenset({'ground beef', 'tomato sauce'}), support=0.005332622317024397, ordered_statistics=[0.005332622317024397], ordered_statistics_min=0.005332622317024397, ordered_statistics_max=0.005332622317024397, ordered_statistics_mean=0.005332622317024397, ordered_statistics_stdev=0.0), RelationRecord(items=frozenset({'whole wheat pasta', 'olive oil'}), support=0.007998933475536596, ordered_statistics=[0.007998933475536596], ordered_statistics_min=0.007998933475536596, ordered_statistics_max=0.007998933475536596, ordered_statistics_mean=0.007998933475536596, ordered_statistics_stdev=0.0), RelationRecord(items=frozenset({'pasta', 'shrimp'}), support=0.005065991201173177, ordered_statistics=[0.005065991201173177], ordered_statistics_min=0.005065991201173177, ordered_statistics_max=0.005065991201173177, ordered_statistics_mean=0.005065991201173177, ordered_statistics_stdev=0.0), RelationRecord(items=frozenset({'shrimp', 'chocolate', 'frozen vegetables'}), support=0.005332622317024397, ordered_statistics=[0.005332622317024397], ordered_statistics_min=0.005332622317024397, ordered_statistics_max=0.005332622317024397, ordered_statistics_mean=0.005332622317024397, ordered_statistics_stdev=0.0), RelationRecord(items=frozenset({'ground beef', 'spaghetti', 'cooking oil'}), support=0.004799360085321957, ordered_statistics=[0.004799360085321957], ordered_statistics_min=0.004799360085321957, ordered_statistics_max=0.004799360085321957, ordered_statistics_mean=0.004799360085321957, ordered_statistics_stdev=0.0), RelationRecord(items=frozenset({'ground beef', 'spaghetti', 'frozen vegetables'}), support=0.0086655112651, ordered_statistics=[0.0086655112651], ordered_statistics_min=0.0086655112651, ordered_statistics_max=0.0086655112651, ordered_statistics_mean=0.0086655112651, ordered_statistics_stdev=0.0), RelationRecord(items=frozenset({'olive oil', 'milk', 'frozen vegetables'}), support=0.004799360085321957, ordered_statistics=[0.004799360085321957], ordered_statistics_min=0.004799360085321957, ordered_statistics_max=0.004799360085321957, ordered_statistics_mean=0.004799360085321957, ordered_statistics_stdev=0.0), RelationRecord(items=frozenset({'shrimp', 'frozen vegetables', 'mineral water'}), support=0.00719904012798, ordered_statistics=[0.00719904012798], ordered_statistics_min=0.00719904012798, ordered_statistics_max=0.00719904012798, ordered_statistics_mean=0.00719904012798, ordered_statistics_stdev=0.0), RelationRecord(items=frozenset({'olive oil', 'spaghetti', 'frozen vegetables'}), support=0.005732568990801, ordered_statistics=[0.005732568990801], ordered_statistics_min=0.005732568990801, ordered_statistics_max=0.005732568990801, ordered_statistics_mean=0.005732568990801, ordered_statistics_stdev=0.0), RelationRecord(items=frozenset({'spaghetti', 'shrimp', 'frozen vegetables'}), support=0.005999200106652446, ordered_statistics=[0.005999200106652446], ordered_statistics_min=0.005999200106652446, ordered_statistics_max=0.005999200106652446, ordered_statistics_mean=0.005999200106652446, ordered_statistics_stdev=0.0), RelationRecord(items=frozenset({'tomatoes', 'spaghetti', 'frozen vegetables'}), support=0.0066657778962804, ordered_statistics=[0.0066657778962804], ordered_statistics_min=0.0066657778962804, ordered_statistics_max=0.0066657778962804, ordered_statistics_mean=0.0066657778962804, ordered_statistics_stdev=0.0), RelationRecord(items=frozenset({'grated cheese', 'spaghetti', 'ground beef'}), support=0.005332622317024397, ordered_statistics=[0.005332622317024397], ordered_statistics_min=0.005332622317024397, ordered_statistics_max=0.005332622317024397, ordered_statistics_mean=0.005332622317024397, ordered_statistics_stdev=0.0), RelationRecord(items=frozenset({'herb & pepper', 'ground beef', 'mineral water'}), support=0.0066657778962, ordered_statistics=[0.0066657778962], ordered_statistics_min=0.0066657778962, ordered_statistics_max=0.0066657778962, ordered_statistics_mean=0.0066657778962, ordered_statistics_stdev=0.0)]
```

4. Aplicación del algoritmo Apriori



```
print(ResultadosC1[0])  
  
RelationRecord(items=frozenset({'chicken', 'light cream'}), support=0.004532728969470737, ·
```

La primera regla contiene dos elementos: **chicken** y **light cream** que comúnmente se compran juntos.

- Esto tiene sentido, las personas que compran **crema ligera** tienen cuidado con lo que comen, por lo que, es probable que compren **pollo**, en lugar de carne roja.
- El soporte es de 0.0045, la confianza de 0.2905, la elevación de 4.84, esto es, **4.84** veces más probabilidades de que compren crema ligera.



```
print(ResultadosC1[1])  
print(ResultadosC1[2])  
  
⇒ RelationRecord(items=frozenset({'escalope', 'mushroom cream sauce'}), support=0.005732568990801226,  
RelationRecord(items=frozenset({'pasta', 'escalope'}), support=0.005865884548726837, ordered_statis
```

4. Aplicación del algoritmo Apriori



```
for item in ResultadosC1:  
    #El primer índice de la lista  
    Emparejar = item[0]  
    items = [x for x in Emparejar]  
    print("Regla: " + str(item[0]))  
  
    #El segundo índice de la lista  
    print("Soporte: " + str(item[1]))  
  
    #El tercer índice de la lista  
    print("Confianza: " + str(item[2][0][2]))  
    print("Lift: " + str(item[2][0][3]))  
    print("=====")
```

```
⇒ Regla: frozenset({'chicken', 'light cream'})  
Soporte: 0.004532728969470737  
Confianza: 0.29059829059829057  
Lift: 4.84395061728395  
=====  
Regla: frozenset({'escalope', 'mushroom cream sauce'})  
Soporte: 0.005732568990801226  
Confianza: 0.3006993006993007  
Lift: 3.790832696715049  
=====  
Regla: frozenset({'pasta', 'escalope'})  
Soporte: 0.005865884548726837  
Confianza: 0.3728813559322034  
Lift: 4.700811850163794  
=====  
Regla: frozenset({'herb & pepper', 'ground beef'})  
Soporte: 0.015997866951073192  
Confianza: 0.3234501347708895  
Lift: 3.2919938411349285  
=====  
Regla: frozenset({'ground beef', 'tomato sauce'})  
Soporte: 0.005332622317024397  
Confianza: 0.3773584905660377  
Lift: 3.840659481324083  
=====
```

4. Aplicación del algoritmo Apriori

Configuración 2

Obtener reglas para aquellos artículos que se compran al menos 30 veces al día, entonces, $30 \times 7 = 210$ veces en una semana, entonces:

- i) El soporte mínimo se calcula de $210/7500 = 0.028$ (2.8%).
- ii) La confianza mínima para las reglas de 25%.
- iii) La elevación mayor a 1.

Algoritmo



```
ReglasC2 = apriori(TransaccionesLista,
                     min_support=0.028,
                     min_confidence=0.25,
                     min_lift = 1.01)
```

4. Aplicación del algoritmo Apriori

```
▶ ResultadosC2 = list(ReglasC2)
print(len(ResultadosC2))
```

10

```
▶ ResultadosC2
```

```
[RelationRecord(items=frozenset({'burgers', 'eggs'}), support=0.02879616051193174, ordered_statistics=
RelationRecord(items=frozenset({'chocolate', 'mineral water'}), support=0.05265964538061592, ordered_
RelationRecord(items=frozenset({'eggs', 'mineral water'}), support=0.05092654312758299, ordered_stati
RelationRecord(items=frozenset({'frozen vegetables', 'mineral water'}), support=0.03572856952406346,
RelationRecord(items=frozenset({'ground beef', 'mineral water'}), support=0.040927876283162246, order
RelationRecord(items=frozenset({'ground beef', 'spaghetti'}), support=0.03919477403012932, ordered_st
RelationRecord(items=frozenset({'milk', 'mineral water'}), support=0.04799360085321957, ordered_stati
RelationRecord(items=frozenset({'spaghetti', 'milk'}), support=0.03546193840821224, ordered_statistic
RelationRecord(items=frozenset({'pancakes', 'mineral water'}), support=0.03372883615517931, ordered_
RelationRecord(items=frozenset({'spaghetti', 'mineral water'}), support=0.05972536995067324, ordered_
```

4. Aplicación del algoritmo Apriori

```
▶ print(ResultadosC2[0])  
RelationRecord(items=frozenset({'burgers', 'eggs'}), support=0.02879616051193174, ordered_statistics=
```

La primera regla contiene dos elementos: **hamburguesas** y **huevos** que comúnmente se compran juntos.

- Tiene sentido, algunas personas que compran hamburguesas consumen también huevos, como comida de preparación rápida.
- El soporte es de 0.028 (2.8%), la confianza de 0.33 (33%), la elevación de 1.83, esto es, hay casi 2 veces más probabilidades de que cuando se compre hamburguesas se compre también huevos.

4. Aplicación del algoritmo Apriori

```
▶ for item in ResultadosC2:  
    #El primer índice de la lista  
    Emparejar = item[0]  
    items = [x for x in Emparejar]  
    print("Regla: " + str(item[0]))  
  
    #El segundo índice de la lista  
    print("Soporte: " + str(item[1]))  
  
    #El tercer índice de la lista  
    print("Confianza: " + str(item[2][0][2]))  
    print("Lift: " + str(item[2][0][3]))  
    print("=====")
```

```
Regla: frozenset({'burgers', 'eggs'})  
Soporte: 0.02879616051193174  
Confianza: 0.33027522935779813  
Lift: 1.8378297443715457  
=====  
Regla: frozenset({'chocolate', 'mineral water'})  
Soporte: 0.05265964538061592  
Confianza: 0.3213995117982099  
Lift: 1.3483320682317521  
=====  
Regla: frozenset({'eggs', 'mineral water'})  
Soporte: 0.05092654312758299  
Confianza: 0.28338278931750743  
Lift: 1.188844688294532  
=====  
Regla: frozenset({'frozen vegetables', 'mineral water'})  
Soporte: 0.03572856952406346  
Confianza: 0.37482517482517486  
Lift: 1.57246288387228  
=====  
Regla: frozenset({'ground beef', 'mineral water'})  
Soporte: 0.040927876283162246  
Confianza: 0.41655359565807326  
Lift: 1.7475215442008991  
=====
```

Reglas de asociación

Conclusión

- Los algoritmos como Apriori son útiles para encontrar relaciones entre transacciones.
- Son fáciles de implementar y tienen una gran capacidad de explicación.
- No obstante, para conocimientos más avanzados en los sistemas de recomendación, como los utilizados por Google o Amazon, se utilizan este y otros algoritmos.

Reglas de asociación

Mejoras

Hay técnicas adicionales que se pueden aplicar a Apriori para mejorar la eficiencia. Algunos de estos son:

- **Transaction reduction:** para eliminar las transacciones poco frecuentes.
- **Partitioning:** los items posiblemente frecuentes deben ser frecuentes en una de las particiones.
- **Dynamic Itemset Counting:** para reducir el número de recuento sobre los datos.
- **Sampling:** para recoger muestras aleatorias.
- **Hashing:** para reducir el análisis de la base de datos.

Sugerencia:

- Revisar los algoritmo FP Growth y Eclat, los cuales pueden ser más eficiente con respecto a Apriori.



Universidad Nacional Autónoma de México
Facultad de Ingeniería

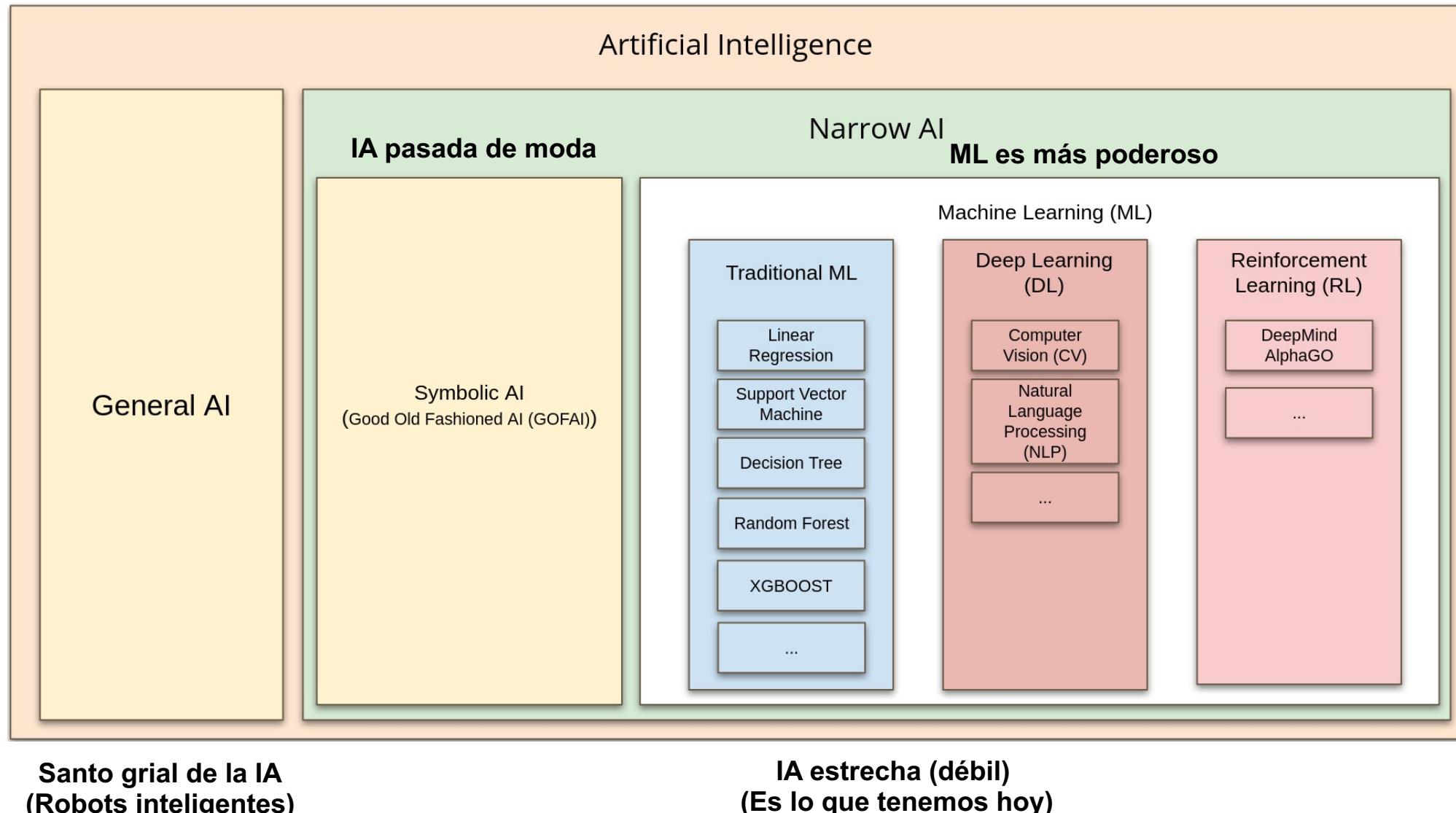
Métricas de distancia (Funciones de distancia)

Guillermo Molero-Castillo

guillermo.molero@ingenieria.unam.edu

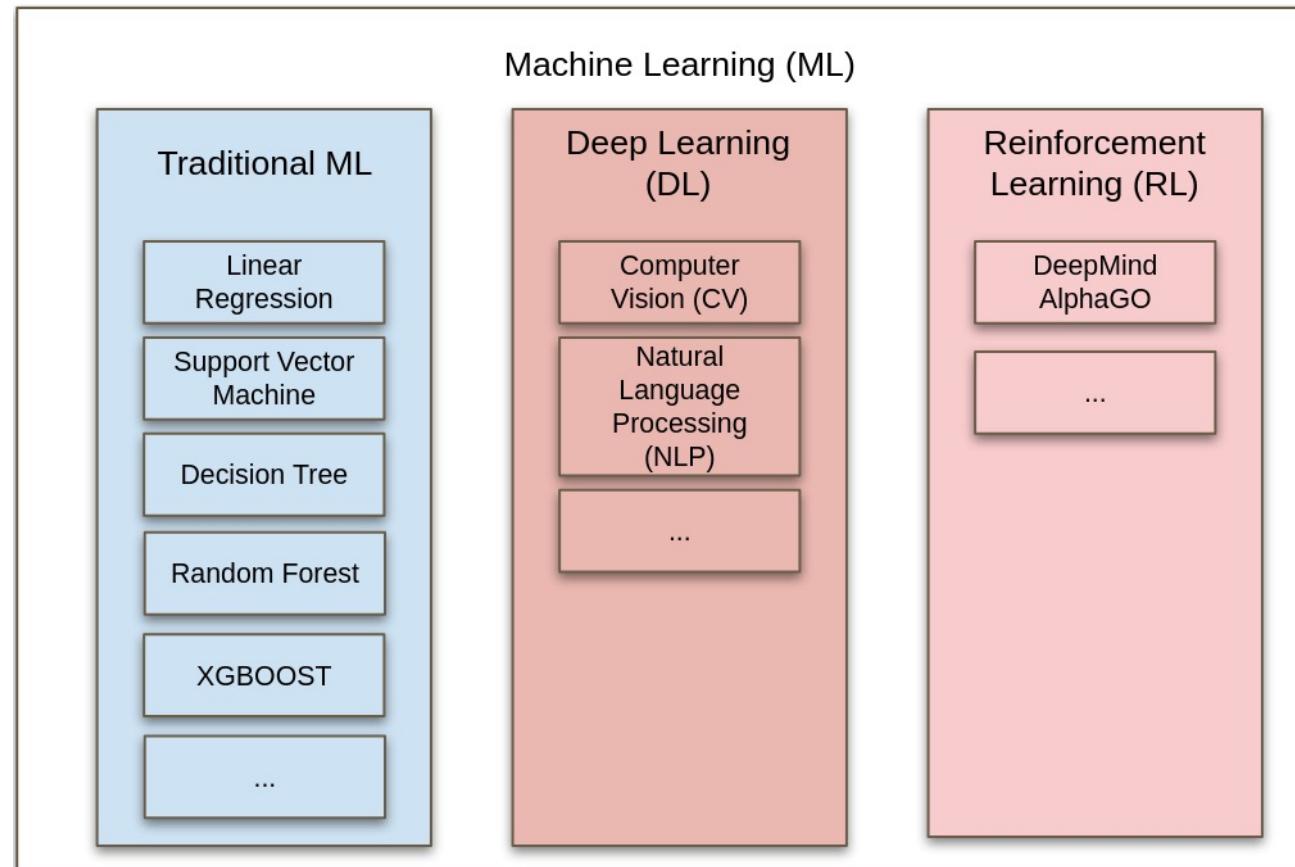
Septiembre, 2021

Tipos de Inteligencia Artificial



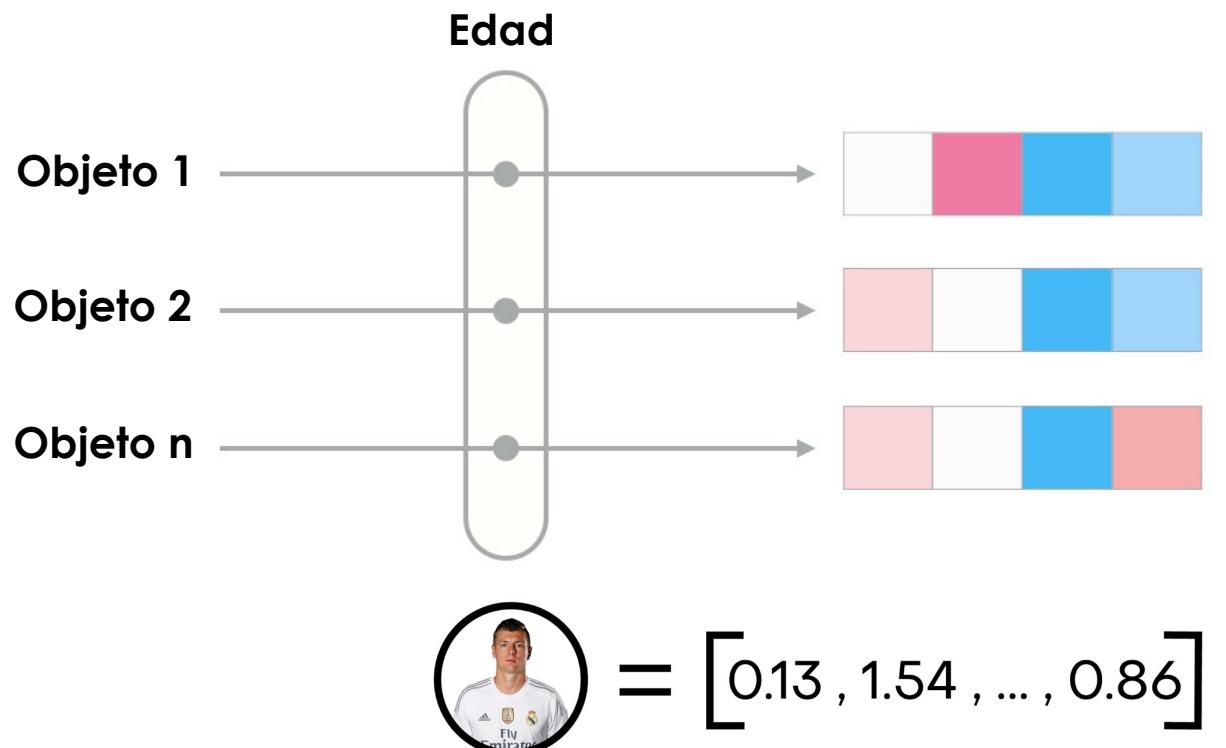
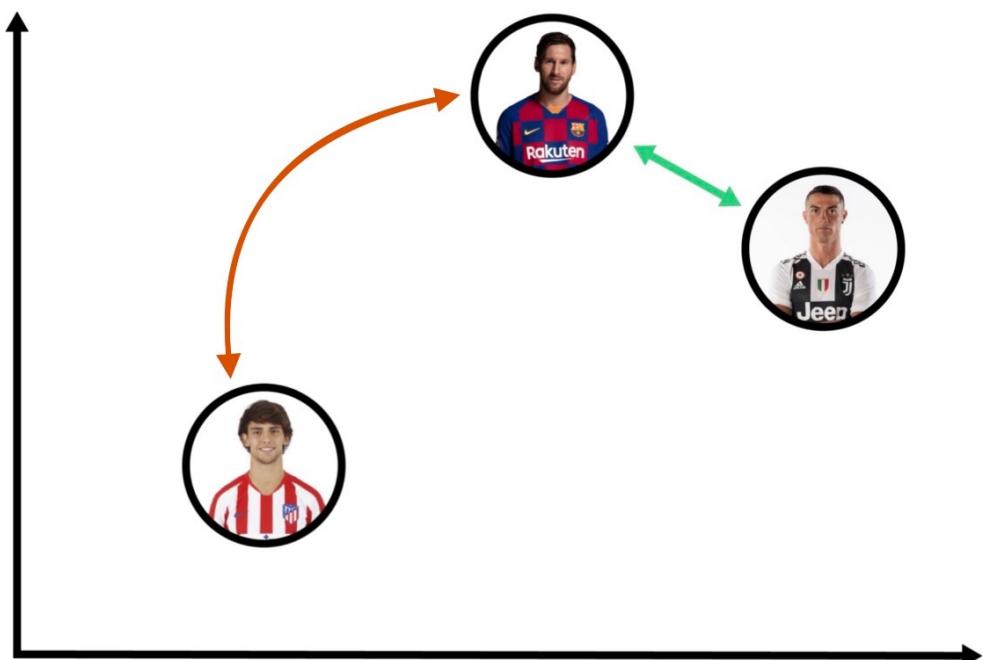
Tipos de Inteligencia Artificial

ML utiliza algoritmos basados en métodos matemáticos, que han existido desde décadas pasadas, pero no se les llamó ML o AI hasta hace algunos años.



Métricas de distancia

Muchos de estos algoritmos utilizan medidas de distancia, las cuales son importantes (más de lo que se imagina), para identificar objetos (elementos) con características similares y no similares (disímiles).



Métricas de distancia

Estas medidas de distancia, conocidas también como **búsqueda de similitud vectorial**, juegan un papel importante en el aprendizaje automático.

ID	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	Faltas/Año	Antigüedad	Sexo
E1	10000	Sí	No	0	Alquiler	No	7	15	M
E2	20000	No	Sí	1	Alquiler	Sí	3	3	F
E3	15000	Sí	Sí	2	Prop	Sí	5	10	M
E4	30000	Sí	Sí	1	Alquiler	No	15	7	F
E5	10000	Sí	Sí	0	Prop	Sí	1	6	M
E6	40000	No	Sí	0	Alquiler	Sí	3	16	F
E7	25000	No	No	0	Alquiler	Sí	0	8	M
E8	20000	No	Sí	0	Prop	Sí	2	6	F
E9	20000	Sí	Sí	3	Prop	No	7	5	M
E10	30000	Sí	Sí	2	Prop	No	1	20	M
E11	45000	No	No	0	Alquiler	No	2	12	F
E12	8000	Sí	Sí	2	Prop	No	3	1	M
E13	20000	No	No	0	Alquiler	No	27	5	F
E14	10000	No	Sí	0	Alquiler	Sí	0	7	M
E15	8000	No	Sí	0	Alquiler	No	3	2	M



¿Algunas ideas para medir las similitudes entre estos empleados con un determinado salario?

Se utiliza la información (características) de los vector de datos

Métricas de distancia

Medidas de distancia

- Una medida de distancia es una puntuación objetiva que resume la diferencia entre dos elementos (objetos), como: compras, ventas, diagnósticos, personas, usuarios, entre otros.
- Estas mediciones se utilizan para 'aprender de los datos'. Algunos algoritmos que utilizan medidas de distancia en su funcionamiento (núcleo) son:
 - K vecinos más cercanos (KNN).
 - Aprendizaje de cuantificación vectorial (LVQ).
 - Mapas autoorganizados (SOM).
 - Máquinas de soporte vectorial basados en kernel (SVM).
 - Clustering (K-means).
 - Clustering (Jerárquico).
 - El error (estimado y real) en problemas de regresión también son una distancia.

* Se utilizan diferentes medidas de distancia en función de los tipos de datos.

Métricas de distancia

Medidas de distancia

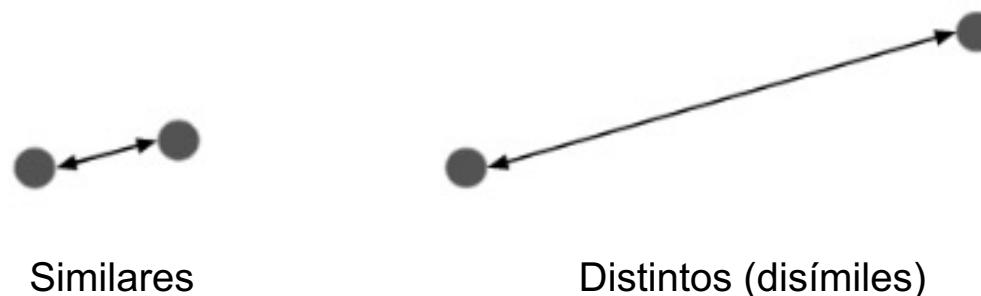
- Algunas medidas de distancia destacadas son:
 - Distancia Euclídea (Euclidiana).
 - Distancia de Chebyshev.
 - Distancia de Manhattan (Geometría del taxista).
 - Distancia de Minkowsky.
- Saber qué medida de distancia utilizar es útil para obtener modelos más precisos.
- Por ejemplo, si se quiere analizar búsquedas similares de un grupo de usuarios, éstas pueden ser imprecisas y variadas. Unos pueden buscar algo genérico como 'zapatos negros' o algo más preciso como 'Nike AF1 LV8'.
- ¿Qué pasa si los datos contienen **información geoespacial**?

Métricas de distancia

Medidas de distancia

Matemáticamente, una distancia es una función, $d(a, b)$, que asigna un valor positivo a cada par de objetos (puntos) de un espacio n-dimensional. Esta tiene las siguientes propiedades:

- **No negativa**, el valor puede ser mayor o igual a cero: $d(a, b) \geq 0$
- **Simétrica**, la distancia entre a y b es la misma que entre b y a : $d(a, b) = d(b, a)$
- La distancia de dos objetos en un mismo punto es cero: $d(a, a) = 0$



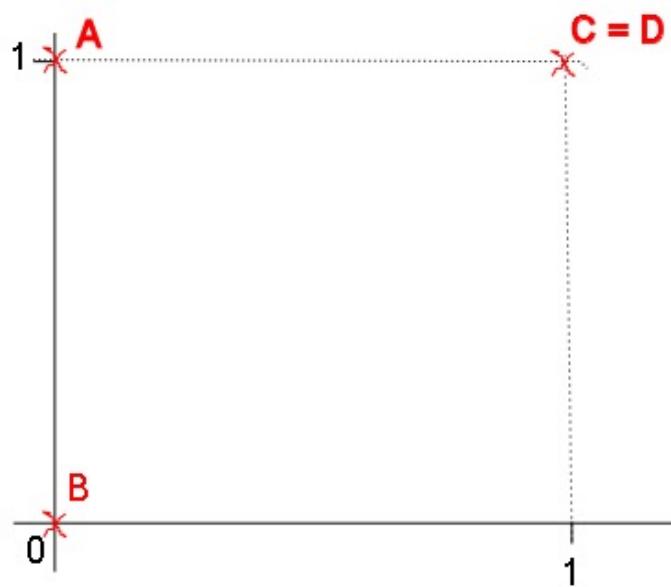
Métricas de distancia

Medidas de distancia

¿Qué pasa cuando dos elementos son iguales?

La distancia es cero: $d(a, a) = 0$

ID	X	Y
A	0	1
B	0	0
C	1	1
D	1	1



$$D_{(C,D)} = \sqrt{(1 - 1)^2 + (1 - 1)^2} = \sqrt{0} = 0$$

Métricas de distancia

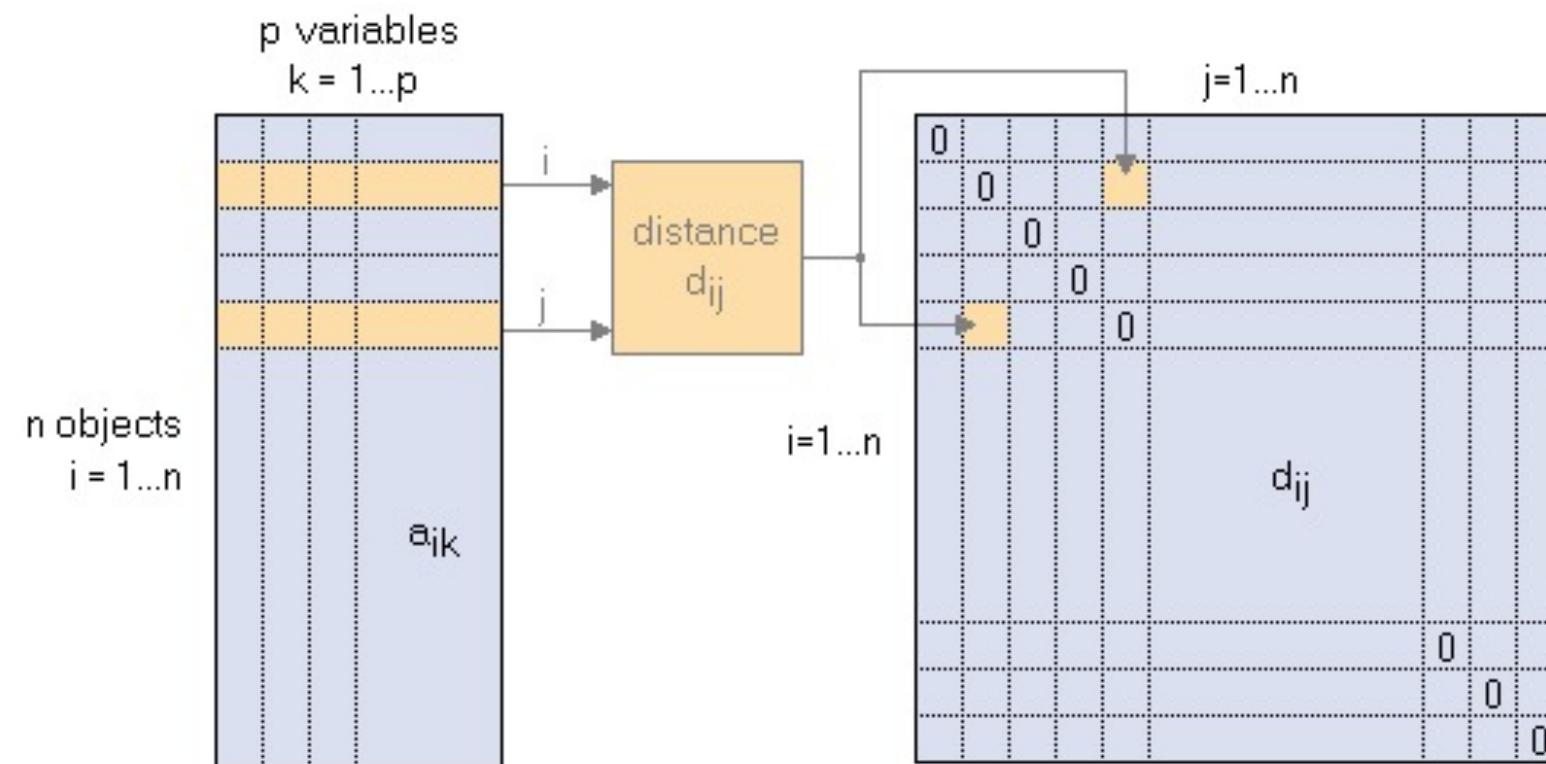
Aplicaciones de medidas de distancia

Son muchas las aplicaciones actuales que se basan en el cálculo de similitudes (distancias) entre vectores de datos (objetos). A eso se conoce como **aprendizaje basado en instancias**. Por ejemplo:

- Clasificaciones o reconocimiento de imágenes.
- Reconocimiento facial.
- Sistemas de recomendación.
- Recuperación de información.
- Agrupamiento de elementos.
- Marketing dirigido.
- Por mencionar algunas.

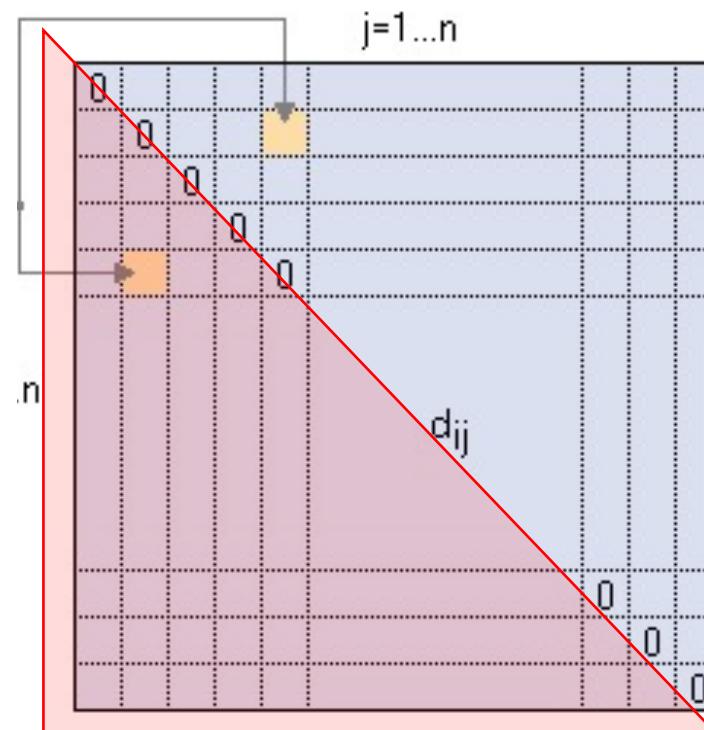
Métricas de distancia

Matriz de distancias



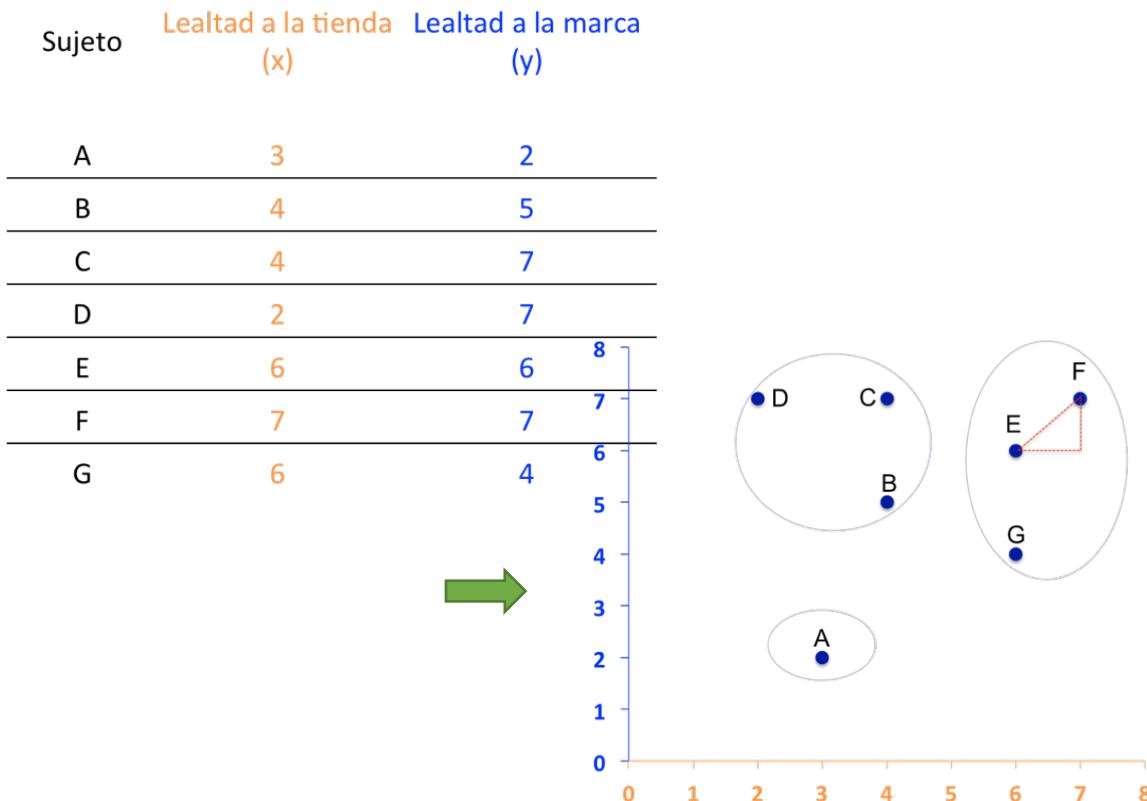
Métricas de distancia

Matriz de distancias

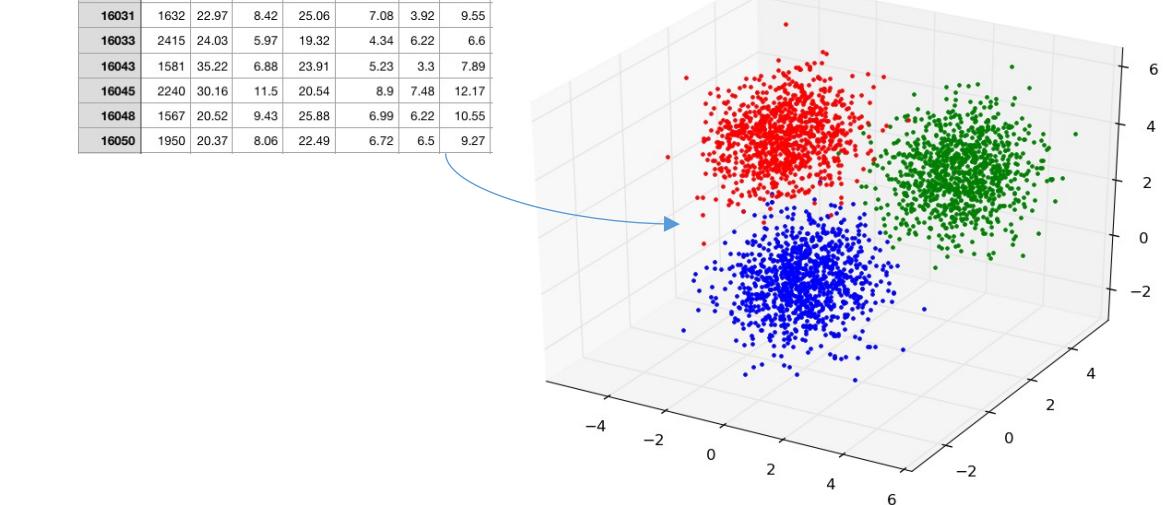


Métricas de distancia

Por ejemplo, mediante el **análisis de similitudes** se pueden formar grupos con elementos con características similares, de acuerdo a la semejanza de estos.



Estacion	Altitud	EneP	EneTO	EneTM	EneTMin	FebP	FebTO	FebTM	FebTMin	MarP	MarTO	MarTM	MarTMin
16006	360	28.24	19.64	33.37	14.03	1.26	20.85	34.66	15.27	1.3	23.22	36.31	18.22
16007	682	21.48	17.37	33.84	15.49	2.6	18.59	35.44	16.7	2.08	20.44	37.17	18
16014	1708	18.94	7.08	25.02	4.26	5.88	8.24	27.01	5.27	4.34	10.04	29.43	6.95
16016	1840	19.47	6.32	21.31	3.76	6.36	7.92	23.29	5.16	7.07	10.02	25.27	6.93
16017	1694	18.04	7.04	24.59	4.78	6.55	8.39	26.61	5.99	6.82	10.79	29.29	7.84
16020	2020	23.9	5.24	23.45	3.07	9.13	5.97	25.07	3.91	7.29	7.18	26.98	5.01
16023	1500	13.76	5.24	22.09	1.44	5.18	6.41	23.46	2.58	5.43	8.58	25.52	4.36
16024	1693	14.54	7.17	23.39	5.62	2.86	8.72	25.47	7.22	3.19	11.22	27.93	9.64
16027	1831	22.26	9.81	23.3	5.86	4.73	10.9						
16031	1632	22.97	8.42	25.06	7.08	3.92	9.55						
16033	2415	24.03	5.97	19.32	4.34	6.22	6.6						
16043	1581	35.22	6.88	23.91	5.23	3.3	7.89						
16045	2240	30.16	11.5	20.54	8.9	7.48	12.17						
16048	1567	20.52	9.43	25.88	6.99	6.22	10.55						
16050	1950	20.37	8.06	22.49	6.72	6.5	9.27						

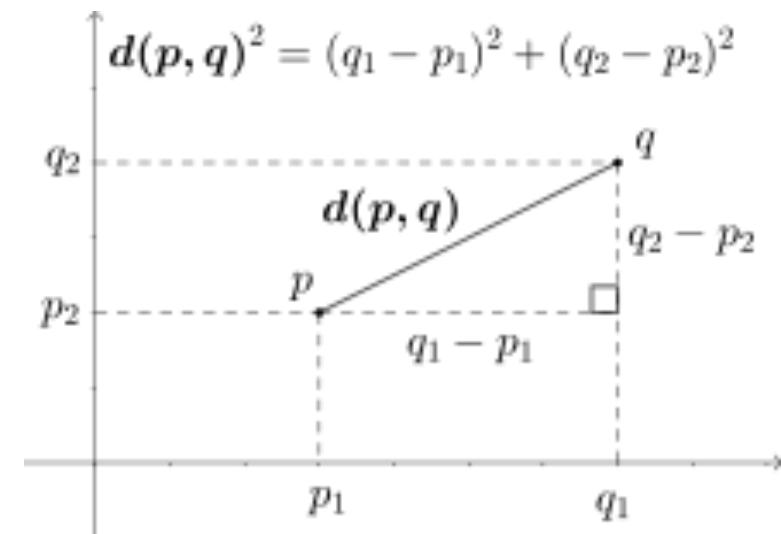
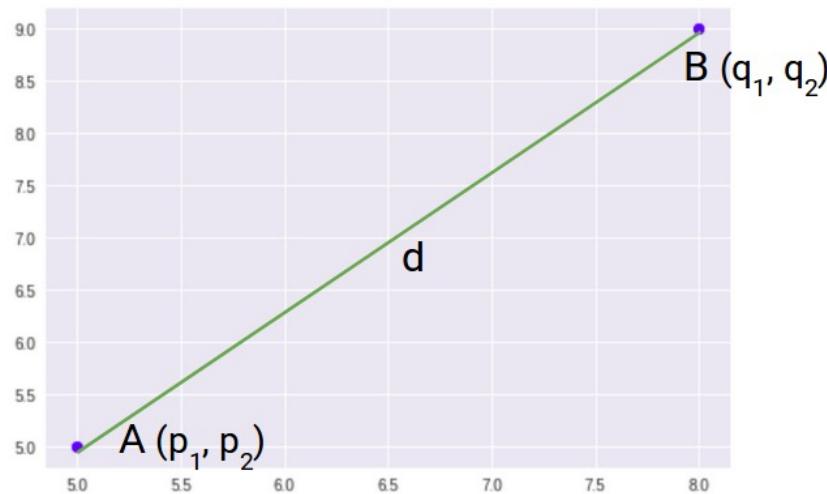


La complejidad está cuando se tiene un amplio número de variables.

1. Distancia Euclíadiana

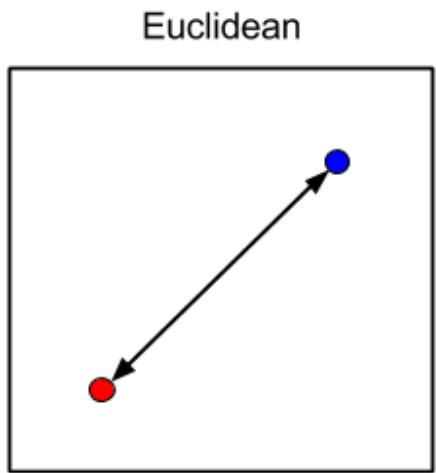
1. Distancia Euclídea

- **Distancia Euclídea** (euclídea, por Euclides) es una de las métricas más utilizadas para calcular la distancia entre dos puntos, conocida también como **espacio euclídeo**.
- Sus bases se encuentran en la aplicación del Teorema de Pitágoras, donde la distancia viene a ser la longitud de la hipotenusa.



1. Distancia Euclídea

Dimensiones:



$$dist(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

$$dist(p, q) = \sqrt{(p_1 - q_1)^2}$$

1 dimensión

$$dist(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$$

2 dimensiones

$$dist(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2}$$

3 dimensiones

$$dist(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2 + \dots + (p_n - q_n)^2}$$

n dimensiones

1. Distancia Euclídea

Ejemplo:

Sujeto	Lealtad a la tienda (x)	Lealtad a la marca (y)
A	3	2
B	4	5
C	4	7
D	2	7
E	6	6
F	7	7
G	6	4

$$dist(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

$$dist(p, q) = d_{ij} = D_{(A,B)} = \sqrt{(3 - 4)^2 + (2 - 5)^2} = \sqrt{(-1)^2 + (-3)^2} = \sqrt{10} = 3.16$$

$$dist(p, q) = d_{ij} = D_{(E,F)} = \sqrt{(6 - 7)^2 + (6 - 7)^2} = \sqrt{(-1)^2 + (-1)^2} = \sqrt{2} = 1.41$$

Sujetos	A	B	C	D	E	F	G
A	---						
B	3.16	---					
C	5.10	2.00	---				
D	5.10	2.83	2.00	---			
E	5.00	2.24	2.24	4.12	---		
F	6.40	3.61	3.00	5.00	1.41	---	
G	3.61	2.24	3.61	5.00	2.00	3.16	---

1. Distancia Euclídea

Retomando dos vectores de datos de empleados:

ID	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	Faltas/Año	Antigüedad	Sexo
E1	10000	1	0	0	0	0	7	15	1
E2	20000	0	1	1	0	1	3	3	0

$$dist(p, q) = dij = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

$$dist_{(E1, E2)} = \sqrt{(10000 - 20000)^2 + (1 - 0)^2 + (0 - 1)^2 + (0 - 1)^2 + (0 - 0)^2 + (0 - 1)^2 + (7 - 3)^2 + (15 - 3)^2 + (1 - 0)^2}$$

$$dist_{(E1, E2)} = \sqrt{(-10000)^2 + (1)^2 + (-1)^2 + (-1)^2 + (0)^2 + (-1)^2 + (4)^2 + (12)^2 + (1)^2} = 10000.008$$

1. Distancia Euclídea

Retomando dos vectores de datos de empleados:

ID	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	Faltas/Año	Antigüedad	Sexo
E1	10000	1	0	0	0	0	7	15	1
E2	20000	0	1	1	0	1	3	3	0

$$dist(p, q) = dij = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$



```
from scipy.spatial import distance
E1 = (10000,1,0,0,0,0,7,15,1)
E2 = (20000,0,1,1,0,1,3,3,0)
dstEuclidiana = distance.euclidean(E1,E2)
dstEuclidiana
```

10000.008249996597

1. Distancia Euclídea

Retomando dos vectores de datos de empleados:

ID	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	Faltas/Año	Antigüedad	Sexo
E1	10000	1	0	0	0	0	7	15	1
E2	20000	0	1	1	0	1	3	3	0

```
▶ from math import sqrt  
E1 = (10000,1,0,0,0,0,7,15,1) #datos del punto 1  
E2 = (20000,0,1,1,0,1,3,3,0) #datos del punto 2  
#La función zip() es un iterador de tuplas  
dst1 = sqrt(sum((E1-E2)**2 for E1, E2 in zip(E1, E2)))  
dst1
```

10000.008249996597

```
▶ import numpy as np  
import matplotlib as plt  
E1 = np.array([10000,1,0,0,0,0,7,15,1])  
E2 = np.array([20000,0,1,1,0,1,3,3,0])  
dst2 = np.sqrt(np.sum((E1-E2)**2))  
dst2
```

10000.008249996597

2. Distancia de Chebyshev

2. Distancia de Chebyshev

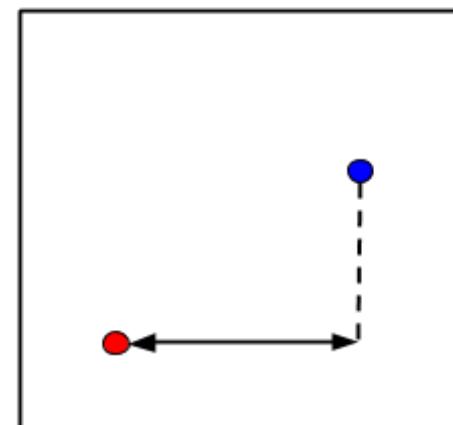
- La **distancia de Chebyshev** es el valor máximo absoluto de las diferencias entre las coordenadas de un par de elementos.
- Lleva el nombre del matemático ruso Pafnuty Chebyshev, conocido por su trabajo en la geometría analítica y teoría de números.
- Otro nombre para la distancia de Chebyshev es **métrica máxima**.

ID	F ₁	F ₂	F ₃
A	2	3	4
B	5	9	11

$$d_{Cheb}(p, q) = \max |p_i - q_i|$$

$$d_{Cheb}(A, B) = \max\{|2 - 5|, |3 - 9|, |4 - 11|\} = \max\{3, 6, 7\} = 7$$

Chebychev



Se utiliza en la programación de movimientos de robots industriales.

2. Distancia de Chebyshev

Retomando dos vectores de datos de empleados:

ID	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	Faltas/Año	Antigüedad	Sexo
E1	10000	1	0	0	0	0	7	15	1
E2	20000	0	1	1	0	1	3	3	0

$$d_{Cheb}(p, q) = \max |p_i - q_i|$$

$$d_{Cheb(E_1, E_2)} = \max\{|10000 - 20000|, |1 - 0|, |0 - 1|, |0 - 1|, |0 - 0|, |0 - 1|, |7 - 3|, |15 - 3|, |1 - 0|\}$$

$$d_{Cheb(E_1, E_2)} = \max\{|-10000|, |1|, |-1|, |-1|, |0|, |-1|, |4|, |12|, |1|\}$$

$$d_{Cheb(E_1, E_2)} = 10000$$

2. Distancia de Chebyshev

Retomando dos vectores de datos de empleados:

ID	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	Faltas/Año	Antigüedad	Sexo
E1	10000	1	0	0	0	0	7	15	1
E2	20000	0	1	1	0	1	3	3	0

$$d_{Cheb}(p, q) = \max |p_i - q_i|$$



```
from scipy.spatial import distance
E1 = (10000,1,0,0,0,0,7,15,1)
E2 = (20000,0,1,1,0,1,3,3,0)
dstChebyshev = distance.chebyshev(E1,E2)
dstChebyshev
```

10000

3. Distancia de Manhattan

3. Distancia de Manhattan

- La **distancia euclíadiana** es una buena métrica. Sin embargo, en la vida real, por ejemplo en una ciudad, es imposible moverse de un punto a otro de manera recta.
- Se utiliza la **distancia de Manhattan** si se necesita calcular la distancia entre dos puntos en una ruta similar a una cuadrícula (información geoespacial).
- Se llama Manhattan debido al diseño de cuadrícula de la mayoría de las calles de la isla de Manhattan, Nueva York (USA).



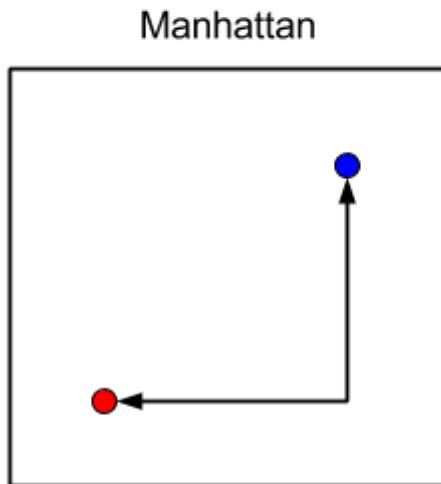
3. Distancia de Manhattan

- La **distancia de Manhattan** también se conoce como geometría del taxi, distancia de la manzana de la ciudad, y distancia rectilínea.

ID	F ₁	F ₂	F ₃
A	2	3	4
B	5	9	11

$$d_{Manh}(p, q) = \sum_{i=1}^n |p_i - q_i|$$

$$d_{Manh}(A, B) = |2 - 5| + |3 - 9| + |4 - 11| = 3 + 6 + 7 = 16$$



3. Distancia de Manhattan

Retomando dos vectores de datos de empleados:

ID	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	Faltas/Año	Antigüedad	Sexo
E1	10000	1	0	0	0	0	7	15	1
E2	20000	0	1	1	0	1	3	3	0

$$d_{Manh}(p, q) = \sum_{i=1}^n |p_i - q_i|$$

$$d_{Manh(E1, E2)} = |10000 - 20000| + |1 - 0| + |0 - 1| + |0 - 1| + |0 - 0| + |0 - 1| + |7 - 3| + |15 - 3| + |1 - 0|$$

$$d_{Manh(E1, E2)} = |-10000| + |1| + |-1| + |-1| + |0| + |-1| + |4| + |12| + |1|$$

$$d_{Manh(E1, E2)} = 10021$$

3. Distancia de Manhattan

Retomando dos vectores de datos de empleados:

ID	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	Faltas/Año	Antigüedad	Sexo
E1	10000	1	0	0	0	0	7	15	1
E2	20000	0	1	1	0	1	3	3	0

$$d_{Manh}(p, q) = \sum_{i=1}^n |p_i - q_i|$$



```
from scipy.spatial import distance
E1 = (10000,1,0,0,0,0,7,15,1)
E2 = (20000,0,1,1,0,1,3,3,0)
dstManhattan = distance.cityblock(E1,E2)
dstManhattan
```

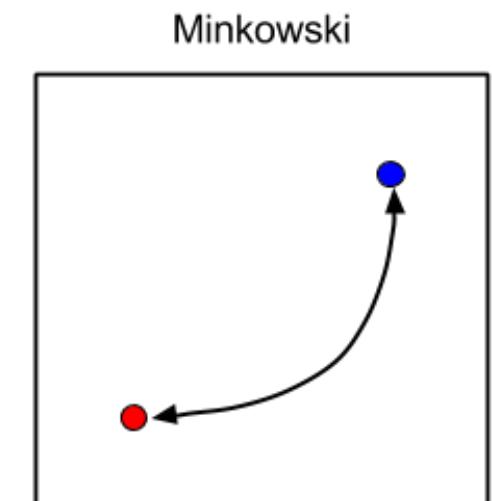
10021

4. Distancia de Minkowski

4. Distancia de Minkowski

- La **distancia de Minkowski** es una distancia entre dos puntos en un espacio n-dimensional. Es una métrica de distancia generalizada: Euclídea, Manhattan y Chebyshev.

$$d_{Mink}(q, p) = \lambda \sqrt[n]{\sum_{i=1}^n (q_i - p_i)^\lambda} = \left(\sum_{i=1}^n (q_i - p_i)^\lambda \right)^{1/\lambda}$$



donde λ es el **orden** para calcular la distancia de tres formas diferentes:

- $\lambda = 1$, distancia de Manhattan (métrica L^1)
- $\lambda = 2$, distancia Euclídea (métrica L^2)
- $\lambda = 3$, distancia de Chebyshev (métrica L)
- En la actualidad se usan valores intermedios, por ejemplo, $\lambda = 1.5$, que proporciona un equilibrio entre las medidas.

4. Distancia de Minkowski

Dado dos vectores de datos:

ID	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	Faltas/Año	Antigüedad	Sexo
E1	10000	1	0	0	0	0	7	15	1
E2	20000	0	1	1	0	1	3	3	0

$$d_{Mink}(q, p) = \sqrt{\lambda} \sum_{i=1}^n (q_i - p_i)^{\lambda}$$



```
from scipy.spatial import distance
E1 = (10000,1,0,0,0,0,7,15,1)
E2 = (20000,0,1,1,0,1,3,3,0)
dstMinkowski = distance.minkowski(E1,E2, 1.5)
dstMinkowski
```

10000.363791487287

Métricas de distancia

```
▶ from scipy.spatial import distance  
E1 = (10000,1,0,0,0,0,7,15,1)  
E2 = (20000,0,1,1,0,1,3,3,0)  
dstEuclidiana = distance.euclidean(E1,E2)  
dstEuclidiana
```

10000.0082499

```
▶ from scipy.spatial import distance  
E1 = (10000,1,0,0,0,0,7,15,1)  
E2 = (20000,0,1,1,0,1,3,3,0)  
dstChebyshev = distance.chebyshev(E1,E2)  
dstChebyshev
```

10000

```
▶ from scipy.spatial import distance  
E1 = (10000,1,0,0,0,0,7,15,1)  
E2 = (20000,0,1,1,0,1,3,3,0)  
dstManhattan = distance.cityblock(E1,E2)  
dstManhattan
```

10021

```
▶ from scipy.spatial import distance  
E1 = (10000,1,0,0,0,0,7,15,1)  
E2 = (20000,0,1,1,0,1,3,3,0)  
dstMinkowski = distance.minkowski(E1,E2, 1.5)  
dstMinkowski
```

10000.363791487287

Métricas de distancia

Euclidianas

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	10000.01													
3	5000.00	5000.01												
4	20000.00	10000.01	15000.00											
5	10.95	10000.00	5000.00	20000.01										
6	30000.00	20000.00	25000.00	10000.01	30000.00									
7	15000.00	5000.00	10000.00	5000.02	15000.00	15000.00								
8	10000.01	3.46	5000.00	10000.01	10000.00	20000.00	5000.00							
9	10000.01	5.29	5000.00	10000.00	10000.00	20000.00	5000.01	6.16						
10	20000.00	10000.01	15000.00	19.18	20000.01	10000.00	5000.02	10000.01	10000.01					
11	35000.00	25000.00	30000.00	15000.01	35000.00	5000.00	20000.00	25000.00	25000.00	15000.00				
12	2000.05	12000.00	7000.01	22000.00	2000.01	32000.00	17000.00	12000.00	12000.00	22000.01	37000.00			
13	10000.03	24.15	5000.05	10000.01	10000.03	20000.02	5000.07	25.08	20.32	10000.05	25000.01	12000.02		
14	10.77	10000.00	5000.00	20000.01	2.00	30000.00	15000.00	10000.00	10000.00	20000.00	35000.00	2000.01	10000.04	
15	2000.05	12000.00	7000.01	22000.00	2000.01	32000.00	17000.00	12000.00	12000.00	22000.01	37000.00	2.65	12000.02	2000.01

Chebyshev

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	10000													
3	5000	5000												
4	20000	10000	15000											
5	9	10000	5000	20000										
6	30000	20000	25000	10000	30000									
7	15000	5000	10000	5000	15000	15000								
8	10000	3	5000	10000	10000	20000	5000							
9	10000	4	5000	10000	10000	20000	5000	5						
10	20000	10000	15000	14	20000	10000	5000	10000	10000					
11	35000	25000	30000	15000	35000	5000	20000	25000	25000	15000				
12	2000	12000	7000	22000	2000	32000	17000	12000	12000	22000	37000			
13	10000	24	5000	10000	10000	20000	5000	25	20	10000	25000	12000		
14	8	10000	5000	20000	1	30000	15000	10000	10000	20000	35000	2000	10000	
15	2000	12000	7000	22000	2000	32000	17000	12000	12000	22000	37000	2	12000	2000

Métricas de distancia

Manhattan

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	10021													
3	5012	5013												
4	20019	10018	15017											
5	18	10009	5010	20019										
6	30009	20014	25013	10024	30015									
7	15016	5011	10012	5021	15006	15013								
8	10019	6	5011	10018	10003	20012	5007							
9	10015	12	5009	10014	10011	20022	5017	12						
10	20015	10024	15015	30	20017	10012	5019	10020	10022					
11	35010	25013	30012	15021	35012	5007	20008	25009	25019	15015				
12	2022	12007	7012	22021	2010	32021	17016	12011	12009	22021	37018			
13	10032	29	5034	10017	10032	20037	5032	29	27	10047	25032	12034		
14	18	10009	5012	20019	4	30013	15002	10005	10015	20019	35010	2014	10032	
15	2019	12004	7015	22020	2009	32016	17011	12008	12012	22024	37013	5	12029	2009

Minkowski

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	10000.01													
3	5000.00	5000.01												
4	20000.00	10000.01	15000.00											
5	10.95	10000.00	5000.00	20000.01										
6	30000.00	20000.00	25000.00	10000.01	30000.00									
7	15000.00	5000.00	10000.00	5000.02	15000.00	15000.00								
8	10000.01	3.46	5000.00	10000.01	10000.00	20000.00	5000.00							
9	10000.01	5.29	5000.00	10000.00	10000.00	20000.00	5000.01	6.16						
10	20000.00	10000.01	15000.00	19.18	20000.01	10000.00	5000.02	10000.01	10000.01					
11	35000.00	25000.00	30000.00	15000.01	35000.00	5000.00	20000.00	25000.00	25000.00	15000.00				
12	2000.05	12000.00	7000.01	22000.00	2000.01	32000.00	17000.00	12000.00	12000.00	22000.01	37000.00			
13	10000.03	24.15	5000.05	10000.01	10000.03	20000.02	5000.07	25.08	20.32	10000.05	25000.01	12000.02		
14	10.77	10000.00	5000.00	20000.01	2.00	30000.00	15000.00	10000.00	10000.00	20000.00	35000.00	2000.01	10000.04	
15	2000.05	12000.00	7000.01	22000.00	2000.01	32000.00	17000.00	12000.00	12000.00	22000.01	37000.00	2.65	12000.02	2000.01

Consideraciones finales

- Una buena métrica de distancia ayuda a mejorar significativamente el rendimiento del proceso de clasificación, clusterización, recuperación de información y otras aplicaciones.
- La función utilizada para medir distancias depende de la situación en la que esté trabajando. Por ejemplo, en algunas áreas, la distancia euclídea puede ser óptima y útil para calcular distancias.
- Hay aplicaciones que requieren de un enfoque refinado para calcular distancias entre puntos, como la distancia Minkowski.



Universidad Nacional Autónoma de México
Facultad de Ingeniería

Métricas de distancia

Práctica 3

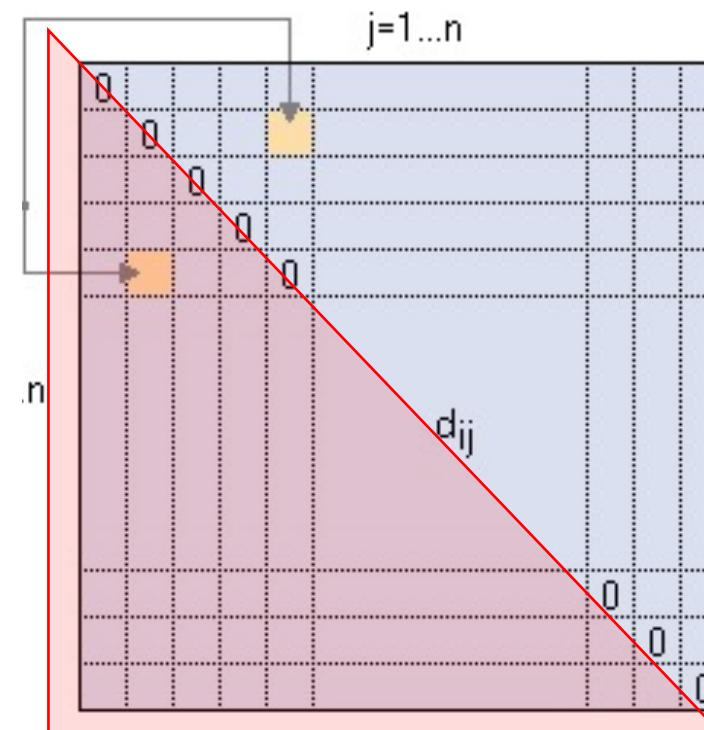
Guillermo Molero-Castillo

guillermo.molero@ingenieria.unam.edu

Septiembre, 2021

Práctica

Objetivo. Obtener las matrices de distancia (Euclídea, Chebyshev, Manhattan, Minkowski) en Google Colab a partir de una matriz de datos.



Práctica

Fuente de datos

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo	comprar
0	6000	1000	0	600	50000	400000		0	2	2
1	6745	944	123	429	43240	636897		1	3	6
2	6455	1033	98	795	57463	321779		2	1	8
3	7098	1278	15	254	54506	660933		0	0	3
4	6167	863	223	520	41512	348932		0	0	3
...
197	3831	690	352	488	10723	363120		0	0	2
198	3961	1030	270	475	21880	280421		2	3	8
199	3184	955	276	684	35565	388025		1	3	8
200	3334	867	369	652	19985	376892		1	2	5
201	3988	1157	105	382	11980	257580		0	0	4

202 rows × 10 columns

Fuente de datos

- ingresos: son ingresos mensuales de 1 o 2 personas, si están casados.
- gastos_comunes: son gastos mensuales de 1 o 2 personas, si están casados.
- pago_coche
- gastos_otros
- ahorros
- vivienda: valor de la vivienda.
- estado_civil: 0-soltero, 1-casado, 2-divorciado
- hijos: cantidad de hijos menores (no trabajan).
- trabajo: 0-sin trabajo, 1-autonomo, 2-asalariado, 3-empresario, 4-autonomos, 5-asalariados, 6-autonomo y asalariado, 7-empresario y autonomo, 8-empresarios o empresario y autónomo
- comprar: 0-alquilar, 1-comprar casa a través de crédito hipotecario con tasa fija a 30 años.

1. Importar las bibliotecas necesarias

```
▶ import pandas as pd # Para la manipulación y análisis de datos  
import numpy as np # Para crear vectores y matrices n dimensionales  
import matplotlib.pyplot as plt # Para generar gráficas a partir de los datos  
from scipy.spatial.distance import cdist # Para el cálculo de distancias  
  
▶ from google.colab import files  
files.upload()
```

2. Importar los datos



```
Hipoteca = pd.read_csv("Hipoteca.csv")  
Hipoteca
```

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	edad	deudas
0	6000	1000	0	600	50000	400000	Soltero/a	30	10000
1	6745	944	123	429	43240	636897	Casado/a	35	15000
2	6455	1033	98	795	57463	321779	Soltero/a	32	12000
3	7098	1278	15	254	54506	660933	Divorciado/a	38	18000
4	6167	863	223	520	41512	348932	Soltero/a	34	14000
...
197	3831	690	352	488	10723	363120	Soltero/a	30	10000
198	3961	1030	270	475	21880	280421	Soltero/a	35	15000
199	3184	955	276	684	35565	388025	Soltero/a	32	12000
200	3334	867	369	652	19985	376892	Soltero/a	34	14000
201	3988	1157	105	382	11980	257580	Soltero/a	36	16000

202 rows × 10 columns

3. Matrices de distancias

a) Euclíadiana

```
▶ DstEuclidian = cdist(Hipoteca, Hipoteca, metric='euclidean')
MEuclidian = pd.DataFrame(DstEuclidian)
```

```
▶ print(MEuclidian)
#MEuclidian
```

	0	1	...	200	201
0	0.000000	236994.701964	...	37975.571227	147421.532182
1	236994.701964	0.000000	...	261065.405879	380612.957023
2	78577.840350	315439.176808	...	66722.600009	78717.767975
3	260974.591407	26550.527773	...	286156.617026	405600.560294
4	51769.581416	287970.807817	...	35401.101452	96032.256950
..
197	53923.596347	275716.907131	...	16605.967753	105548.977428
198	122858.123985	357126.266127	...	96491.998140	24895.261437
199	18967.999420	249015.957900	...	19149.935143	132563.033841
200	37975.571227	261065.405879	...	0.000000	119582.974486
201	147421.532182	380612.957023	...	119582.974486	0.000000

[202 rows x 202 columns]

3. Matrices de distancias

a) Euclíadiana

```
▶ print(MEuclidiana.round(3))
```

	0	1	2	...	199	200	201
0	0.000	236994.702	78577.840	...	18967.999	37975.571	147421.532
1	236994.702	0.000	315439.177	...	249015.958	261065.406	380612.957
2	78577.840	315439.177	0.000	...	69848.439	66722.600	78717.768
3	260974.591	26550.528	339168.030	...	273593.155	286156.617	405600.560
4	51769.581	287970.808	31494.808	...	39655.592	35401.101	96032.257
..
197	53923.596	275716.907	62456.927	...	35184.046	16605.968	105548.977
198	122858.124	357126.266	54616.720	...	108473.744	96491.998	24895.261
199	18967.999	249015.958	69848.439	...	0.000	19149.935	132563.034
200	37975.571	261065.406	66722.600	...	19149.935	0.000	119582.974
201	147421.532	380612.957	78717.768	...	132563.034	119582.974	0.000

[202 rows x 202 columns]

3. Matrices de distancias

a) Euclíadiana

```
DstEucliana = cdist(Hipoteca.iloc[0:10], Hipoteca.iloc[0:10], metric='euclidean')
MEucliana = pd.DataFrame(DstEucliana)
print(MEucliana)
```

	0	1	...	8	9
0	0.000000	236994.701964	...	108991.940697	76488.543044
1	236994.701964	0.000000	...	345963.774390	312810.379793
2	78577.840350	315439.176808	...	31548.758977	17030.194685
3	260974.591407	26550.527773	...	369945.815299	337121.576353
4	51769.581416	287970.807817	...	58617.026426	24868.539744
5	39149.060512	276141.622437	...	69857.763606	38195.246432
6	30003.797860	207115.404780	...	138853.960905	105892.923725
7	206425.706195	33742.472390	...	315357.550518	282695.457394
8	108991.940697	345963.774390	...	0.000000	34544.425223
9	76488.543044	312810.379793	...	34544.425223	0.000000

[10 rows x 10 columns]

3. Matrices de distancias

a) Euclíadiana (entre dos objetos)

```
▶ Objeto1 = Hipoteca.iloc[0]
  Objeto2 = Hipoteca.iloc[1]
  dstEuclidiana = distance.euclidean(Objeto1,Objeto2)
  dstEuclidiana
```

236994.70196398906

3. Matrices de distancias

b) Chebyshev

```
▶ DstChebyshev = cdist(Hipoteca, Hipoteca, metric='chebyshev')
MChebyshev = pd.DataFrame(DstChebyshev)
```

```
▶ print(MChebyshev)
```

	0	1	2	...	199	200	201
0	0.0	236897.0	78221.0	...	14435.0	30015.0	142420.0
1	236897.0	0.0	315118.0	...	248872.0	260005.0	379317.0
2	78221.0	315118.0	0.0	...	66246.0	55113.0	64199.0
3	260933.0	24036.0	339154.0	...	272908.0	284041.0	403353.0
4	51068.0	287965.0	27153.0	...	39093.0	27960.0	91352.0
..
197	39277.0	273777.0	46740.0	...	24905.0	13772.0	105540.0
198	119579.0	356476.0	41358.0	...	107604.0	96471.0	22841.0
199	14435.0	248872.0	66246.0	...	0.0	15580.0	130445.0
200	30015.0	260005.0	55113.0	...	15580.0	0.0	119312.0
201	142420.0	379317.0	64199.0	...	130445.0	119312.0	0.0

[202 rows x 202 columns]

3. Matrices de distancias

b) Chebyshev

```
▶ DstChebyshev = cdist(Hipoteca.iloc[0:10], Hipoteca.iloc[0:10], metric='chebyshev')
MChebyshev = pd.DataFrame(DstChebyshev)
print(MChebyshev)
```

```
0          0         1         2       ...        7         8         9
0      0.0  236897.0  78221.0  ...  206291.0  108990.0  75902.0
1  236897.0      0.0  315118.0  ...  30606.0  345887.0  312799.0
2   78221.0  315118.0      0.0  ...  284512.0  30769.0  16852.0
3  260933.0  24036.0  339154.0  ...  54642.0  369923.0  336835.0
4   51068.0  287965.0  27153.0  ...  257359.0  57922.0  24834.0
5   39137.0  276034.0  39084.0  ...  245428.0  69853.0  36765.0
6   29812.0  207085.0  108033.0  ...  176479.0  138802.0  105714.0
7  206291.0  30606.0  284512.0  ...      0.0  315281.0  282193.0
8  108990.0  345887.0  30769.0  ...  315281.0      0.0  33088.0
9   75902.0  312799.0  16852.0  ...  282193.0  33088.0      0.0
```

[10 rows x 10 columns]

3. Matrices de distancias

b) Chebyshev (entre dos objetos)



```
Objeto1 = Hipoteca.iloc[0]
Objeto2 = Hipoteca.iloc[1]
dstChebyshev = distance.chebyshev(Objeto1,Objeto2)
dstChebyshev
```

236897

3. Matrices de distancias

c) Manhattan

```
▶ DstManhattan = cdist(Hipoteca, Hipoteca, metric='cityblock')
MManhattan = pd.DataFrame(DstManhattan)
```

```
▶ print(MManhattan)
```

	0	1	2	...	199	200	201
0	0.0	244759.0	86474.0	...	29640.0	56348.0	182937.0
1	244759.0	0.0	330117.0	...	260529.0	287219.0	413618.0
2	86474.0	330117.0	0.0	...	91786.0	96298.0	112701.0
3	267180.0	36279.0	343632.0	...	296786.0	323494.0	449329.0
4	60166.0	290551.0	43970.0	...	48342.0	52608.0	123615.0
..
197	79103.0	309758.0	91619.0	...	50941.0	23895.0	107776.0
198	150173.0	380902.0	79933.0	...	122357.0	99437.0	33162.0
199	29640.0	260529.0	91786.0	...	0.0	27080.0	155517.0
200	56348.0	287219.0	96298.0	...	27080.0	0.0	128799.0
201	182937.0	413618.0	112701.0	...	155517.0	128799.0	0.0

[202 rows x 202 columns]

3. Matrices de distancias

c) Manhattan

```
▶ DstManhattan = cdist(Hipoteca.iloc[0:10], Hipoteca.iloc[0:10], metric='cityblock')
MManhattan = pd.DataFrame(DstManhattan)
print(MManhattan)
```

	0	1	2	...	7	8	9
0	0.0	244759.0	86474.0	...	214460.0	110235.0	87151.0
1	244759.0	0.0	330117.0	...	45617.0	354186.0	316302.0
2	86474.0	330117.0	0.0	...	284636.0	38493.0	20633.0
3	267180.0	36279.0	343632.0	...	59000.0	375313.0	351115.0
4	60166.0	290551.0	43970.0	...	274210.0	67449.0	27261.0
5	40701.0	284974.0	47121.0	...	253389.0	71574.0	48998.0
6	34820.0	211391.0	120112.0	...	188566.0	143573.0	112221.0
7	214460.0	45617.0	284636.0	...	0.0	323035.0	300521.0
8	110235.0	354186.0	38493.0	...	323035.0	0.0	44100.0
9	87151.0	316302.0	20633.0	...	300521.0	44100.0	0.0

[10 rows x 10 columns]

3. Matrices de distancias

c) Manhattan (entre dos puntos)



```
Objeto1 = Hipoteca.iloc[0]
Objeto2 = Hipoteca.iloc[1]
dstManhattan = distance.cityblock(Objeto1,Objeto2)
dstManhattan
```

244759

3. Matrices de distancias

d) Minkowski

```
▶ DstMinkowski = cdist(Hipoteca, Hipoteca, metric='minkowski', p=1.5)
MMinkowski = pd.DataFrame(DstMinkowski)
```

```
▶ print(MMinkowski)
```

	0	1	...	200	201
0	0.000000	237690.995925	...	42815.775409	155395.390030
1	237690.995925	0.000000	...	264889.398939	385435.511309
2	79782.466760	317144.541987	...	74602.554581	87986.061870
3	261389.573558	28999.550044	...	292321.617039	412690.548292
4	53372.216100	288074.733923	...	39959.337646	102457.030136
..
197	60770.233816	281405.644842	...	18533.862289	105666.374403
198	128687.635109	360119.702102	...	96693.282992	27020.702704
199	21714.620373	250061.119850	...	21366.111532	137107.587276
200	42815.775409	264889.398939	...	0.000000	120748.666597
201	155395.390030	385435.511309	...	120748.666597	0.000000

[202 rows x 202 columns]

3. Matrices de distancias

d) Minkowski

```
▶ DstMinkowski = cdist(Hipoteca.iloc[0:10], Hipoteca.iloc[0:10], metric='minkowski', p=1.5)
  MMinkowski = pd.DataFrame(DstMinkowski)
  print(MMinkowski)
```

```
0          0      1    ...      8      9
0  0.000000  237690.995925  ...  109035.213044  78197.161473
1  237690.995925        0.000000  ...  346609.614856  312975.503513
2  79782.466760  317144.541987  ...  32977.126225  17574.226078
3  261389.573558  28999.550044  ...  370236.872408  338719.479124
4  53372.216100  288074.733923  ...  60284.016224  25100.249754
5  39260.690697  276926.258979  ...  69936.944305  40487.806354
6  30673.683784  207408.636739  ...  139247.210167  106708.022220
7  207250.873149  36799.022688  ...  315980.319533  284964.264428
8  109035.213044  346609.614856  ...        0.000000  36693.205417
9  78197.161473  312975.503513  ...  36693.205417        0.000000
```

[10 rows x 10 columns]

3. Matrices de distancias

d) Minkowski (entre dos puntos)



```
Objeto1 = Hipoteca.iloc[0]
Objeto2 = Hipoteca.iloc[1]
dstMinkowski = distance.minkowski(Objeto1,Objeto2)
dstMinkowski
```

```
236994.70196398906
```

Otras mediciones

```
▶ from scipy.spatial import distance  
E1 = (10000,1,0,0,0,0,7,15,1)  
E2 = (20000,0,1,1,0,1,3,3,0)  
dstEuclidean = distance.euclidean(E1,E2)
```

10000.0082499

```
▶ from scipy.spatial import distance  
E1 = (10000,1,0,0,0,0,7,15,1)  
E2 = (20000,0,1,1,0,1,3,3,0)  
dstChebyshev = distance.chebyshev(E1,E2)
```

dstChebyshev

10000

```
▶ from scipy.spatial import distance  
E1 = (10000,1,0,0,0,0,7,15,1)  
E2 = (20000,0,1,1,0,1,3,3,0)  
dstManhattan =  
dstManhattan
```

10021

```
▶ from scipy.spatial import distance  
E1 = (10000,1,0,0,0,0,7,15,1)  
E2 = (20000,0,1,1,0,1,3,3,0)  
dstMinkowski = distance.minkowski(E1,E2, 1.5)  
dstMinkowski
```

10000.363791487287



Universidad Nacional Autónoma de México
Facultad de Ingeniería

Selección de características

Guillermo Molero-Castillo

guillermo.molero@ingenieria.unam.edu

Octubre, 2021

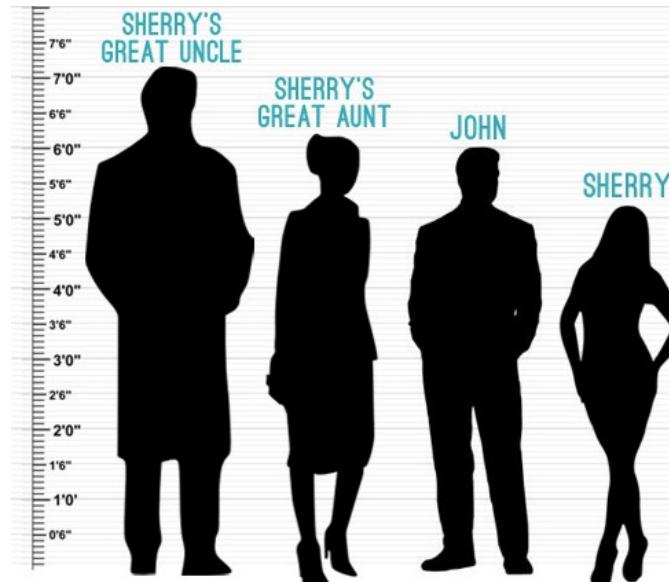
Selección de características

Existen factores que se deben tener en cuenta a la hora de seleccionar un algoritmo en **Machine Learning**.

- En realidad es un arte, y entre lo más importante se encuentra decidir qué datos de entrada va a recibir el sistema.
- **Ejemplo:** Clasificar personas por su **estatura y sexo** a través de imágenes de cámaras de seguridad.

- Estatura
- Tipo de vestimenta
- Complexión física
- Color de piel
- Tipo de peinado
- Tipo de zapatos
- Sombrero
- Otras particularidades (joroba)

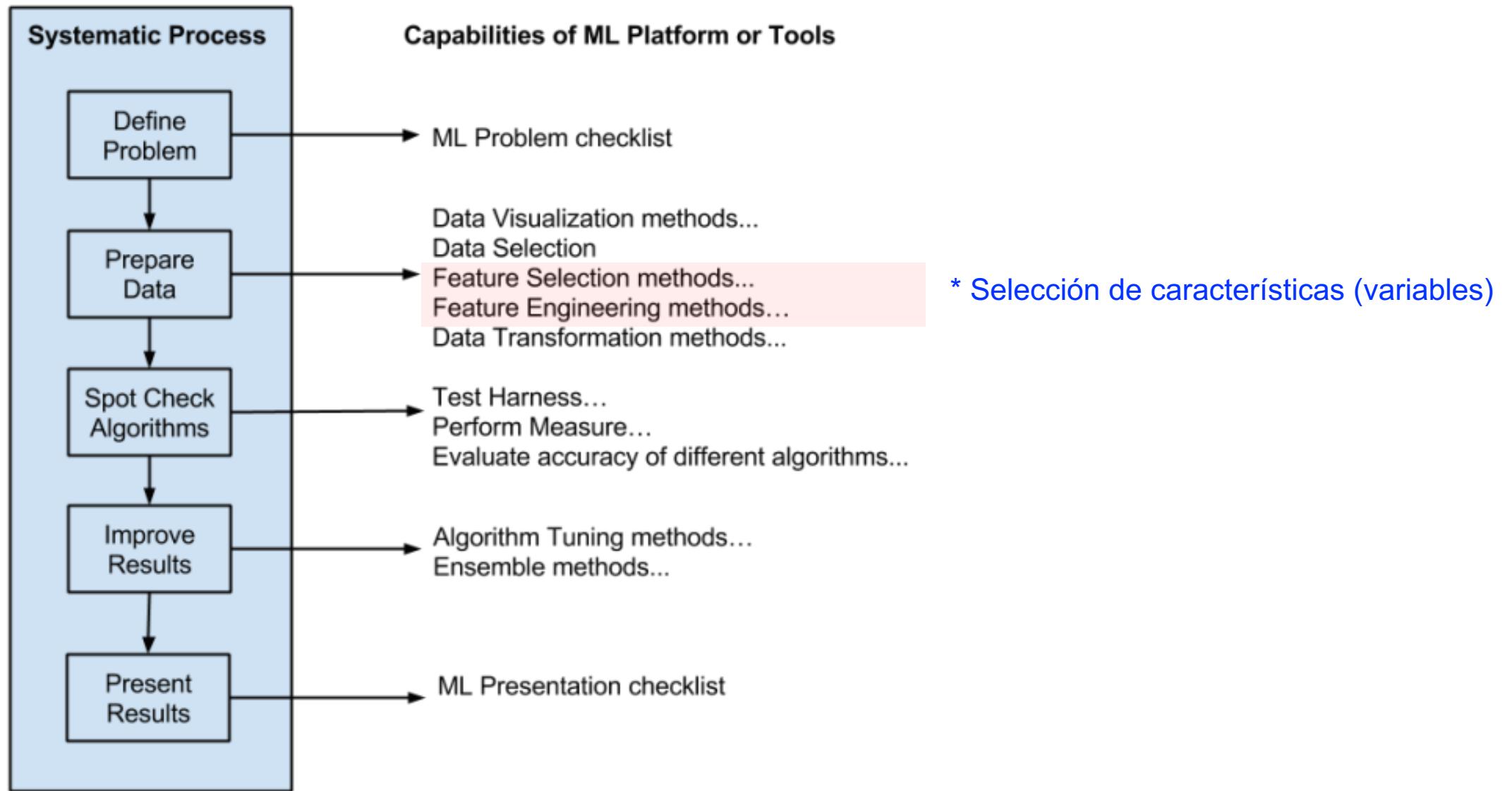
Tamaño = 8 dimensiones



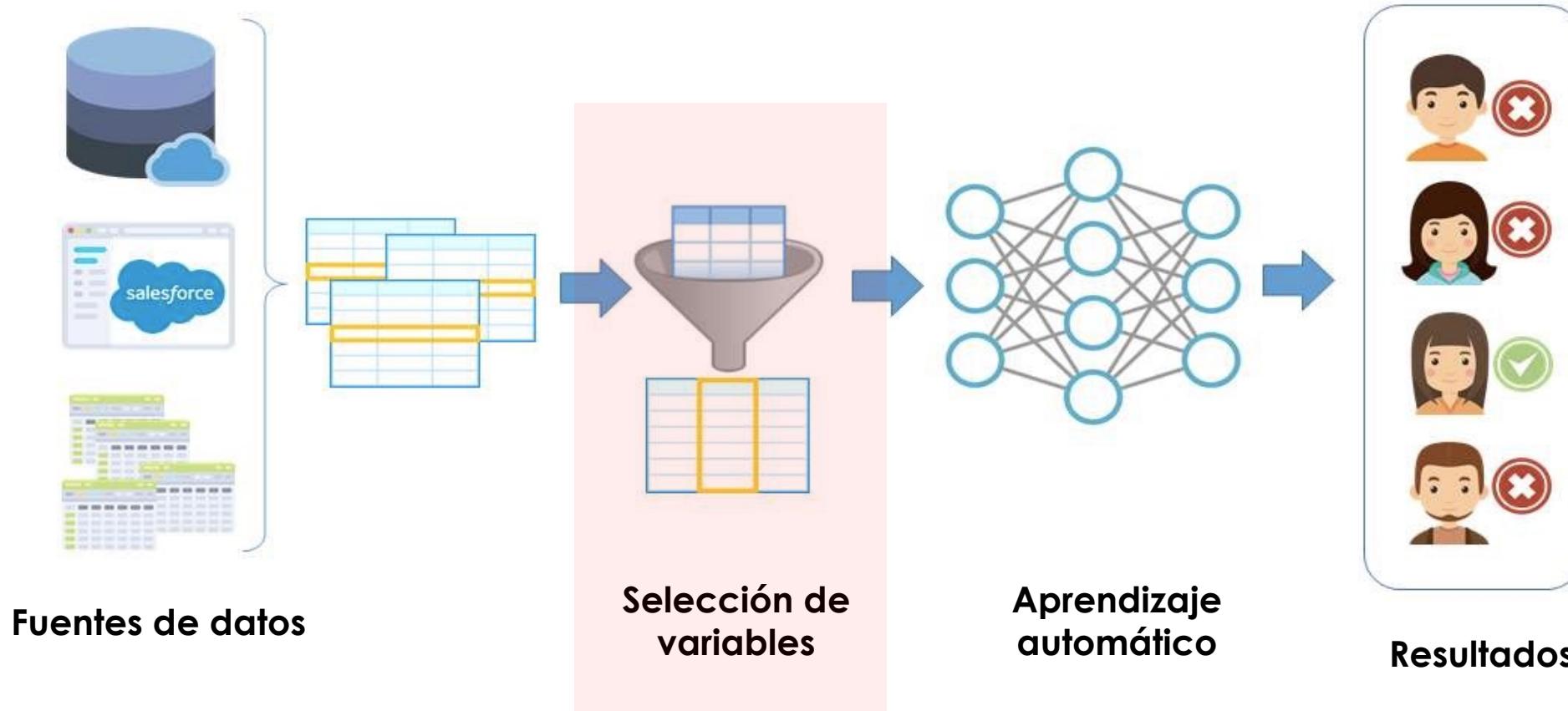
- Estatura
- Tipo de vestimenta
- Complexión física

Tamaño = 3 dimensiones

Selección de características



Selección de características

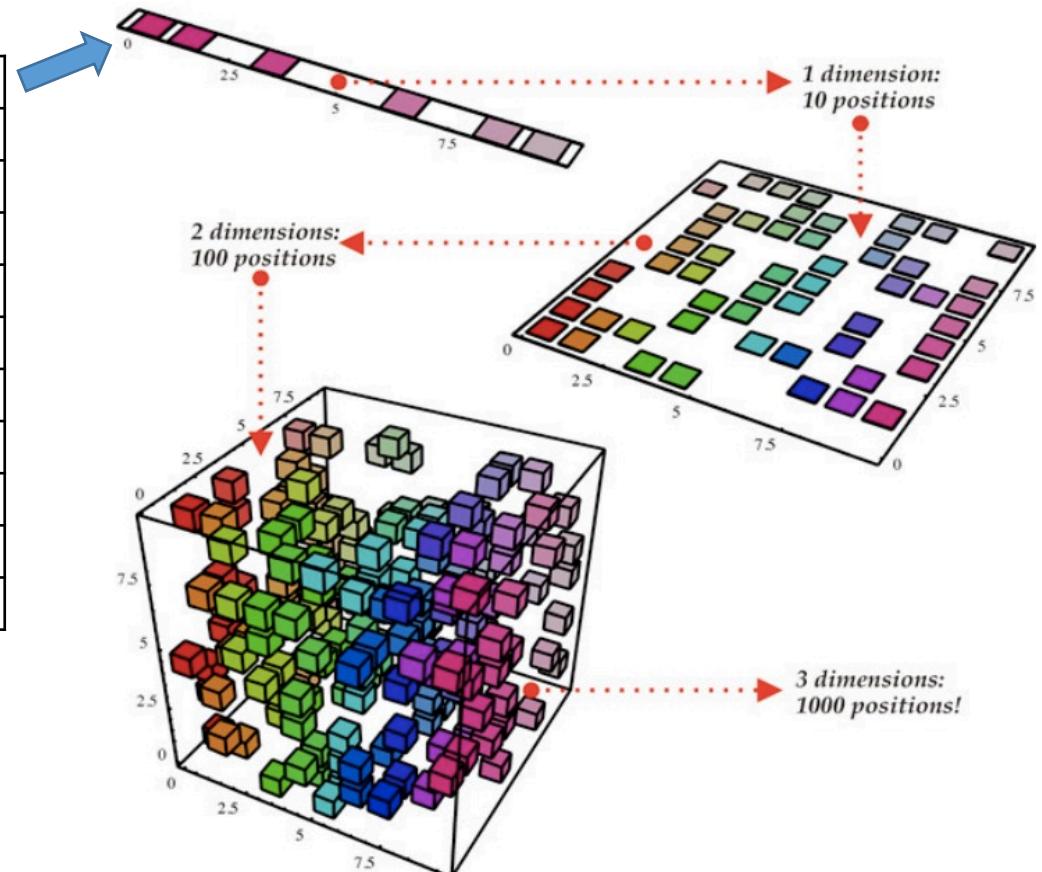


Selección de características

A menudo hay demasiadas **variables**, en función de las cuales se condiciona el resultado final de un modelo.

- Cuanto mayor es el número de variables, es más complejo visualizar los datos y más complejo aún trabajar con éstos.
- No es ilógico pensar que entre más variables (características/atributos) se tenga en cuenta será mejor.
- Sin embargo, esto es un **error común** que no se debe cometer.

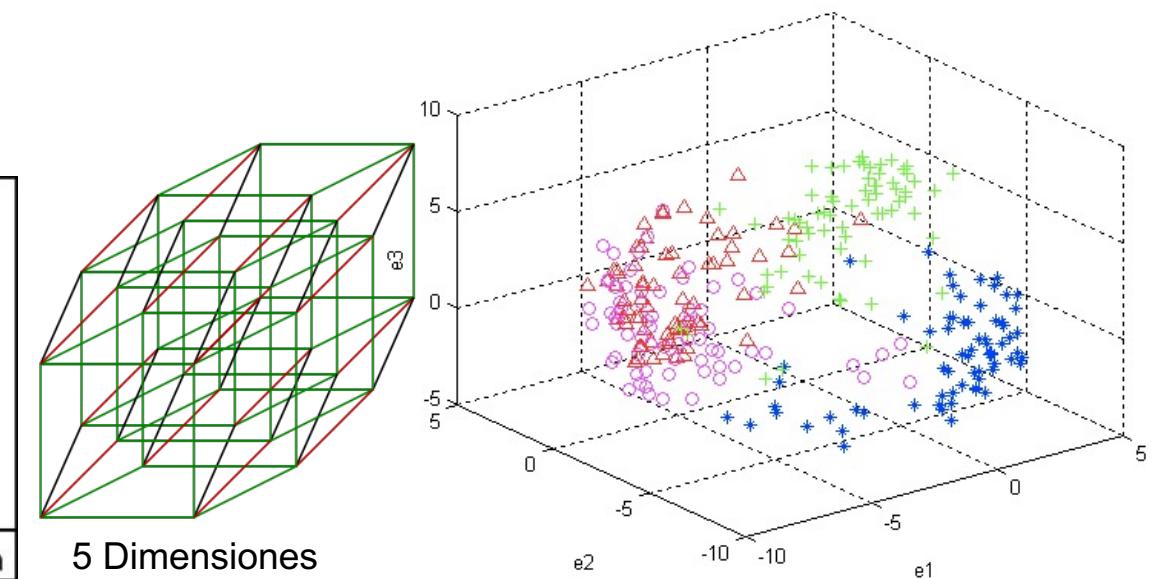
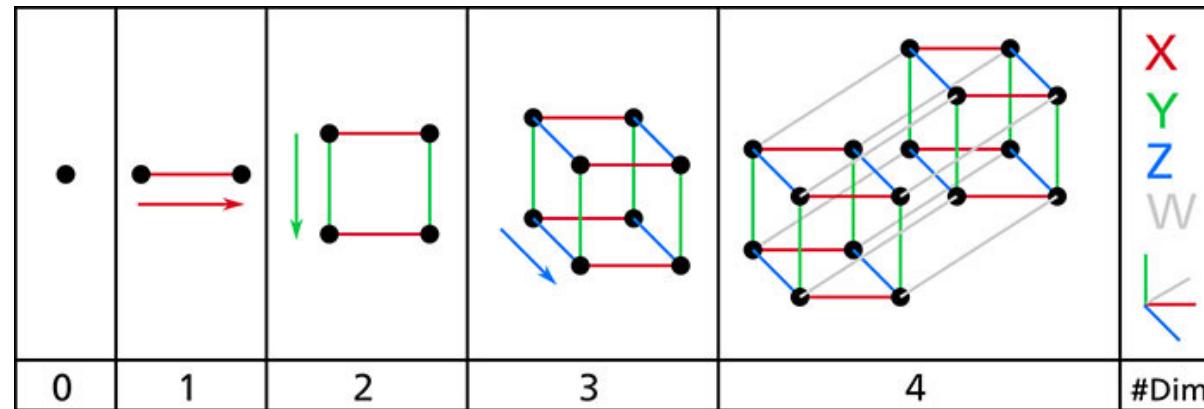
	F ₁
E ₁	3
E ₂	2
E ₃	5
E ₄	10
E ₅	3
E ₆	4
E ₇	6
E ₈	7
E ₉	9
E ₁₀	1



Dimensionalidad de datos

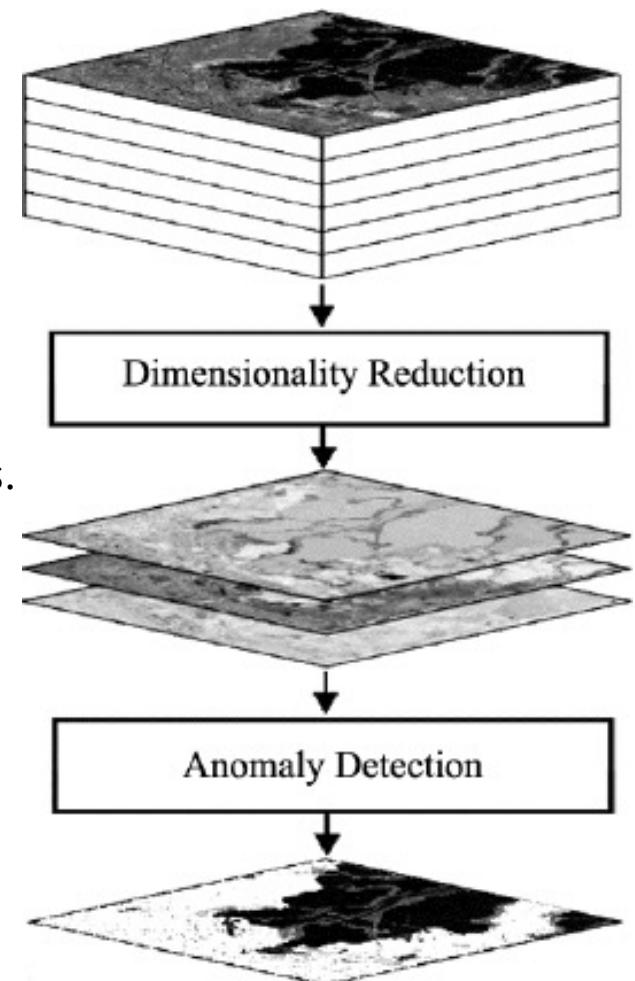
Maldición de la dimensionalidad

- La **maldición de la dimensionalidad de datos** es un problema que se puede presentar si se quiere tener en cuenta todas las características (variables) posibles en un sistema.
- Esta **maldición** hace referencia al aumento exponencial de la dimensionalidad de datos.



Maldición de la dimensionalidad

- En general, si la mayoría de las **variables están correlacionadas**, entonces algunas de éstas son redundantes.
- Aquí es donde se utilizan estrategias para la reducción de la dimensionalidad de datos.
- Esta **reducción de la dimensionalidad** es el proceso de aminorar el número de variables mediante la obtención de alguna función de puntuación, que generalmente mide la relevancia de las características.

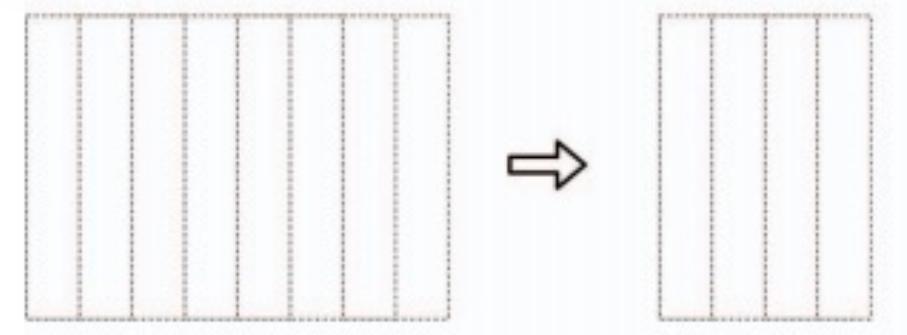


Maldición de la dimensionalidad

Feature selection

- Es el proceso de ordenar las variables por el valor de alguna función de puntuación.
- Para reducir la maldición de la dimensionalidad existen algunas estrategias:
 - Discriminación manual, pero tiene limitaciones.
 - Análisis correlacional de datos (Correlational Data Analysis, **CDA**)
 - Análisis de componentes principales (Principal Component Analysis, **PCA**)
 - Análisis discriminante lineal (Linear Discriminant Analysis, **LDA**)
 - Análisis discriminante generalizado (Generalized Discriminant Analysis, **GDA**)

Feature Selection



Maldición de la dimensionalidad

Ventajas de la reducción de dimensionalidad

- Ayuda en la reducción del espacio de almacenamiento.
- Reduce el tiempo de cálculo.
- Ayuda a eliminar variables redundantes, si las hay.

Desventaja de la reducción de dimensionalidad

- Si no se hace un análisis cuidadoso, puede provocar pérdida de datos valiosos.

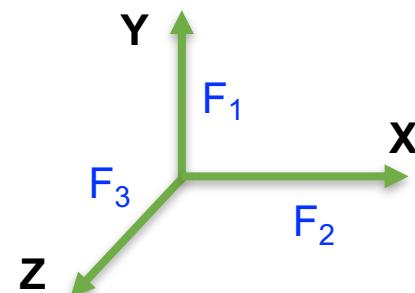
Análisis correlacional de datos

Análisis correlacional de datos

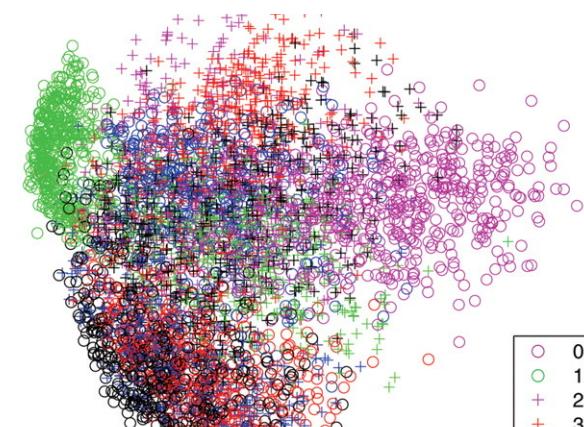
Correlaciones

- El **ACD (CDA)** es útil para reducir el número de variables, de un espacio de alta dimensión a uno de menor número.
- Esto se logra a través de la identificación de correlaciones (**grado de similitud**) entre pares de variables numéricas.

	F_1	F_2	F_3
E_1	3	5	0
E_2	2	1	1
E_3	5	2	0
...
E_n	1	2	0



Existe una superposición de los datos

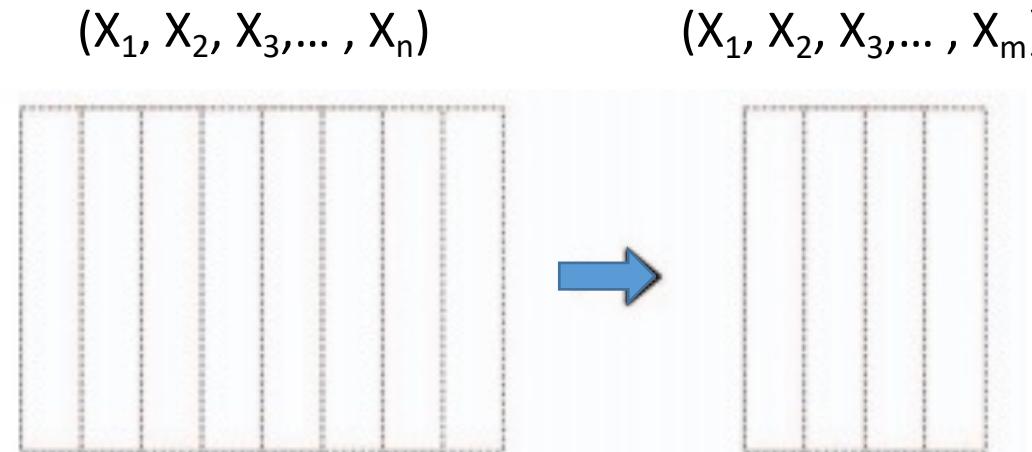


○	0
○	1
+	2
+	3
○	4
+	5
+	6
○	7
○	8
+	9

Análisis correlacional de datos

Correlaciones

- La reducción consiste en que a partir de un conjunto de **variables originales**: $X_1, X_2, X_3, \dots, X_n$
- Se obtiene otro subconjunto de **variables relevantes** : $X_1, X_2, X_3, \dots, X_m$, donde $m < n$.

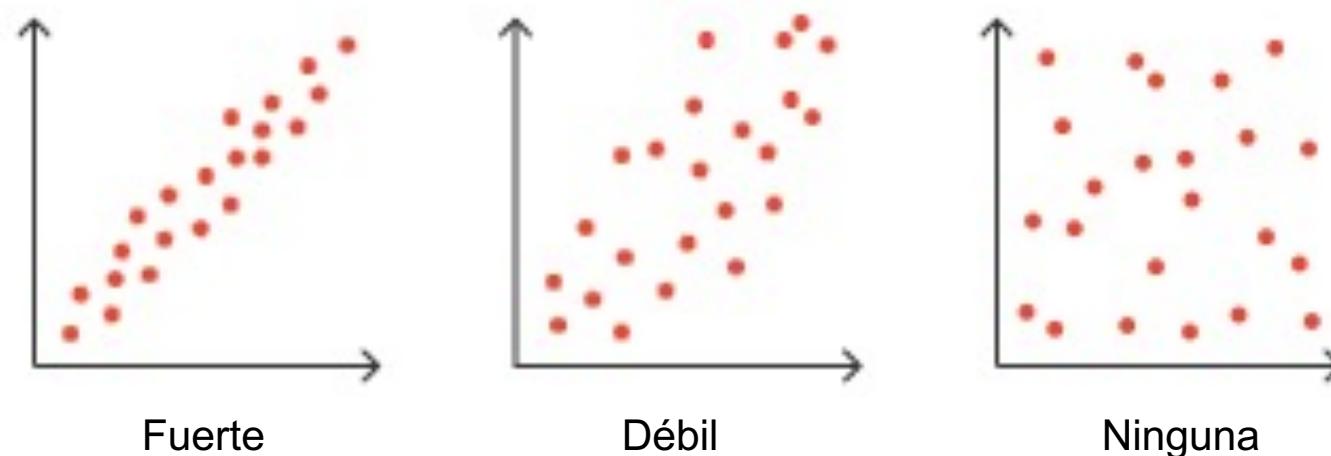


Análisis correlacional de datos

Como paso inicial

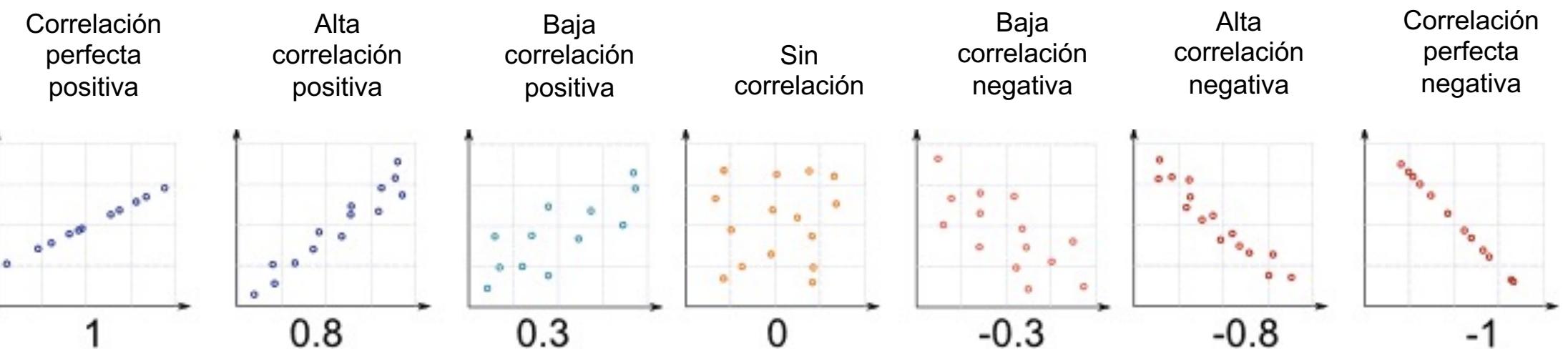
- Es importante hacer una evaluación visual de los datos a través de gráficos de dispersión.
 - Estos gráficos utilizan una colección de puntos (vectores de datos) para mostrar los valores de dos variables.

Fuerza de correlación



Análisis correlacional de datos

Como paso inicial



Análisis correlacional de datos

Coeficiente de correlación

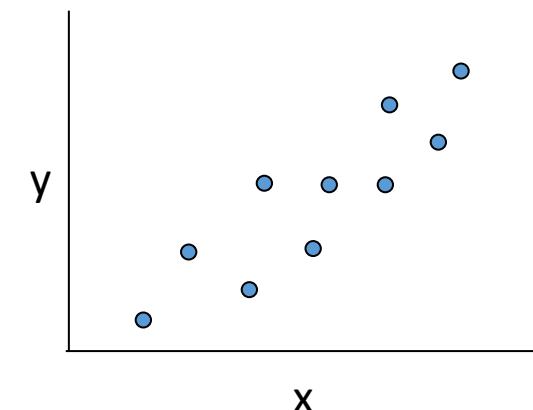
Los valores de correlación, conocidos como coeficiente de correlación de Pearson (su creador, Karl Pearson, 1857-1936), se define como:

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}}$$

\bar{x}, \bar{y} son las medias aritméticas de x e y .

Covarianza

Varianza



Los valores de correlación, en este caso **r** o **R**, pueden variar entre **-1 y 1**.

Análisis correlacional de datos

Coeficiente de correlación

- Cuanto **más cerca está R de 1 o -1**, más fuerte es la correlación.
- Si **R es cercano a -1** las variables están correlacionadas negativamente.
- Si **R es cero** no existe correlación.

Intervalos utilizados para la identificación de correlaciones (Opción sugerida):

- De **-1.0 a -0.67 y 0.67 a 1.0** se conocen como correlaciones **fuertes o altas**.
- De **-0.66 a -0.34 y 0.34 a 0.66** se conocen como correlaciones **moderadas o medias**.
- De **-0.33 a 0.0 y 0.0 a 0.33** se conocen como correlaciones **débiles o bajas**.

Otras opciones utilizadas:

- De **-1.0 a -0.70 y 0.70 a 1.0** se conocen como correlaciones **fuertes o altas**.
- De **-0.69 a -0.31 y 0.31 a 0.70** se conocen como correlaciones **moderadas o medias**.
- De **-0.30 a 0.0 y 0.0 a 0.30** se conocen como correlaciones **débiles o bajas**.

Análisis correlacional de datos

Matriz de correlaciones

- Consiste en crear una **matriz** que aporta información sobre la relación entre pares de variables.
- El objetivo es obtener, a partir de esta matriz, un subconjunto de variables representativas que no tengan dependencia entre sí.

1	$r(X_1, X_2)$	$r(X_1, X_3)$	$r(X_1, X_4)$	$r(X_1, X_5)$	$r(X_1, X_6)$	$r(X_1, X_7)$	$r(X_1, X_8)$
$r(X_2, X_1)$	1	$r(X_2, X_3)$	$r(X_2, X_4)$	$r(X_2, X_5)$	$r(X_2, X_6)$	$r(X_2, X_7)$	$r(X_2, X_8)$
$r(X_3, X_1)$	$r(X_3, X_2)$	1					
$r(X_4, X_1)$	$r(X_4, X_2)$		1				
$r(X_5, X_1)$	$r(X_5, X_2)$			1			
$r(X_6, X_1)$	$r(X_6, X_2)$				1		
$r(X_7, X_1)$	$r(X_7, X_2)$					1	
$r(X_8, X_1)$	$r(X_8, X_2)$	$r(X_8, X_3)$	$r(X_8, X_4)$	$r(X_8, X_5)$	$r(X_8, X_6)$	$r(X_8, X_7)$	1

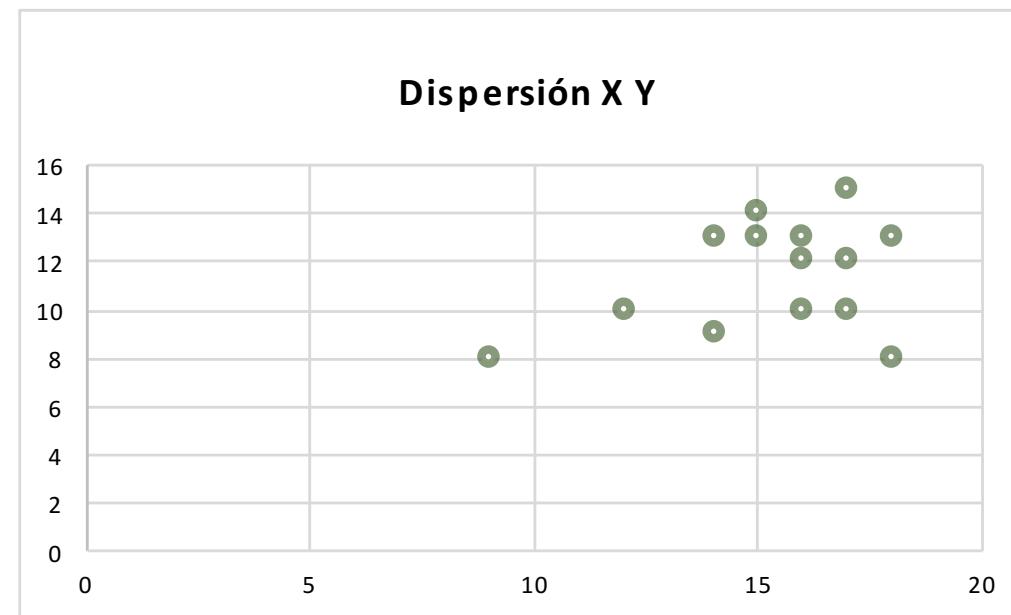
Ejemplo ilustrativo

Ejemplo ilustrativo

Sean las temperaturas de dos ciudades (**X** –Ciudad de México–, **Y** –Puebla–), determinar el coeficiente de correlación de Pearson:

Día	X	Y
Día 1	18	13
Día 2	17	15
Día 3	15	14
Día 4	16	13
Día 5	14	9
Día 6	12	10
Día 7	9	8
Día 8	15	13
Día 9	16	12
Día 10	14	13
Día 11	16	10
Día 12	18	8
Día 13	17	10
Día 14	17	12

Diagrama de dispersión



Ejemplo ilustrativo

Sean las temperaturas de dos ciudades (**X** –Ciudad de México–, **Y** –Puebla–), determinar el coeficiente de correlación de Pearson:

Día	X	Y	$x = X - \bar{X}'$	$y = Y - \bar{Y}'$	x^2	y^2	xy
Día 1	18	13	2.71	1.57	7.37	2.47	4.27
Día 2	17	15	1.71	3.57	2.94	12.76	6.12
Día 3	15	14	-0.29	2.57	0.08	6.61	-0.73
Día 4	16	13	0.71	1.57	0.51	2.47	1.12
Día 5	14	9	-1.29	-2.43	1.65	5.90	3.12
Día 6	12	10	-3.29	-1.43	10.80	2.04	4.69
Día 7	9	8	-6.29	-3.43	39.51	11.76	21.55
Día 8	15	13	-0.29	1.57	0.08	2.47	-0.45
Día 9	16	12	0.71	0.57	0.51	0.33	0.41
Día 10	14	13	-1.29	1.57	1.65	2.47	-2.02
Día 11	16	10	0.71	-1.43	0.51	2.04	-1.02
Día 12	18	8	2.71	-3.43	7.37	11.76	-9.31
Día 13	17	10	1.71	-1.43	2.94	2.04	-2.45
Día 14	17	12	1.71	0.57	2.94	0.33	0.98
Total	214	160			78.86	65.43	26.29
Media (\bar{X}')	15.29						
Media (\bar{Y}')	11.43						

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

$$r = \frac{\sum xy}{\sqrt{(\sum x^2)(\sum y^2)}}$$

$$r = \frac{26.29}{\sqrt{78.86 * 65.43}} = \frac{26.29}{71.83} = 0.36$$

¿Qué pasa con variables cualitativas?

Variables cualitativas

En el caso de variables cualitativas

Pacientes, 7 variables:

- **Capacidad.** Capacidad del paciente para acudir a una consulta. (1-10)
- **Necesidad.** Importancia que le da el paciente a la consulta médica. (1-10)
- **Transporte.** Disponibilidad de transporte del paciente. (1-10)
- **Cuidado.** Disponibilidad para tener el cuidado de los niños. (1-10)
- **Permiso.** En caso de trabajar, facilidad para solicitar permisos médicos. (1-10)
- **Satisfacción.** Satisfacción del cliente con la atención médica. (1-10)
- **Facilidad.** Facilidad para obtener una cita y eficiencia de la misma. (1-10)
- **Visita.** Visita del paciente durante el último año (0 - no visitó, 1 - si visitó)

Variables cualitativas

En el caso de variables cualitativas

	Capacidad	Importancia	Transporte	Cuidado	Permito	Satisfacción	Facilidad	Visita
Capacidad	1							
Importancia	-0.737	1						
Transporte	0.312	-0.0104	1					
Cuidado	0.312	-0.0104	0.379	1				
Permito	0.277	0.060	0.623	0.623	1			
Satisfacción	0.220	-0.134	0.654	0.654	0.626	1		
Facilidad	0.389	-0.033	0.650	0.650	0.659	0.896	1	
Visita	0.396	-0.542	-0.503	-0.503	-0.425	-0.399	-0.328	1

- **R1.** Existe una relación fuerte (negativa) entre la **capacidad** que tiene el paciente para acudir a una consulta y la **Importancia** que le da el paciente a la consulta médica.
- **R2.** Se tiene una relación fuerte (positiva) entre la **satisfacción** del paciente con la atención médica y la **facilidad** que tiene para obtener una cita.

Ejemplo en Google Colab

Contexto

- El sector inmobiliario de Melbourne está en auge.
- Objetivo: Encontrar información de interés para predecir la próxima tendencia inmobiliaria.
- Dado que cada vez es más difícil adquirir una unidad de 2 dormitorios a un precio razonable.

The screenshot shows a dataset page on Kaggle. The title is "Melbourne Housing Snapshot" with the subtitle "Snapshot of Tony Pino's Melbourne Housing Dataset". It features a background image of a Melbourne cityscape. A user profile for "DanB" is shown, indicating the dataset was updated 3 years ago (Version 5). Below the title, there are tabs for "Data", "Tasks (1)", "Code (3,509)", "Discussion (2)", "Activity", and "More". There are also buttons for "Download (451 KB)" and "New Notebook". At the bottom, there are sections for "Usability 7.1", "License CC BY-NC-SA 4.0", and "Tags social science, real estate, demographics, housing, australia". Below these are sections for "Description" and "Context".

Fuente: <https://www.kaggle.com/dansbecker/melbourne-housing-snapshot>

Diccionario de datos

Item	Column name	Definición
1	Rooms	Número de habitaciones
2	Price	Precio en dólares
3	Method	S - propiedad vendida; SP - propiedad vendida antes; PI - propiedad transferida; PN - vendida antes no revelada; SN - vendida no revelada; NB - sin oferta; VB - oferta del proveedor; W - retirada antes de la subasta; SA - vendida después de subasta; SS - vendida después del precio de subasta no revelado. N/A - precio u oferta más alta no disponible.
4	Type	br - dormitorio (s); h - casa, cabaña, villa, semi, terraza; u - unidad, dúplex; t - casa adosada; dev site – en desarrollo; o res - otro residencial.
5	SellerG	Agente de bienes raíces
6	Date	Fecha de venta
7	Distance	Distancia del CBD (Centro de negocios)
8	Regionname	Región general (oeste, noroeste, norte, noreste ...)
9	Propertycount	Número de propiedades que existen en el suburbio
10	Bedroom2	Número de dormitorios (de otra fuente)
11	Bathroom	Cantidad de baños
12	Car	Número de estacionamientos
13	Landsize	Tamaño del terreno
14	BuildingArea	Tamaño del edificio
15	CouncilArea	Consejo de gobierno de la zona (Municipio)

Importar las bibliotecas y datos

```
▶ import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
%matplotlib inline  
  
[ ] from google.colab import files  
files.upload()
```

Datos

- El conjunto de datos corresponde a **Melbourne Housing Snapshot de Kaggle**. Este conjunto de datos incluye: dirección, tipo de inmueble, suburbio, método de venta, habitaciones, precio, agente inmobiliario, fecha de venta y Distancia desde C.B.D. (Distrito Central de Negocios).

Fuente: <https://www.kaggle.com/dansbecker/melbourne-housing-snapshot>

Importar las bibliotecas y datos

```
[3] DatosMelbourne = pd.read_csv('melb_data.csv')
```

```
[4] DatosMelbourne
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance	Postcode	Bedroom2
0	Abbotsford	85 Turner St	2	h	1480000.0	S	Biggin	3/12/2016	2.5	3067.0	2.0
1	Abbotsford	25 Bloomberg St	2	h	1035000.0	S	Biggin	4/02/2016	2.5	3067.0	2.0
2	Abbotsford	5 Charles St	3	h	1465000.0	SP	Biggin	4/03/2017	2.5	3067.0	3.0

Objetivo:

- Una buena práctica es observar los datos para tener una imagen clara de éstos.
- Si se quiere ver solo las primeras filas se usa `head()`. Por ejemplo: `DatosMelbourne.head()`

1. Tipos de datos



DatosMelbourne.dtypes

```
Suburb          object
Address         object
Rooms           int64
Type            object
Price           float64
Method          object
SellerG         object
Date            object
Distance        float64
Postcode        float64
Bedroom2        float64
Bathroom        float64
Car             float64
Landsize        float64
BuildingArea    float64
YearBuilt       float64
CouncilArea     object
Latitude        float64
Longitude       float64
Regionname      object
Propertycount   float64
dtype: object
```

- El atributo `dtypes` muestra los tipos de datos las variables.
- Se observa que el conjunto de datos tiene una combinación de variables categóricas (objeto) y numéricas (flotante e int).

2. Identificación de datos faltantes

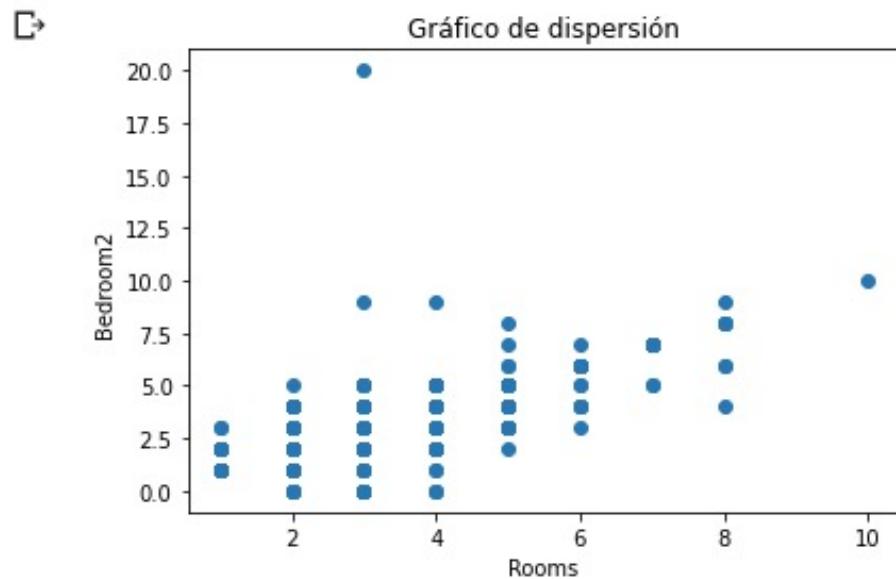
Una función útil de Pandas es `isnull().sum()` que regresa la suma de todos los valores nulos en cada variable.

DatosMelbourne.isnull().sum()

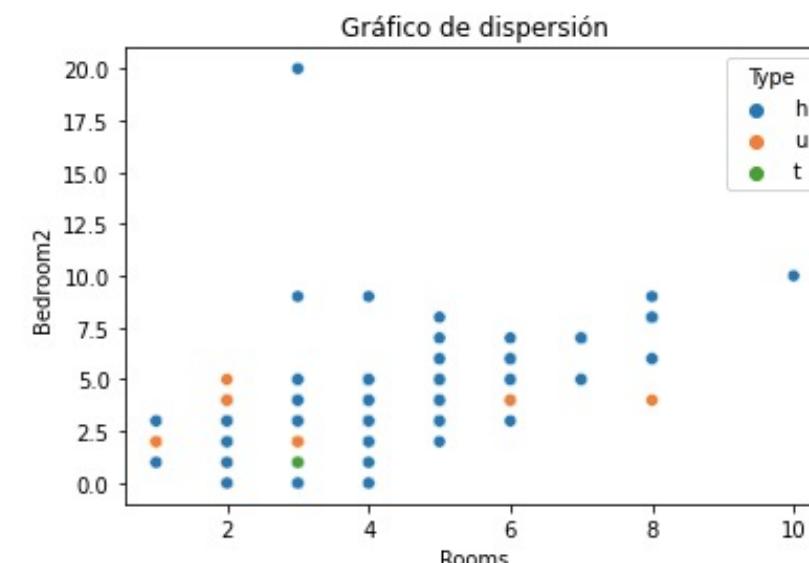
Suburb	0
Address	0
Rooms	0
Type	0
Price	0
Method	0
SellerG	0
Date	0
Distance	0
Postcode	0
Bedroom2	0
Bathroom	0
Car	62
Landsize	0
BuildingArea	6450
YearBuilt	5375
CouncilArea	1369
Latitude	0
Longtitude	0
Regionname	0
Propertycount	0
<i>dtype:</i>	<i>int64</i>

3. Evaluación visual

```
▶ plt.plot(DatosMelbourne['Rooms'], DatosMelbourne['Bedroom2'], 'o')
plt.title('Gráfico de dispersión')
plt.xlabel('Rooms')
plt.ylabel('Bedroom2')
plt.show()
```

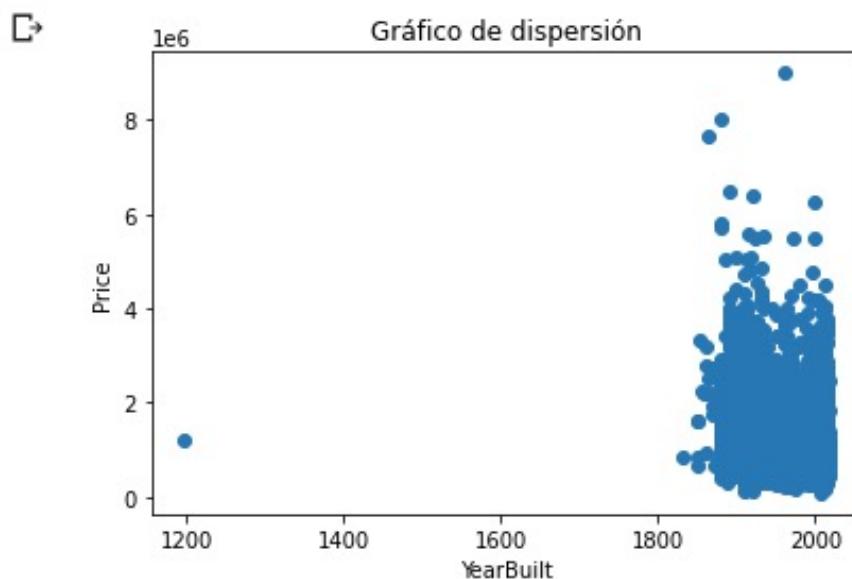


```
▶ sns.scatterplot(x='Rooms', y ='Bedroom2', data=DatosMelbourne, hue='Type')
plt.title('Gráfico de dispersión')
plt.xlabel('Rooms')
plt.ylabel('Bedroom2')
plt.show()
```

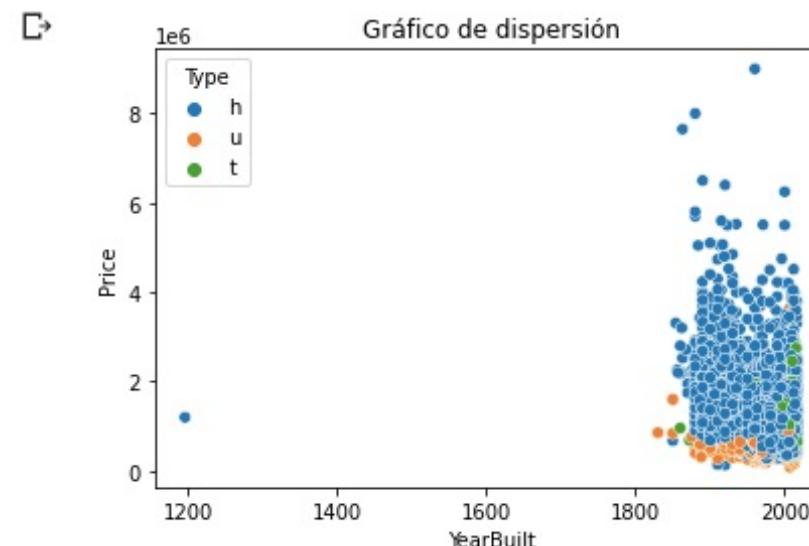


3. Evaluación visual

```
▶ plt.plot(DatosMelbourne['YearBuilt'], DatosMelbourne['Price'], 'o')
plt.title('Gráfico de dispersión')
plt.xlabel('YearBuilt')
plt.ylabel('Price')
plt.show()
```



```
▶ sns.scatterplot(x='YearBuilt', y ='Price', data=DatosMelbourne, hue='Type')
plt.title('Gráfico de dispersión')
plt.xlabel('YearBuilt')
plt.ylabel('Price')
plt.show()
```



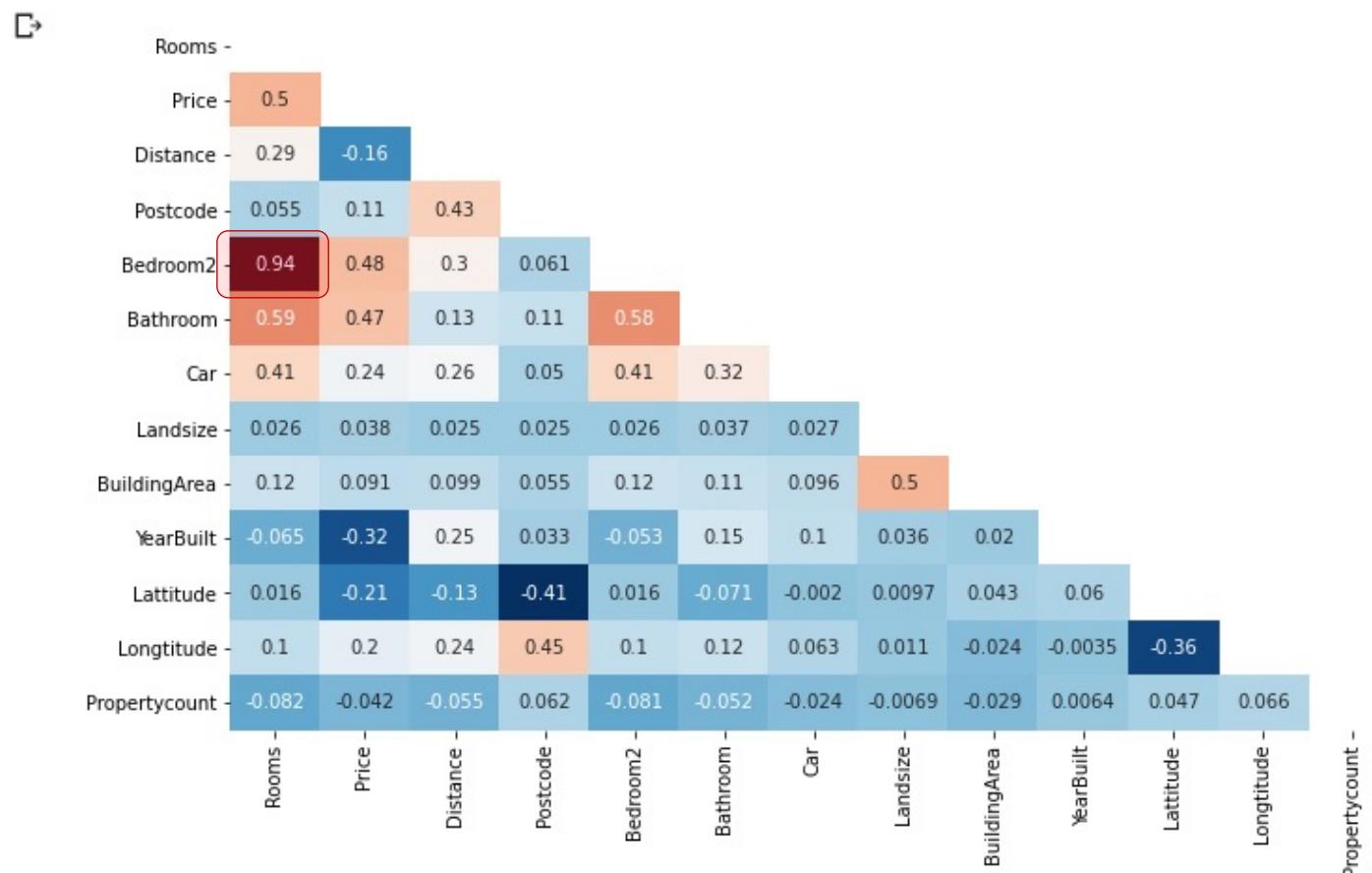
4. Identificación de relaciones entre variables

DatosMelbourne.corr()

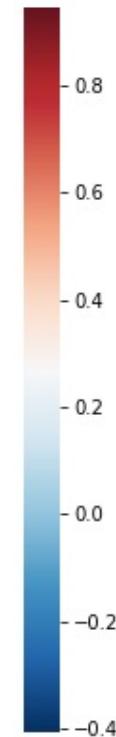
	Rooms	Price	Distance	Postcode	Bedroom2	Bathroom	Car	Landsize	BuildingArea
Rooms	1.000000	0.496634	0.294203	0.055303	0.944190	0.592934	0.408483	0.025678	0.124127
Price	0.496634	1.000000	-0.162522	0.107867	0.475951	0.467038	0.238979	0.037507	0.090981
Distance	0.294203	-0.162522	1.000000	0.431514	0.295927	0.127155	0.262994	0.025004	0.099481
Postcode	0.055303	0.107867	0.431514	1.000000	0.060584	0.113664	0.050289	0.024558	0.055475
Bedroom2	0.944190	0.475951	0.295927	0.060584	1.000000	0.584685	0.405325	0.025646	0.122319
Bathroom	0.592934	0.467038	0.127155	0.113664	0.584685	1.000000	0.322246	0.037130	0.111933
Car	0.408483	0.238979	0.262994	0.050289	0.405325	0.322246	1.000000	0.026770	0.096101
Landsize	0.025678	0.037507	0.025004	0.024558	0.025646	0.037130	0.026770	1.000000	0.500485
BuildingArea	0.124127	0.090981	0.099481	0.055475	0.122319	0.111933	0.096101	0.500485	1.000000
YearBuilt	-0.065413	-0.323617	0.246379	0.032863	-0.053319	0.152702	0.104515	0.036451	0.019665
Latitude	0.015948	-0.212934	-0.130723	-0.406104	0.015925	-0.070594	-0.001963	0.009695	0.043420
Longitude	0.100771	0.203656	0.239425	0.445357	0.102238	0.118971	0.063395	0.010833	-0.023810
Propertycount	-0.081530	-0.042153	-0.054910	0.062304	-0.081350	-0.052201	-0.024295	-0.006854	-0.028840

4. Identificación de relaciones entre variables

```
plt.figure(figsize=(14,7))
MatrizInf = np.triu(DatosMelbourne.corr())
sns.heatmap(DatosMelbourne.corr(), cmap='RdBu_r', annot=True, mask=MatrizInf)
plt.show()
```



- Se traza un mapa de calor a través de la biblioteca de Seaborn.
- La intensidad del color es proporcional a los valores de los coeficientes de correlación.



5. Elección de variables

Variable eliminada con base en el análisis correlacional

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance	Postcode	Bathroom
0	Abbotsford	85 Turner St	2	h	1480000.0	S	Biggin	3/12/2016	2.5	3067.0	1.0
1	Abbotsford	25 Bloomberg St	2	h	1035000.0	S	Biggin	4/02/2016	2.5	3067.0	1.0
2	Abbotsford	5 Charles St	3	h	1465000.0	SP	Biggin	4/03/2017	2.5	3067.0	2.0
3	Abbotsford	40 Federation La	3	h	850000.0	PI	Biggin	4/03/2017	2.5	3067.0	2.0
4	Abbotsford	55a Park St	4	h	1600000.0	VB	Nelson	4/06/2016	2.5	3067.0	1.0
...



Universidad Nacional Autónoma de México
Facultad de Ingeniería

Aprendizaje no supervisado Clustering Jerárquico

Guillermo Molero-Castillo

guillermo.molero@ingenieria.unam.edu

Octubre, 2021

Classical Machine Learning

Task Driven

Supervised Learning

(Pre Categorized Data)



Classification

(Divide the socks by Color)

Eg. Identity Fraud Detection



Regression

(Divide the Ties by Length)

Eg. Market Forecasting

Data Driven

Unsupervised Learning

(Unlabelled Data)

Clustering

(Divide by Similarity)

Eg. Targeted Marketing

Association

(Identify Sequences)

Eg. Customer Recommendation

Dimensionality Reduction

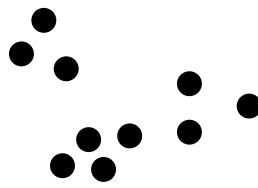
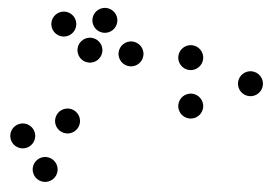
(Wider Dependencies)

Eg. Big Data Visualization

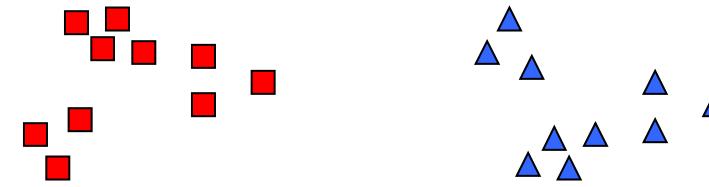
Clustering

Contexto

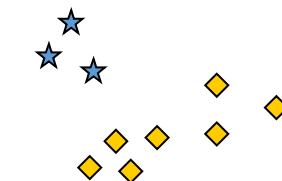
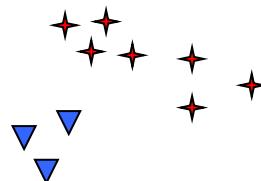
Noción de clúster



¿Cuántos clústeres hay?



Dos clústeres



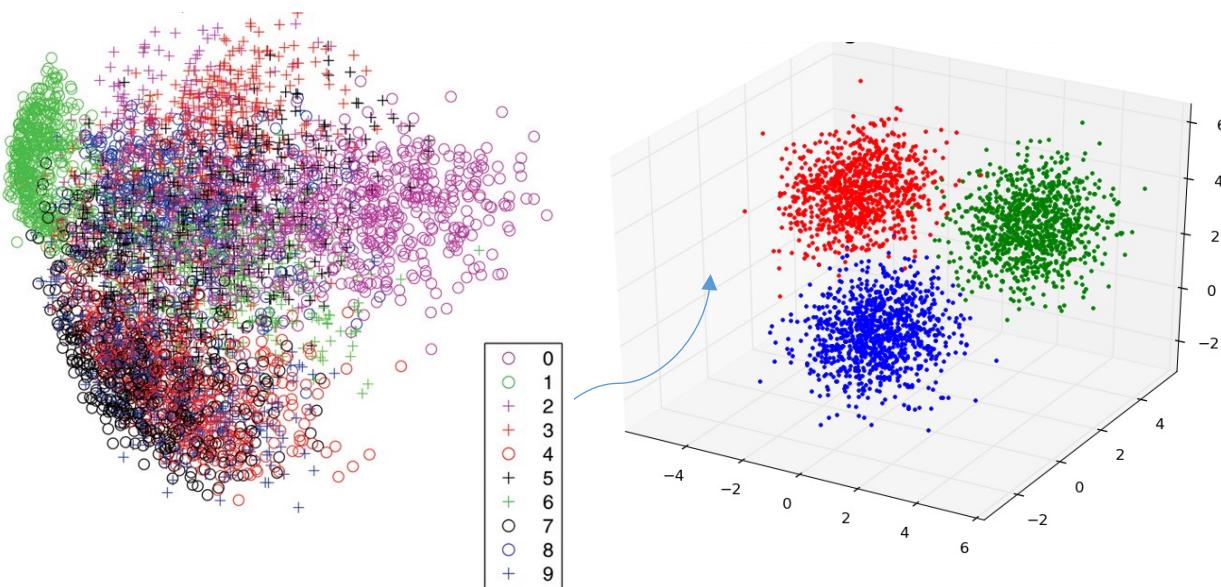
Cuatro clústeres



Seis clústeres

Contexto

- La **IA** aplicada en el análisis clústeres consiste en la segmentación y delimitación de grupos de objetos (elementos), que son unidos por características comunes que éstos comparten (aprendizaje no supervisado).
- El objetivo es dividir una población heterogénea de elementos en un número de grupos naturales (regiones o segmentos homogéneos), de acuerdo a sus similitudes.



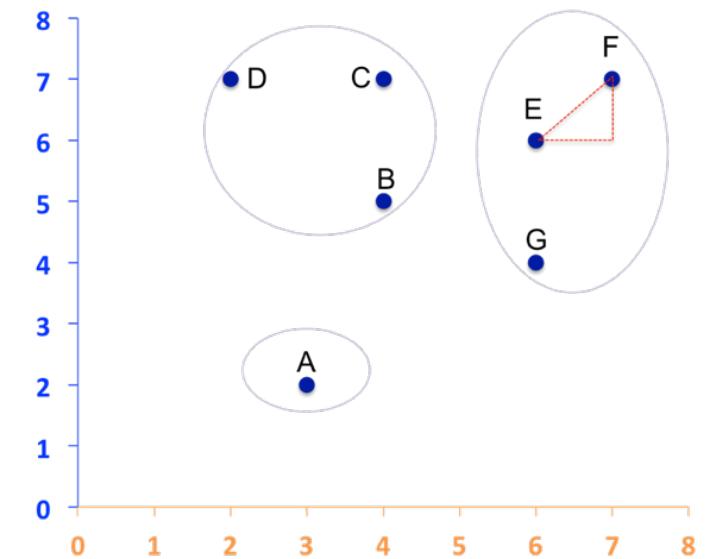
Los grupos nacen a partir de los datos y se descubren una serie de patrones ocultos en éstos.

Contexto

- Para hacer **clustering** es necesario saber el **grado de similitud (medidas de distancia)** entre los elementos.

Sujeto	Lealtad a la tienda (x)	Lealtad a la marca (y)
A	3	2
B	4	5
C	4	7
D	2	7
E	6	6
F	7	7
G	6	4

Sujetos	A	B	C	D	E	F	G
A	---						
B	3.16	---					
C	5.10	2.00	---				
D	5.10	2.83	2.00	---			
E	5.00	2.24	2.24	4.12	---		
F	6.40	3.61	3.00	5.00	1.41	---	
G	3.61	2.24	3.61	5.00	2.00	3.16	---



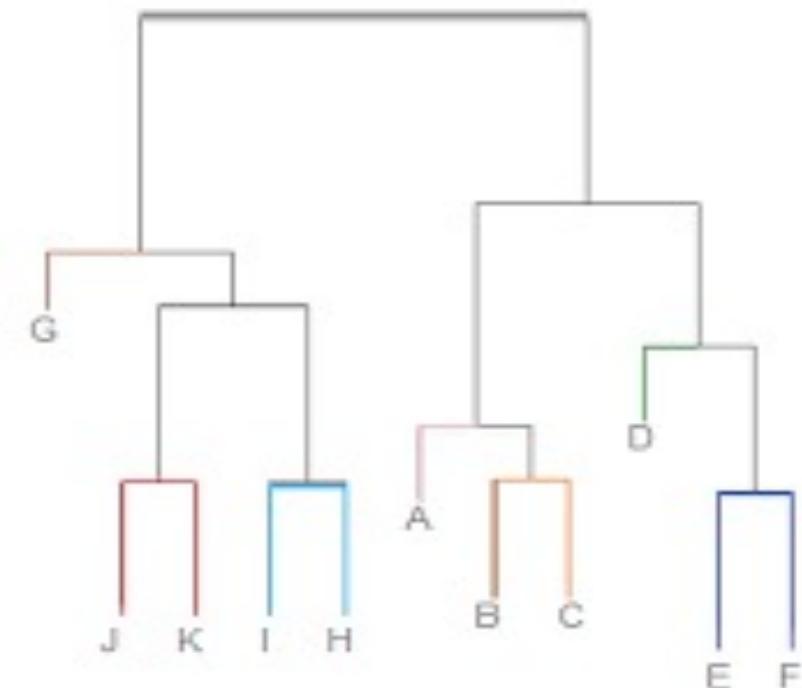
Aplicaciones

- **Marketing.** Para descubrir segmentos de clientes con fines de marketing.
- **Seguro.** Para el análisis de los clientes, sus pólizas e identificar posibles fraudes.
- **Urbanismo.** Para organizar tipos de viviendas y analizar sus valores en función de su ubicación geográfica.
- **Biología.** Para organizar diferentes especies de plantas y animales.
- **Otras.** Estudios demográficos, regiones afectadas por terremotos, identificación de zonas peligrosas, regionalizaciones climáticas, comunidades de usuarios para los sistemas de recomendación, entre otros.

Clustering Jerárquico

Clustering Jerárquico

- El algoritmo de **clustering jerárquico** organiza los elementos, de manera recursiva, en una estructura en forma de árbol.
- Este árbol representa las relaciones de similitud entre los distintos elementos.



Pros

- Facilidad de manejo de los datos.
- No se asume un número particular de grupos.

Contras

- Una vez que se toma la decisión de combinar dos grupos, no se puede regresar atrás.
- Lento para grandes conjuntos de datos, $O(n^2\log(n))$.

Clustering Jerárquico

Pasos para formar grupos (clústeres)

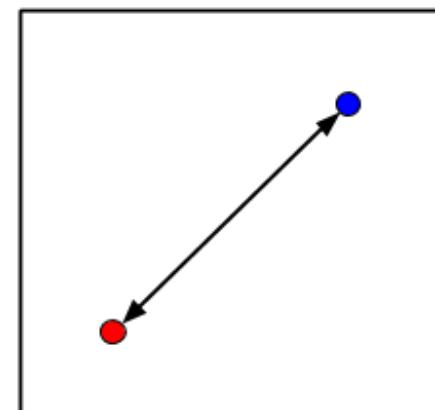
Son cuatro los pasos necesarios:

1. Utilizar un método para medir la similitud de los elementos.
2. Utilizar un método para agrupar a los elementos.
3. Utilizar un método para decidir la cantidad adecuada de grupos.
4. Interpretación de los grupos.

Clustering Jerárquico

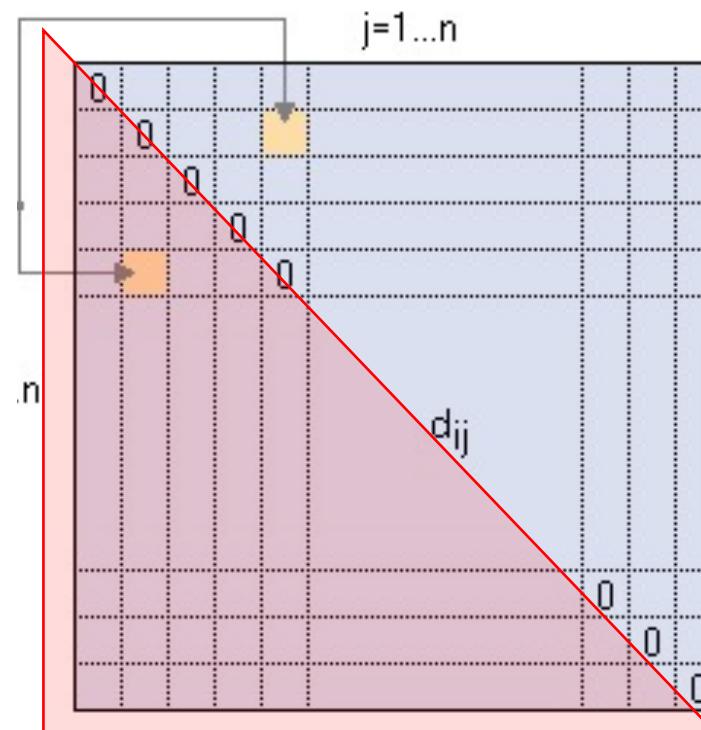
1. Métodos para medir la similitud (algunas métricas conocidas, vistas en clase):

- Distancia Euclidiana (Euclídea).
- Distancia de Chebyshev.
- Distancia de Manhattan (Geometría del taxista).
- Distancia de Minkowsky.



Métricas de distancia

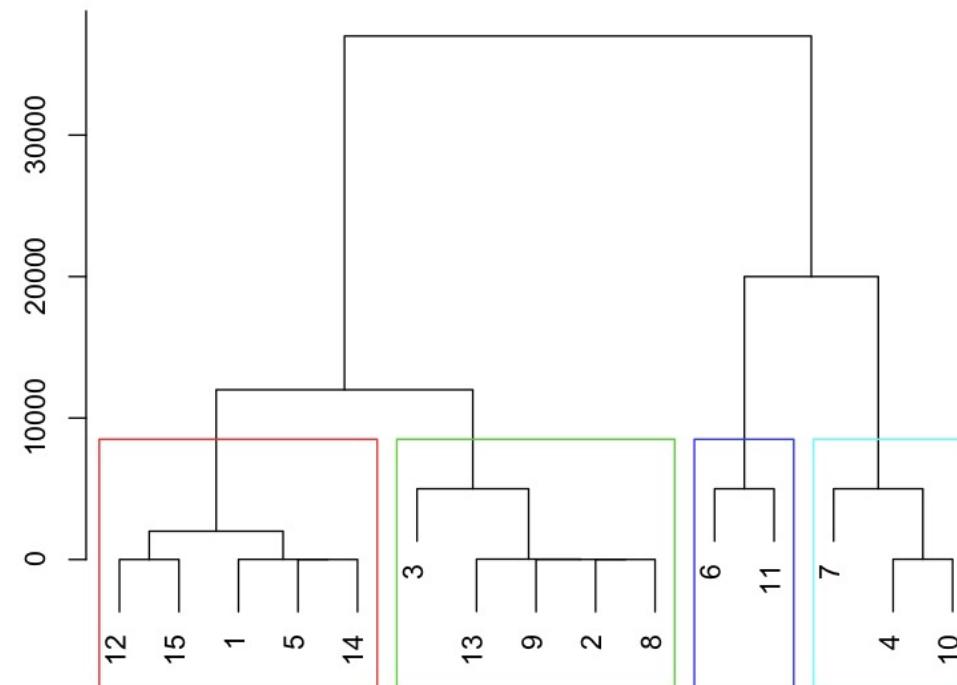
Matriz de distancias



Algoritmo Ascendente Jerárquico

Algoritmo Ascendente Jerárquico

- Consiste en agrupar en cada iteración aquellos 2 elementos más cercanos (clúster) -los de menor distancia-. De esta manera se va construyendo una estructura en forma de árbol.
- El proceso concluye cuando se forma un único clúster (grupo).



Algoritmo Ascendente Jerárquico

Pseudocódigo

- 1 **Calcular** la matriz de distancias/similitud
- 2 **Inicialización:** Cada elemento es un clúster
- 3 **Repetir**
- 4 Combinar los dos clústeres más cercanos
- 5 Actualizar la matriz de distancias/similitud
- 6 **Hasta** que sólo quede un clúster

Cuando se trabaja con **clustering**, dado que son algoritmos basados en distancias, es fundamental estandarizar los datos para que cada una de las variables contribuyan por igual en el análisis.

La razón es que si existen diferencias entre los rangos de las variables, aquellas con rangos más grandes predominarán sobre las que tienen rangos pequeños. Por ejemplo, un rango entre [0 y 100] dominará sobre otra que oscila entre [0 y 1]. Esto dará lugar a resultados sesgados.

$$z = \frac{\text{value} - \text{mean}}{\text{standard deviation}}$$

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

	Feature1	Feature2	Feature13
1	14.230	1.710	1,065.000
2	13.200	1.780	1,050.000
3	13.160	2.360	1,185.000
4	14.370	1.950	1,480.000
5	13.240	2.590	735.000
6	14.200	1.760	1,450.000
7	14.060	2.150	1,295.000
8	14.830	1.640	1,045.000
9	13.860	1.350	1,045.000
10	14.100	2.160	1,510.000
11	14.120	1.480	1,280.000
12	13.750	1.730	1,320.000

	Feature1	Feature2	Feature13
1	0.842	0.192	0.561
2	0.571	0.206	0.551
3	0.561	0.320	0.647
4	0.879	0.239	0.857
5	0.582	0.366	0.326
6	0.834	0.202	0.836
7	0.797	0.279	0.725
8	1.000	0.178	0.547
9	0.745	0.121	0.547
10	0.808	0.281	0.879
11	0.813	0.146	0.715
12	0.716	0.196	0.743

Algoritmo Ascendente Jerárquico

Pseudocódigo

- 1 **Calcular** la matriz de distancias/similitud
- 2 **Inicialización:** Cada elemento es un clúster
- 3 **Repetir**
- 4 Combinar los dos clústeres más cercanos
- 5 Actualizar la matriz de distancias/similitud
- 6 **Hasta** que sólo quede un clúster

Given:

A set X of objects $\{x_1, \dots, x_n\}$
A distance function $dist(c_1, c_2)$

for $i = 1$ to n

$c_i = \{x_i\}$

end for

$C = \{c_1, \dots, c_n\}$

$I = n+1$

while $C.size > 1$ **do**

- $(c_{min1}, c_{min2}) = \text{minimum } dist(c_i, c_j) \text{ for all } c_i, c_j \text{ in } C$
- remove c_{min1} and c_{min2} from C
- add $\{c_{min1}, c_{min2}\}$ to C
- $I = I + 1$

end while

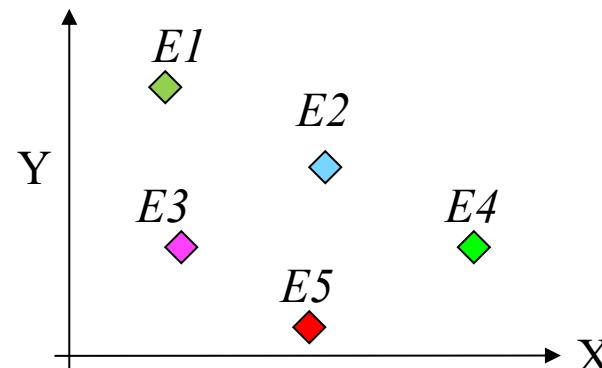
Algoritmo Ascendente Jerárquico

Procedimiento

1

Construir una matriz de distancias entre los elementos. **Por ejemplo, Euclidiana.**

	F1	F2	F3	...	F10
E1	1	4	0.5		3
E2	2	6	0.7		2
E3	4	4	0.4		4
E4	6	2	0.2		2
E5	7	2	0.2		2



$$dist(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

d	1	2	3	4	5
1	0	5	6	10	13
2	5	0	1	5	8
3	6	1	0	4	7
4	10	5	4	0	3
5	13	8	7	3	0

Al tener 5 elementos, la matriz de distancias es de 5×5 .
17

Algoritmo Ascendente Jerárquico

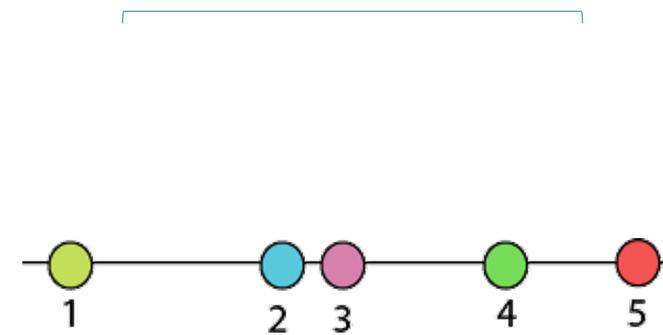
Procedimiento

2

Cada elemento representa un grupo (clúster).

	F1	F2	F3	...	F10
E1	1	4	0.5		3
E2	2	6	0.7		2
E3	4	4	0.4		4
E4	6	2	0.2		2
E5	7	2	0.2		2

5 clústeres iniciales



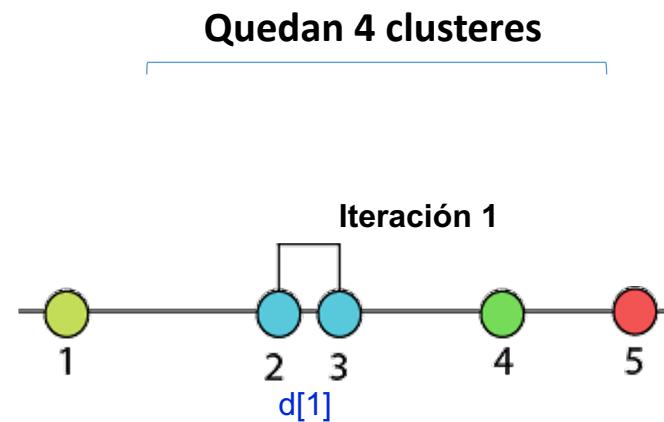
Algoritmo Ascendente Jerárquico

Procedimiento

3

En cada iteración se encuentra el par más cercano de elementos (clústeres) y se forma un único grupo.

- Menor distancia entre 2 objetos (grupos).



d	1	2	3	4	5
1	0	5	6	10	13
2		0	1	5	8
3			0	4	7
4				0	3
5					0

Algoritmo Ascendente Jerárquico

Procedimiento

4

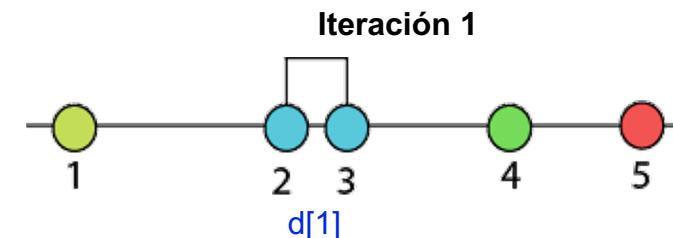
Se actualizan las distancias entre el nuevo grupo y los clústeres anteriores.

d	1	2	3	4	5
1	0	5	6	10	13
2		0	1	5	8
3			0	4	7
4				0	3
5					0

Iteración 1

Se promedian las nuevas
distancias

d	1	(2-3)	4	5
1	0	5.5	10	13
(2-3)		0	4.5	7.5
4			0	3
5				0



Algoritmo Ascendente Jerárquico

Procedimiento

5

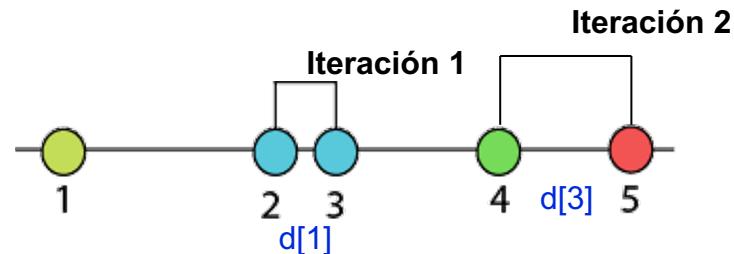
Se repiten los **pasos 3 y 4** hasta que todos los elementos se agrupen en un solo clúster.

d	1	(2-3)	4	5
1	0	5.5	10	13
(2-3)		0	4.5	7.5
4			0	3
5				0



Iteración 2
Se promedian las nuevas distancias

d	1	(2-3)	(4-5)
1	0	5.5	11.5
(2-3)		0	6
(4-5)			0



Algoritmo Ascendente Jerárquico

Procedimiento

5

Se repiten los **pasos 3 y 4** hasta que todos los elementos se agrupen en un solo cluster.

d	1	(2-3)	(4-5)
1	0	5.5	11.5
(2-3)		0	6
(4-5)			0

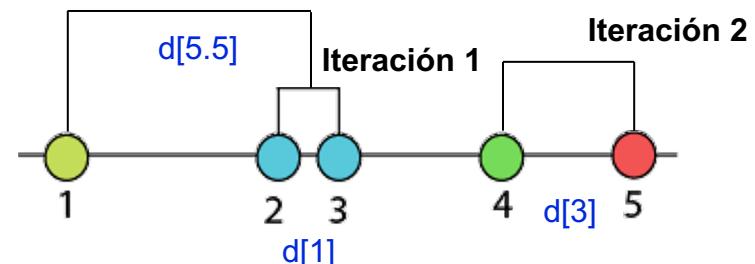
Iteración 3



Se promedian las nuevas
distancias

d	(1-2-3)	(4-5)
(1-2-3)	0	8.75
(4-5)		0

Iteración 3



Algoritmo Ascendente Jerárquico

Procedimiento

5

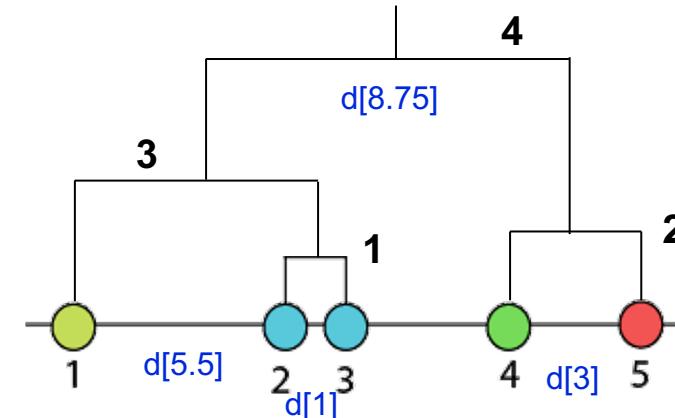
Se repiten los **pasos 3 y 4** hasta que todos los elementos se agrupen en un solo cluster.

d	(1-2-3)	(4-5)
(1-2-3)	0	8.75
(4-5)		0

Iteración 4



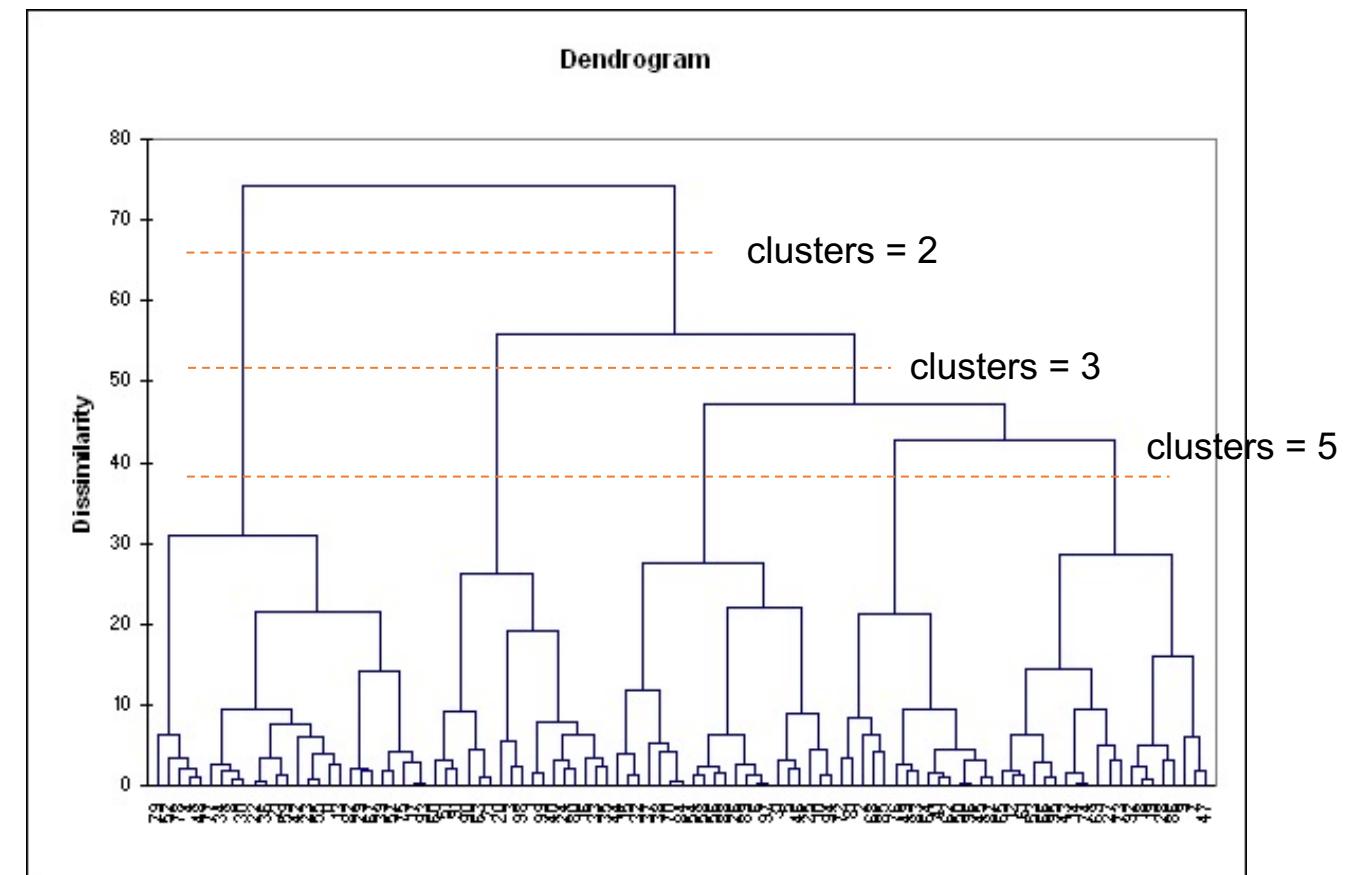
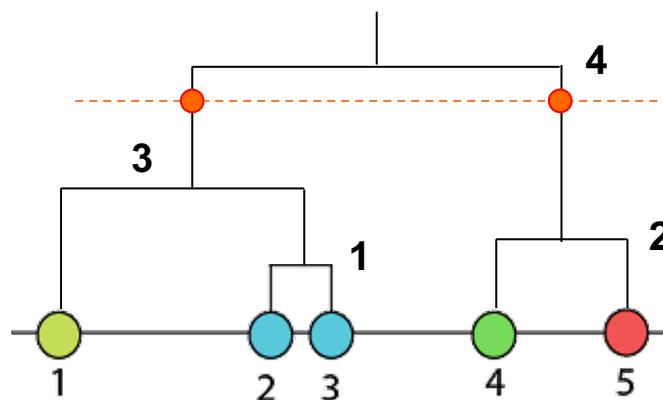
d	(1-2-3-4-5)
(1-2-3-4-5)	0



Algoritmo Ascendente Jerárquico

Método para decidir la cantidad de grupos

Se puede obtener el **número deseado** de clústeres "cortando" el árbol al nivel adecuado.



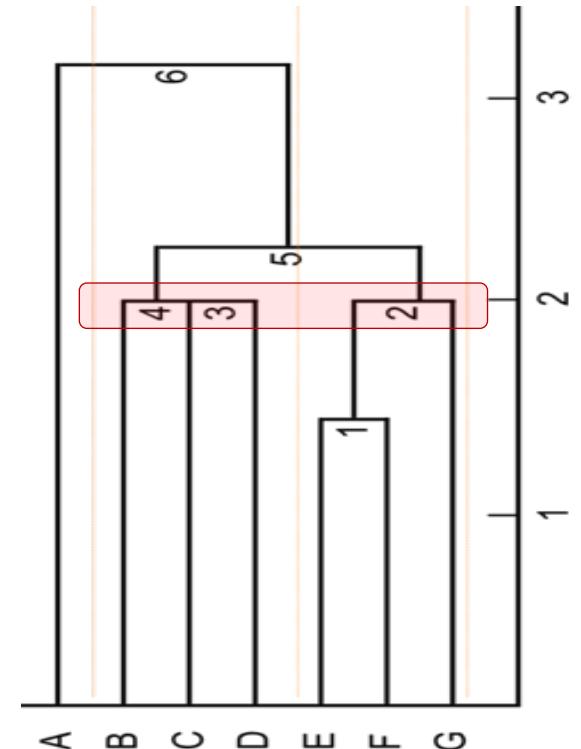
Algoritmo Ascendente Jerárquico

Distancias iguales

Sujetos	A	B	C	D	E	F	G
A	---						
B	3.16	---					
C	5.10	2.00	---				
D	5.10	2.83	2.00	---			
E	5.00	2.24	2.24	4.12	---		
F	6.40	3.61	3.00	5.00	1.41	---	
G	3.61	2.24	3.61	5.00	2.00	3.16	---



Paso	Distancia mínima entre sujetos	Sujetos
1	1.414	E-F
2	2.000	E-G
3	2.000	C-D
4	2.000	B-C
5	2.236	B-E
6	3.162	A-B
...		



Algoritmo Ascendente Jerárquico

Ejemplo ilustrativo

	ID	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	FaltasAño	Antiguedad	Sexo
1	E1	10000	1	0	0	0	0	7	15	1
2	E2	20000	0	1	1	0	1	3	3	0
3	E3	15000	1	1	2	1	1	5	10	1
4	E4	30000	1	1	1	0	0	15	7	0
5	E5	10000	1	1	0	1	1	1	6	1
6	E6	40000	0	1	0	0	1	3	16	0
7	E7	25000	0	0	0	0	1	0	8	1
8	E8	20000	0	1	0	1	1	2	6	0
9	E9	20000	1	1	3	1	0	7	5	1
10	E10	30000	1	1	2	1	0	1	20	1
11	E11	45000	0	0	0	0	0	2	12	0
12	E12	8000	1	1	2	1	0	3	1	1
13	E13	20000	0	0	0	0	0	27	5	0
14	E14	10000	0	1	0	0	1	0	7	1
15	E15	8000	0	1	0	0	0	3	2	1

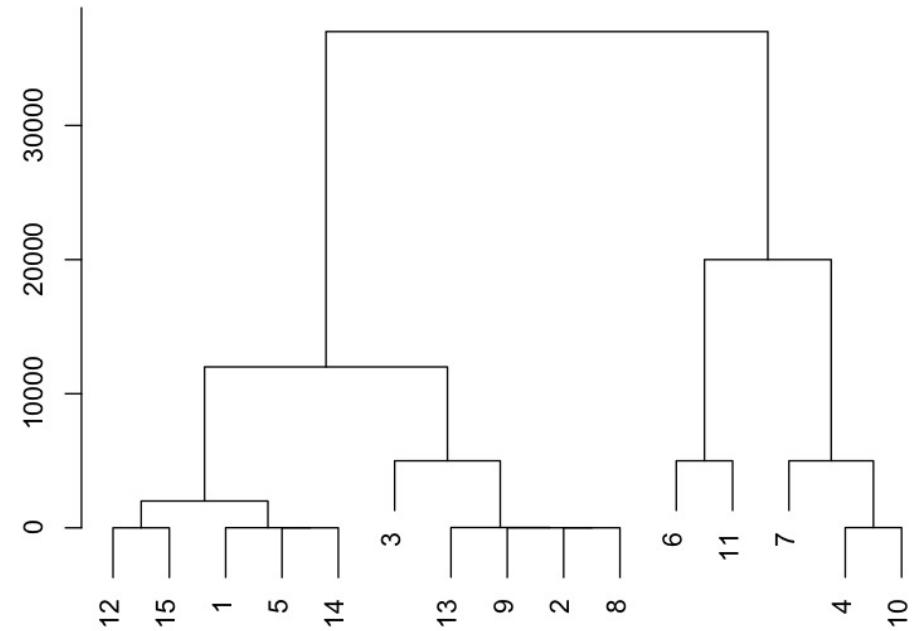
Algoritmo Ascendente Jerárquico

2

Obtención de grupos

	ID	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	FaltasAño	Antiguedad	Sexo
1	E1	10000	1	0	0	0	0	7	15	1
2	E2	20000	0	1	1	0	1	3	3	0
3	E3	15000	1	1	2	1	1	5	10	1
4	E4	30000	1	1	1	0	0	15	7	0
5	E5	10000	1	1	0	1	1	1	6	1
6	E6	40000	0	1	0	0	1	3	16	0
7	E7	25000	0	0	0	0	1	0	8	1
8	E8	20000	0	1	0	1	1	2	6	0
9	E9	20000	1	1	3	1	0	7	5	1
10	E10	30000	1	1	2	1	0	1	20	1
11	E11	45000	0	0	0	0	0	2	12	0
12	E12	8000	1	1	2	1	0	3	1	1
13	E13	20000	0	0	0	0	0	27	5	0
14	E14	10000	0	1	0	0	1	0	7	1
15	E15	8000	0	1	0	0	0	3	2	1

Cluster Dendrogram



Algoritmo Ascendente Jerárquico

3

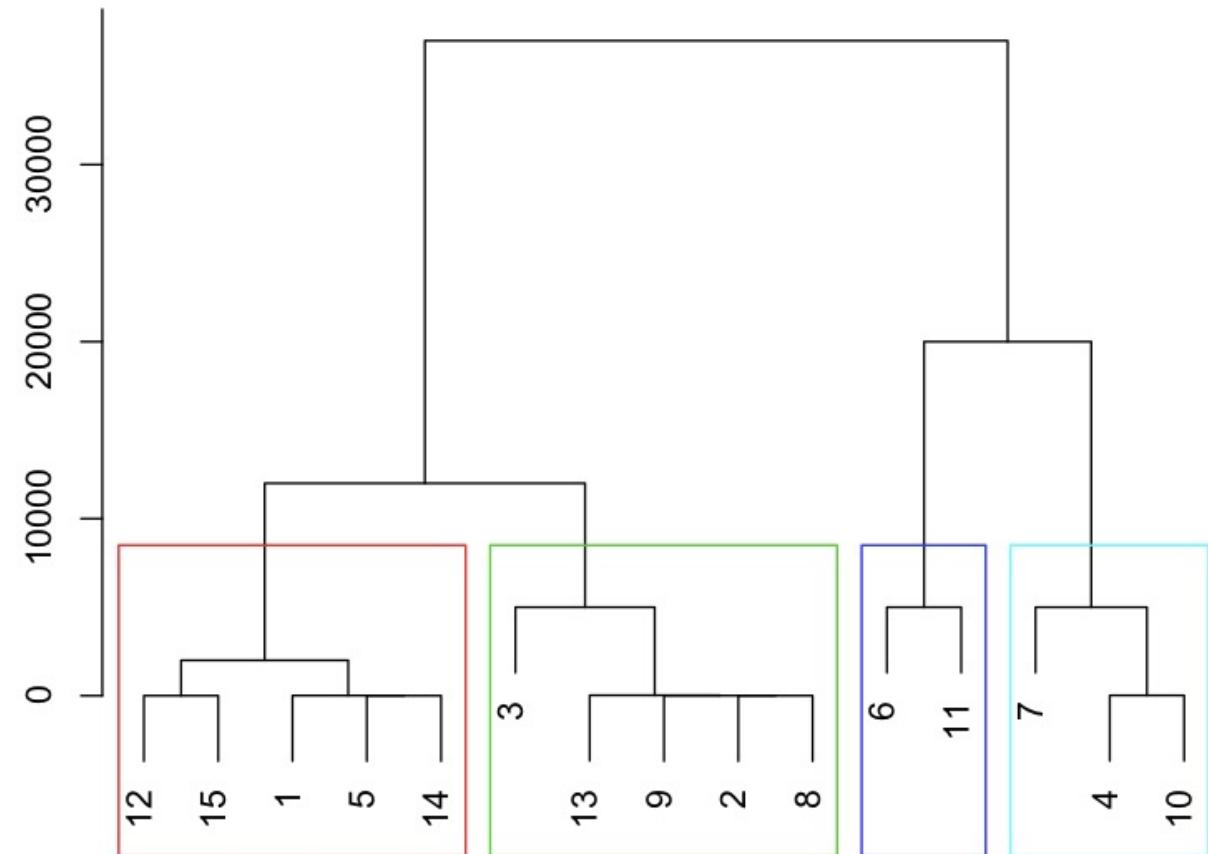
Definición de grupos

Asignación de grupos

[1] 1 2 2 3 1 4 3 2 2 3 4 1 2 1 1

Elementos por grupos

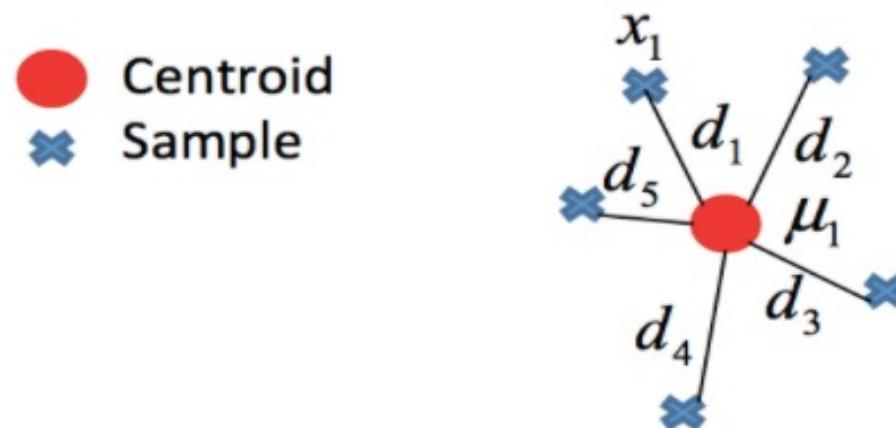
1 2 3 4
5 5 3 2



4 Interpretación

Cetroides:

- El centroide es el punto que ocupa la posición media en un cluster.
- La ubicación del centroide se calcula de manera iterativa.



Algoritmo Ascendente Jerárquico

4

Interpretación

Centroides	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	FaltasAño	Antiguedad	Sexo
[1,]	9200.00	0.6000000	0.8000000	0.4	0.4000000	0.4000000	2.800000	6.20000	1.0000000
[2,]	19000.00	0.4000000	0.8000000	1.2	0.6000000	0.6000000	8.800000	5.80000	0.4000000
[3,]	28333.33	0.6666667	0.6666667	1.0	0.3333333	0.3333333	5.333333	11.66667	0.6666667
[4,]	42500.00	0.0000000	0.5000000	0.0	0.0000000	0.5000000	2.500000	14.00000	0.0000000

Cluster Dendrogram

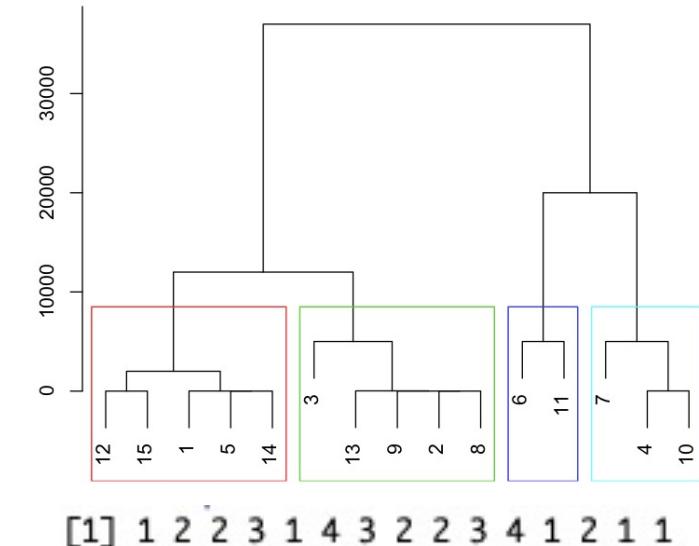
Clúster 1: 5 empleados

Salario : 9200
 Casado : Si = 0.6 / No = 0.4
 Coche : Si = 0.8 / No = 0.2
 Hijos : 0.4
 Vivienda : Prop = 0.4
 Alquiler = 0.6
 Sindicato : Si = 0.4 / No = 0.6
 Faltas/Año : 2.8 (3)
 Antigüedad : 6.2 (6)
 Sexo : M = 1

Clúster 2: 5 empleados

Salario : 19000
 Casado : Si = 0.4 / No = 0.6
 Coche : Si = 0.8 / No = 0.2
 Hijos : 1.2
 Vivienda : Prop = 0.6
 Alquiler = 0.4
 Sindicato : Si = 0.6 / No = 0.4
 Faltas/Año : 8.8 (9)
 Antigüedad : 5.8 (6)
 Sexo : M = 0.4 / F = 0.6

...



- **Clúster 1 [5 elementos –1, 5, 12, 14, 15–].** Empleados con salario promedio de \$ 9200, casados en su mayoría (60%), con coche (80%) y casi sin hijos (0.4). Solo el 40% tiene vivienda propia, no sindicalizados en su mayoría (60%), con algunas faltas al año (3), una antigüedad promedio de 6 años y todos varones (100%).

Algoritmo Ascendente Jerárquico

4

Interpretación

Centroides	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	FaltasAnno	Antiguedad	Sexo
[1,]	9200.00	0.6000000	0.8000000	0.4	0.4000000	0.4000000	2.800000	6.20000	1.0000000
[2,]	19000.00	0.4000000	0.8000000	1.2	0.6000000	0.6000000	8.800000	5.80000	0.4000000
[3,]	28333.33	0.6666667	0.6666667	1.0	0.3333333	0.3333333	5.333333	11.66667	0.6666667
[4,]	42500.00	0.0000000	0.5000000	0.0	0.0000000	0.5000000	2.500000	14.00000	0.0000000

Cluster Dendrogram

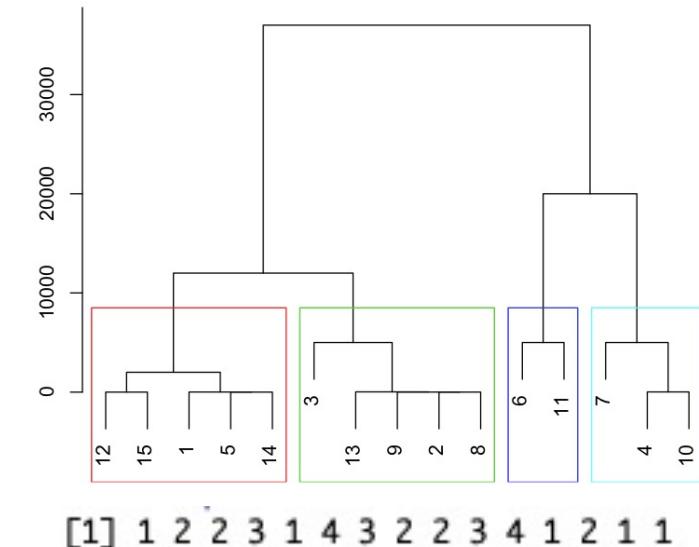
Clúster 3: 3 empleados

Salario : 28333
 Casado : Si = 0.67 / No = 0.33
 Coche : Si = 0.67 / No = 0.33
 Hijos : 1
 Vivienda : Prop = 0.33
 Alquiler = 0.67
 Sindicato : Si = 0.33 / No = 0.67
 Faltas/Año : 5.3 (5)
 Antigüedad : 11.6 (12)
 Sexo : M = 0.67 / F = 0.33

Clúster 4: 2 empleados

Salario : 42500
 Casado : No = 1
 Coche : Si = 0.5 / No = 0.5
 Hijos : 0
 Vivienda : Alquiler = 1
 Sindicato : Si = 0.5 / No = 0.5
 Faltas/Año : 2.5 (3)
 Antigüedad : 14
 Sexo : F = 1

...



- **Clúster 3 [5 elementos –4, 7, 10–].** Empleados con salario promedio de \$ 28333, casados en su mayoría (67%), con coche en su mayoría (67%) y con un hijo. No tienen vivienda propia en su mayoría (67%), no sindicalizados en su mayoría (67%), con varias faltas al año (5), con una antigüedad promedio de 12 años y la mayoría varones (67%).

Ensayo 2

Elaborar un breve ensayo, de dos hojas, sobre **Inteligencia**.

Fecha de entrega: jueves 28 de octubre de 2021

Hora: antes de las 11:00 horas

Formato: digital (LaTex), subir al la carpeta compartida los archivos 'pdf' y 'tex'.

Fuente:

La escalera de la Complejidad. Vida artificial II

Capítulo de libro: Inteligencia (Cap. 5)

<https://libros.univalle.edu.co/index.php/programaeditorial/catalog/book/151>



Universidad Nacional Autónoma de México
Facultad de Ingeniería

Clustering Jerárquico

Práctica 4

Guillermo Molero-Castillo

guillermo.molero@ingenieria.unam.edu

Octubre, 2021

Classical Machine Learning

Supervised Learning

(Pre Categorized Data)

Classification

(Divide the socks by Color)

Eg. Identity Fraud Detection

Regression

(Divide the Ties by Length)

Eg. Market Forecasting

Unsupervised Learning

(Unlabelled Data)

Clustering

(Divide by Similarity)

Eg. Targeted Marketing

Association

(Identify Sequences)

Eg. Customer Recommendation

Dimensionality Reduction

(Wider Dependencies)

Eg. Big Data Visualization

Task Driven

Data Driven

</div

Fuente de datos

- ingresos: son ingresos mensuales de 1 o 2 personas, si están casados.
- gastos_comunes: son gastos mensuales de 1 o 2 personas, si están casados.
- pago_coche
- gastos_otros
- ahorros
- vivienda: valor de la vivienda.
- estado_civil: 0-soltero, 1-casado, 2-divorciado
- hijos: cantidad de hijos menores (no trabajan).
- trabajo: 0-sin trabajo, 1-autonomo, 2-asalariado, 3-empresario, 4-autonomos, 5-asalariados, 6-autonomo y asalariado, 7-empresario y autonomo, 8-empresarios o empresario y autónomo
- comprar: 0-alquilar, 1-comprar casa a través de crédito hipotecario con tasa fija a 30 años.

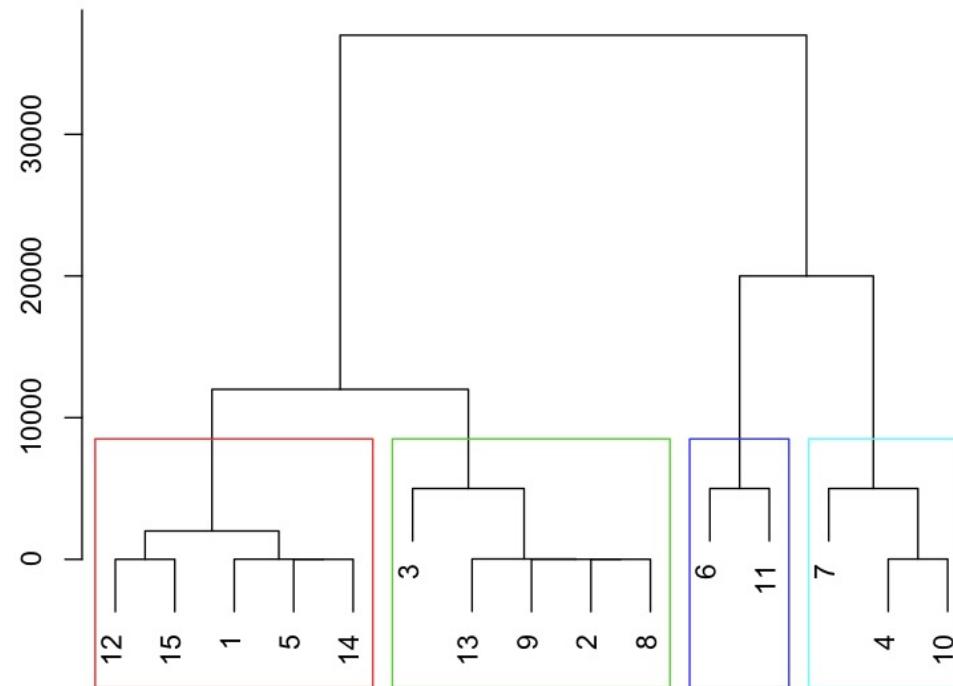
Práctica

Fuente de datos

ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo	comprar
6000	1000	0	600	50000	400000	0	2	2	1
6745	944	123	429	43240	636897	1	3	6	0
6455	1033	98	795	57463	321779	2	1	8	1
7098	1278	15	254	54506	660933	0	0	3	0
6167	863	223	520	41512	348932	0	0	3	1
5692	911	11	325	50875	360863	1	4	5	1

Objetivo

Obtener clústeres de casos de usuarios, con características similares, evaluados para la adquisición de una casa a través de un crédito hipotecario con tasa fija a 30 años.



1. Importar las bibliotecas y los datos

```
▶ import pandas as pd          # Para la manipulación y análisis de datos  
    import numpy as np          # Para crear vectores y matrices n dimensionales  
    import matplotlib.pyplot as plt # Para la generación de gráficas a partir de los datos  
    import seaborn as sns         # Para la visualización de datos basado en matplotlib  
    %matplotlib inline  
  
▶ from google.colab import files  
files.upload()  
  
#from google.colab import drive  
#drive.mount('/content/drive')  
  
□ Elegir archivos Hipoteca.csv  
• Hipoteca.csv(text/csv) - 8014 bytes, last modified: 1/4/2021 - 100% done  
Saving Hipoteca.csv to Hipoteca.csv  
{'Hipoteca.csv': b'ingresos,gastos_comunes,pago_coche,gastos_otros,ahorros,vivienda,estado_civil'}
```

Práctica

1. Importar las bibliotecas y los datos



```
Hipoteca = pd.read_csv("Hipoteca.csv")  
Hipoteca
```

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo	comprar
0	6000	1000	0	600	50000	400000	0	2	2	1
1	6745	944	123	429	43240	636897	1	3	6	0
2	6455	1033	98	795	57463	321779	2	1	8	1
3	7098	1278	15	254	54506	660933	0	0	3	0
4	6167	863	223	520	41512	348932	0	0	3	1
...
197	3831	690	352	488	10723	363120	0	0	2	0
198	3961	1030	270							
199	3184	955	276							
200	3334	867	369							
201	3988	1157	105							

202 rows × 10 columns



```
#'comprar' representa un valor obtenido de un análisis hipotecario preliminar  
print(Hipoteca.groupby('comprar').size())
```

comprar	size
0	135
1	67

dtype: int64

1. Importar las bibliotecas y los datos



Hipoteca.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 202 entries, 0 to 201
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   ingresos         202 non-null    int64  
 1   gastos_comunes  202 non-null    int64  
 2   pago_coche       202 non-null    int64  
 3   gastos_otros    202 non-null    int64  
 4   ahorros          202 non-null    int64  
 5   vivienda         202 non-null    int64  
 6   estado_civil    202 non-null    int64  
 7   hijos            202 non-null    int64  
 8   trabajo          202 non-null    int64  
 9   comprar          202 non-null    int64  
dtypes: int64(10)
memory usage: 15.9 KB
```



print(Hipoteca.groupby('comprar').size())

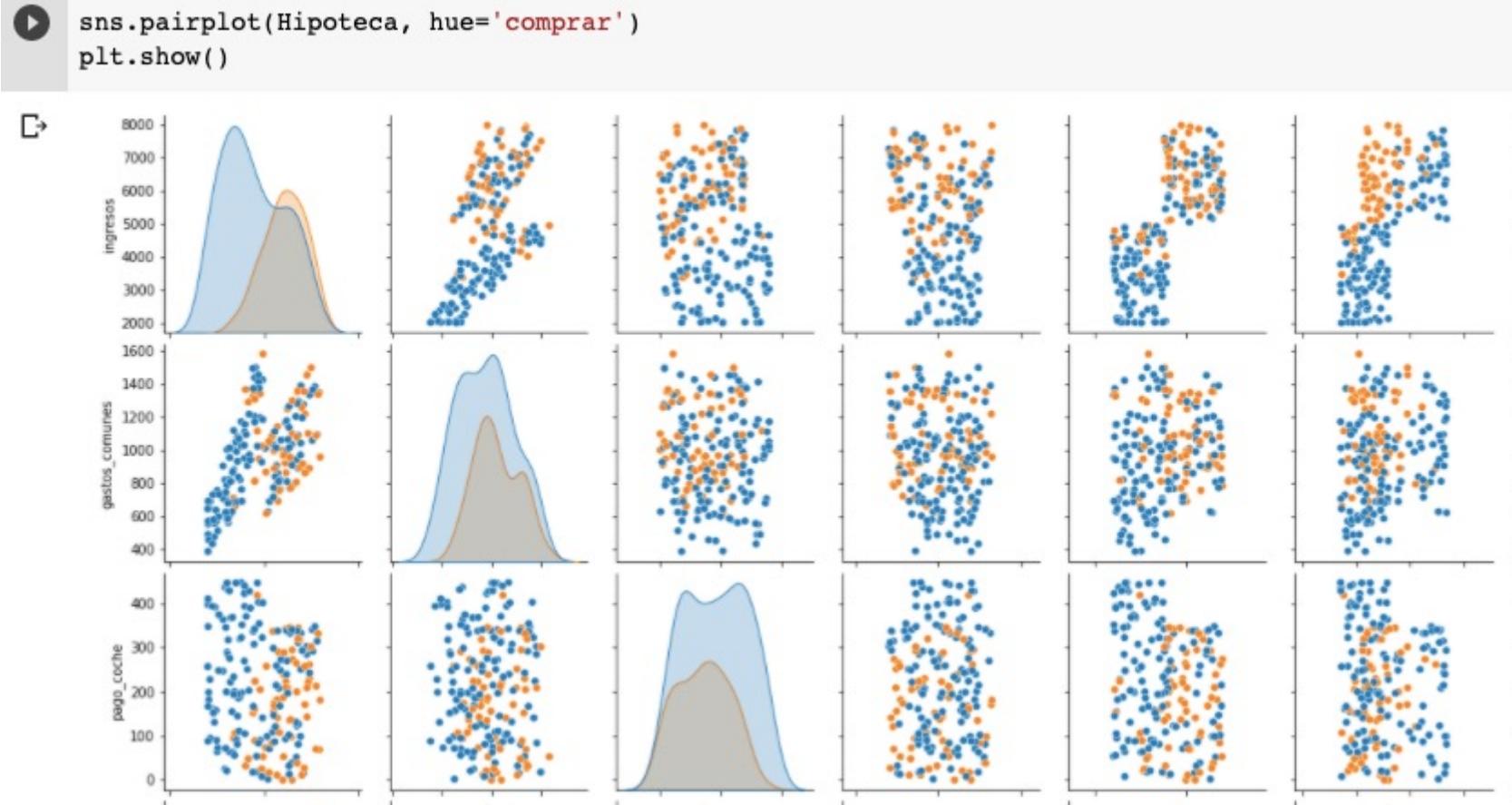
comprar

0	135
1	67

dtype: int64

2. Selección de características

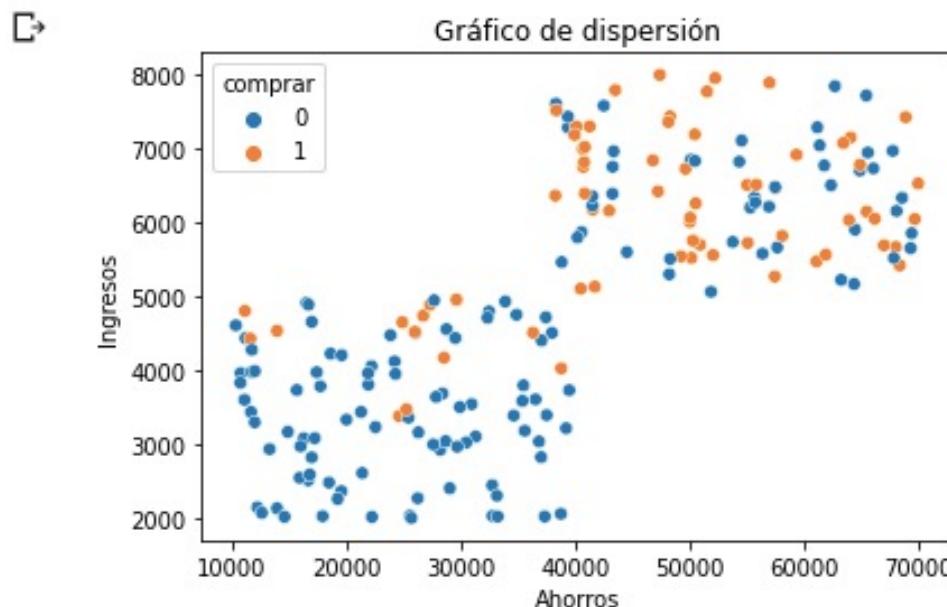
Evaluación visual



2. Selección de características

Evaluación visual

```
▶ sns.scatterplot(x='ahorros', y ='ingresos', data=Hipoteca, hue='comprar')
plt.title('Gráfico de dispersión')
plt.xlabel('Ahorros')
plt.ylabel('Ingresos')
plt.show()
```



2. Selección de características

Matriz de correlaciones

```
CorrHipoteca = Hipoteca.corr(method='pearson')
CorrHipoteca
```

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo	comprar
ingresos	1.000000	0.560211	-0.109780	-0.124105	0.712889	0.614721	-0.042556	-0.024483	-0.038852	0.467123
gastos_comunes	0.560211	1.000000	-0.054400	-0.099881	0.209414	0.204781	-0.057152	-0.072321	-0.079095	0.200191
pago_coche	-0.109780	-0.054400	1.000000	0.010602	-0.193299	-0.094631	0.052239	-0.044858	0.018946	-0.196468
gastos_otros	-0.124105	-0.099881	0.010602	1.000000	-0.064384	-0.054577	-0.020226	0.124845	0.047313	-0.110330
ahorros	0.712889	0.209414	-0.193299	-0.064384	1.000000	0.605836	-0.063039	0.001445	-0.023829	0.340778
vivienda	0.614721	0.204781	-0.094631	-0.054577	0.605836	1.000000	-0.113420	-0.141924	-0.211790	-0.146092
estado_civil	-0.042556	-0.057152	0.052239	-0.020226	-0.063039	-0.113420	1.000000	0.507609	0.589512	0.142799
hijos	-0.024483	-0.072321	-0.044858	0.124845	0.001445	-0.141924	0.507609	1.000000	0.699916	0.272883
trabajo	-0.038852	-0.079095	0.018946	0.047313	-0.023829	-0.211790	0.589512	0.699916	1.000000	0.341537
comprar	0.467123	0.200191	-0.196468	-0.110330	0.340778	-0.146092	0.142799	0.272883	0.341537	1.000000

2. Selección de características

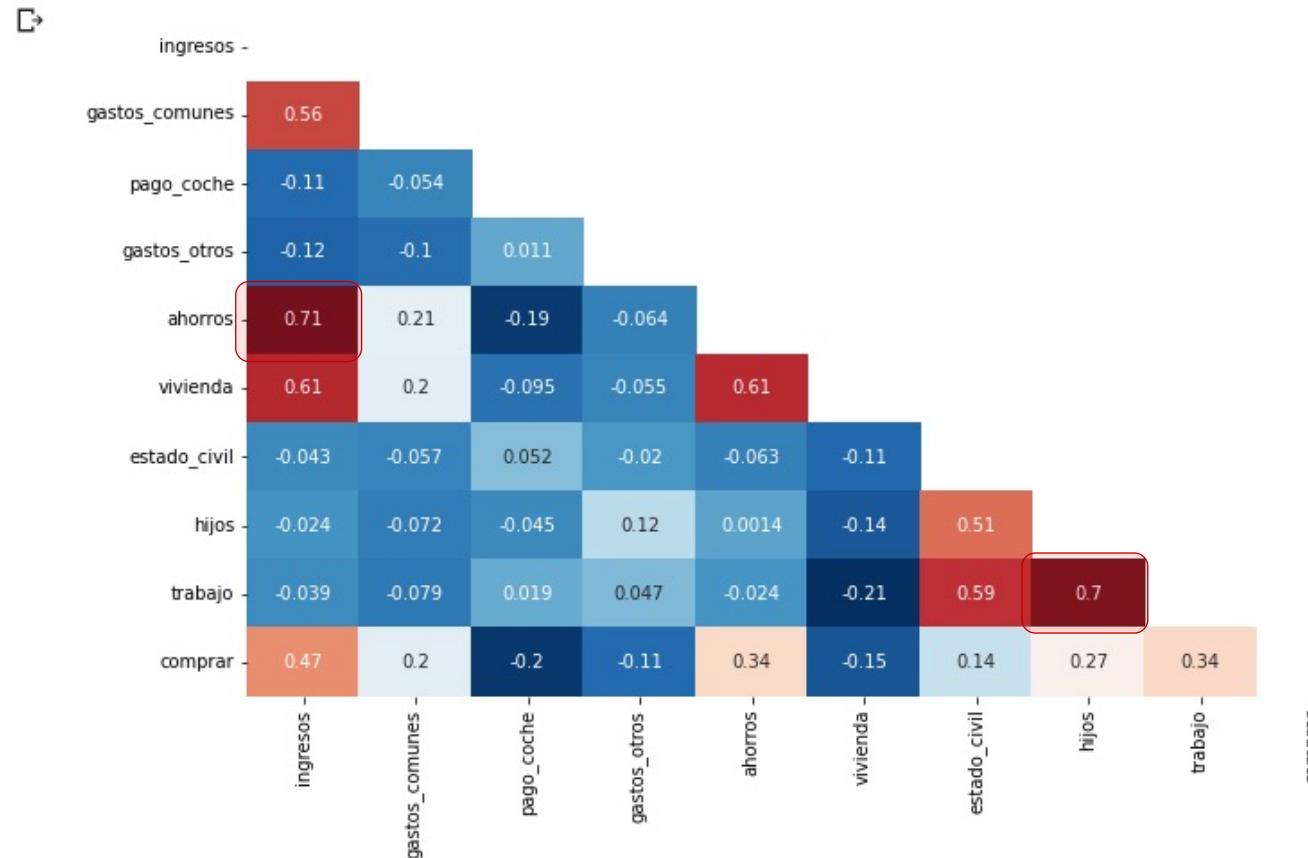
Matriz de correlaciones

```
▶ print(CorrHipoteca['ingresos'].sort_values(ascending=False)[:10], '\n') #Top 10 valores
```

	ingresos	ahorros	vivienda	gastos_comunes	comprar	hijos	trabajo	estado_civil	pago_coche	gastos_otros
	1.000000	0.712889	0.614721	0.560211	0.467123	-0.024483	-0.038852	-0.042556	-0.109780	-0.124105
Name:	ingresos									
dtype:	float64									

2. Selección de características

```
plt.figure(figsize=(14,7))
MatrizInf = np.triu(CorrHipoteca)
sns.heatmap(CorrHipoteca, cmap='RdBu_r', annot=True, mask=MatrizInf)
plt.show()
```



Selección de variables:

- A pesar de existir 2 correlaciones altas, entre 'ingresos' y 'ahorros' (0.71) y 'trabajo' e 'hijos' (0.69); éstas se tomarán en cuenta para obtener una segmentación que combine todas las variables.
- Se suprimirá la variable 'comprar' debido a que representa inherentemente un agrupamiento, y fue un campo calculado con base a un análisis hipotecario preliminar.



2. Selección de características

Elección de variables

```
▶ MatrizHipoteca = np.array(Hipoteca[['ingresos', 'gastos_comunes', ' pago_coche', 'gastos_otros', 'ahorros', 'vivienda',  
pd.DataFrame(MatrizHipoteca)  
#MatrizHipoteca = Hipoteca.iloc[:, 0:9].values      #iloc para seleccionar filas y columnas según su posición
```

	0	1	2	3	4	5	6	7	8
0	6000	1000	0	600	50000	400000	0	2	2
1	6745	944	123	429	43240	636897	1	3	6
2	6455	1033	98	795	57463	321779	2	1	8
3	7098	1278	15	254	54506	660933	0	0	3
4	6167	863	223	520	41512	348932	0	0	3
...

3) Aplicación del algoritmo: Ascendente Jerárquico

Estandarización de datos

```
▶ from sklearn.preprocessing import StandardScaler, MinMaxScaler  
estandarizar = StandardScaler() # Se instancia el objeto StandardScaler o MinMaxScaler  
MEstandarizada = estandarizar.fit_transform(MatrizHipoteca) # Se calculan la media y desviación y se escalan los datos  
  
▶ pd.DataFrame(MEstandarizada)  
  
→

|     | 0        | 1         | 2         | 3         | 4        | 5         | 6         | 7         | 8         |
|-----|----------|-----------|-----------|-----------|----------|-----------|-----------|-----------|-----------|
| 0   | 0.620129 | 0.104689  | -1.698954 | 0.504359  | 0.649475 | 0.195910  | -1.227088 | 0.562374  | -0.984420 |
| 1   | 1.063927 | -0.101625 | -0.712042 | -0.515401 | 0.259224 | 1.937370  | -0.029640 | 1.295273  | 0.596915  |
| 2   | 0.891173 | 0.226266  | -0.912634 | 1.667244  | 1.080309 | -0.379102 | 1.167809  | -0.170526 | 1.387582  |
| 3   | 1.274209 | 1.128886  | -1.578599 | -1.559015 | 0.909604 | 2.114062  | -1.227088 | -0.903426 | -0.589086 |
| 4   | 0.719611 | -0.400042 | 0.090326  | 0.027279  | 0.159468 | -0.179497 | -1.227088 | -0.903426 | -0.589086 |
| ... | ...      | ...       | ...       | ...       | ...      | ...       | ...       | ...       | ...       |


```

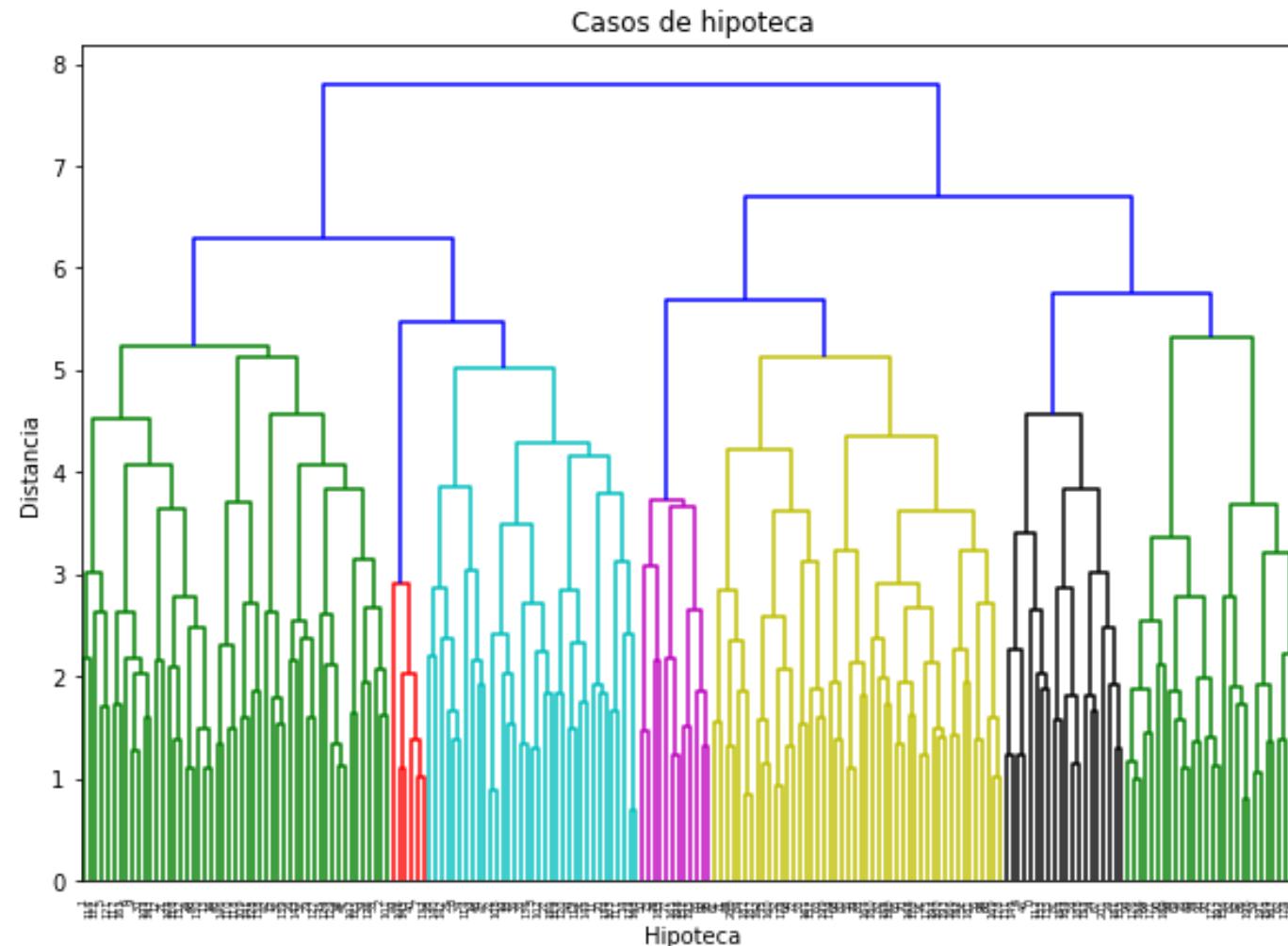
3) Aplicación del algoritmo: Ascendente Jerárquico

Creación del árbol

```
#Se importan las bibliotecas de clustering jerárquico para crear el árbol
import scipy.cluster.hierarchy as shc
from sklearn.cluster import AgglomerativeClustering
plt.figure(figsize=(10, 7))
plt.title("Casos de hipoteca")
plt.xlabel('Hipotecas')
plt.ylabel('Distancia')
Arbol = shc.dendrogram(shc.linkage(MEstandarizada, method='complete', metric='euclidean'))
#plt.axhline(y=5.4, color='orange', linestyle='--')
#Probar con otras mediciones de distancia (euclidean, chebyshev, cityblock)
```

3) Aplicación del algoritmo: Ascendente Jerárquico

Creación del árbol



3) Aplicación del algoritmo: Ascendente Jerárquico

Se crean las etiquetas en los clústeres

```
#Se crean las etiquetas de los elementos en los clústeres
MJerarquico = AgglomerativeClustering(n_clusters=7, linkage='complete', affinity='euclidean')
MJerarquico.fit_predict(MEstandarizada)
MJerarquico.labels_

array([4, 1, 1, 2, 4, 1, 1, 6, 2, 1, 2, 2, 1, 1, 2, 1, 1, 1, 2, 2, 2, 1,
       2, 1, 4, 2, 1, 2, 2, 1, 1, 2, 1, 2, 2, 1, 2, 1, 1, 1, 1, 6, 1, 1, 1,
       2, 2, 4, 2, 1, 6, 5, 3, 3, 3, 4, 3, 3, 0, 4, 0, 3, 3, 0, 3, 0, 3, 0, 3,
       3, 4, 3, 0, 3, 3, 3, 5, 0, 3, 0, 5, 5, 3, 3, 4, 0, 3, 3, 5, 0, 3,
       3, 0, 0, 3, 5, 0, 0, 5, 0, 0, 3, 0, 3, 1, 2, 1, 1, 2, 6, 1, 2, 1,
       1, 2, 4, 2, 2, 1, 1, 1, 2, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 4,
       6, 4, 2, 4, 2, 1, 1, 1, 2, 1, 2, 1, 2, 6, 1, 1, 2, 4, 2, 4, 5, 4,
       4, 4, 0, 3, 3, 0, 3, 3, 1, 3, 5, 3, 0, 3, 3, 3, 0, 0, 3, 0, 3,
       0, 0, 3, 3, 3, 3, 4, 5, 0, 3, 4, 0, 3, 0, 0, 3, 3, 5, 0, 0,
       5, 3, 3, 4])
```

3) Aplicación del algoritmo: Ascendente Jerárquico

Se crean las etiquetas en los clústeres

```
▶ Hipoteca = Hipoteca.drop(columns=['comprar'])
Hipoteca['clusterH'] = MJerarquico.labels_
Hipoteca
```

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo	clusterH
0	6000	1000	0	600	50000	400000	0	2	2	4
1	6745	944	123	429	43240	636897	1	3	6	1
2	6455	1033	98	795	57463	321779	2	1	8	1
3	7098	1278	15	254	54506	660933	0	0	3	2
4	6167	863	223	520	41512	348932	0	0	3	4
...
197	3831									0
198	3961									5
199	3184									3
200	3334									3
201	3988									4
202 rows × 10 columns										

▶ #Cantidad de elementos en los clusters
 Hipoteca.groupby(['clusterH'])['clusterH'].count()

clusterH	0	1	2	3	4	5	6
0	30	51	35	48	20	12	6
1							
2							
3							
4							
5							
6							

Name: clusterH, dtype: int64

3) Aplicación del algoritmo: Ascendente Jerárquico

Se crean las etiquetas en los clústeres

▶ Hipoteca[Hipoteca.clusterH == 6]

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo	clusterH	
7	6470	1035	39	782	57439	606291		0	0	1	6
40	6822	1296	81	786	50433	669054		0	0	0	6
49	6959	1392	333	818	67714	571076		0	0	3	6
106	6205	1179	240	729	56904	661009		0	0	2	6
132	6325	1139	102	754	68527	588004		0	0	0	6
145	5646	1016	215	747	69276	655399		0	0	1	6

3) Aplicación del algoritmo: Ascendente Jerárquico

Obtención de los centroides



```
CentroidesH = Hipoteca.groupby('clusterH').mean()  
CentroidesH
```



clusterH	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo
0	3421.133333	846.466667	309.933333	527.233333	24289.633333	295590.700000	0.233333	0.000000	2.000000
1	6394.019608	1021.627451	192.274510	533.039216	54382.529412	421178.764706	1.490196	2.254902	6.313725
2	6599.542857	1087.428571	204.771429	362.600000	51863.028571	515494.257143	0.685714	0.228571	2.885714
3	3189.687500	785.020833	243.208333	548.270833	23616.854167	277066.687500	1.645833	1.979167	6.208333
4	4843.750000	1009.200000	122.200000	572.850000	36340.650000	337164.850000	0.050000	0.100000	1.900000
5	4466.416667	1315.083333	114.416667	502.750000	23276.166667	269429.916667	1.666667	2.416667	6.750000
6	6404.500000	1176.166667	168.333333	769.333333	61715.500000	625138.833333	0.000000	0.000000	1.166667

Práctica

4) Interpretación

clusterH	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo
0	3421.133333	846.466667	309.933333	527.233333	24289.633333	295590.700000	0.233333	0.000000	2.000000
1	6394.019608	1021.627451	192.274510	533.039216	54382.529412	421178.764706	1.490196	2.254902	6.313725
2	6599.542857	1087.428571	204.771429	362.600000	51863.028571	515494.257143	0.685714	0.228571	2.885714
3	3189.687500	785.020833	243.208333	548.270833	23616.854167	277066.687500	1.645833	1.979167	6.208333
4	4843.750000	1009.200000	122.200000	572.850000	36340.650000	337164.850000	0.050000	0.100000	1.900000
5	4466.416667	1315.083333	114.416667	502.750000	23276.166667	269429.916667	1.666667	2.416667	6.750000
6	6404.500000	1176.166667	168.333333	769.333333	61715.500000	625138.833333	0.000000	0.000000	1.166667

Cluster 0: Conformado por 30 casos de una evaluación hipotecaria, con un ingreso promedio mensual de 3421 USD, con gastos comunes de 846 USD, otros gastos de 527 USD y un pago mensual de coche de 309 USD. Estos gastos en promedio representan casi la mitad del salario mensual (1682 USD). Por otro lado, este grupo de usuarios tienen un ahorro promedio de 24289 USD, y un valor promedio de vivienda (a comprar o hipotecar) de 295590 USD. Además, en su mayoría son solteros (0-soltero), sin hijos menores y tienen un tipo de trabajo asalariado (2-asalariado).

...

```
clusterH
0    30
1    51
2    35
3    48
4    20
5    12
6     6
Name: clusterH, dtype: int64
```

4) Interpretación

clusterH	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo
0	3421.133333	846.466667	309.933333	527.233333	24289.633333	295590.700000	0.233333	0.000000	2.000000
1	6394.019608	1021.627451	192.274510	533.039216	54382.529412	421178.764706	1.490196	2.254902	6.313725
2	6599.542857	1087.428571	204.771429	362.600000	51863.028571	515494.257143	0.685714	0.228571	2.885714
3	3189.687500	785.020833	243.208333	548.270833	23616.854167	277066.687500	1.645833	1.979167	6.208333
4	4843.750000	1009.200000	122.200000	572.850000	36340.650000	337164.850000	0.050000	0.100000	1.900000
5	4466.416667	1315.083333	114.416667	502.750000	23276.166667	269429.916667	1.666667	2.416667	6.750000
6	6404.500000	1176.166667	168.333333	769.333333	61715.500000	625138.833333	0.000000	0.000000	1.166667

Cluster 6: Es un segmento de clientes conformado por solo 6 usuarios, con un ingreso promedio mensual de 6404 USD, con gastos comunes de 1176 USD, otros gastos de 769 USD y un pago mensual de coche de 168 USD. Estos gastos en promedio representan casi una tercera parte del salario mensual (2113 USD). Por otro lado, este grupo de usuarios tienen un ahorro promedio de 61715 USD, y un valor promedio de vivienda (a comprar o hipotecar) de 625138 USD. Además, todos son solteros (0-soltero), sin hijos y tienen un tipo de trabajo en su mayoría autónomos (1-autónomo).

```

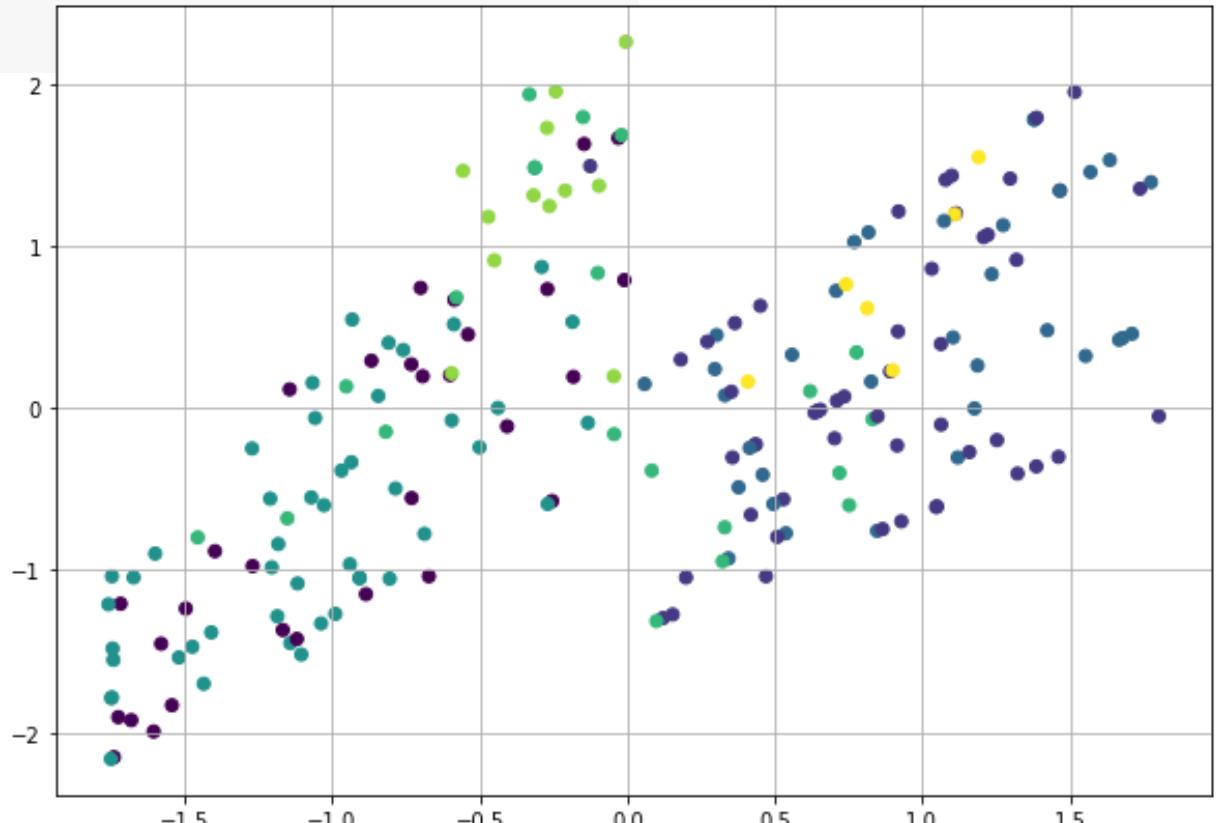
clusterH
0    30
1    51
2    35
3    48
4    20
5    12
6     6
Name: clusterH, dtype: int64

```

4) Interpretación



```
plt.figure(figsize=(10, 7))
plt.scatter(MEstandarizada[:,0], MEstandarizada[:,1], c=MJerarquico.labels_)
plt.grid()
plt.show()
```





Universidad Nacional Autónoma de México
Facultad de Ingeniería

Aprendizaje no supervisado Clustering Particional

Guillermo Molero-Castillo

guillermo.molero@ingenieria.unam.edu

Octubre, 2021

Machine Learning

Classical Machine Learning

Task Driven

Supervised Learning

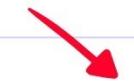
(Pre Categorized Data)



Classification

(Divide the socks by Color)

Eg. Identity Fraud Detection



Regression

(Divide the Ties by Length)

Eg. Market Forecasting

Data Driven

Unsupervised Learning

(Unlabelled Data)

Clustering

(Divide by Similarity)

Eg. Targeted Marketing

Association

(Identify Sequences)

Eg. Customer Recommendation

Dimensionality Reduction

(Wider Dependencies)

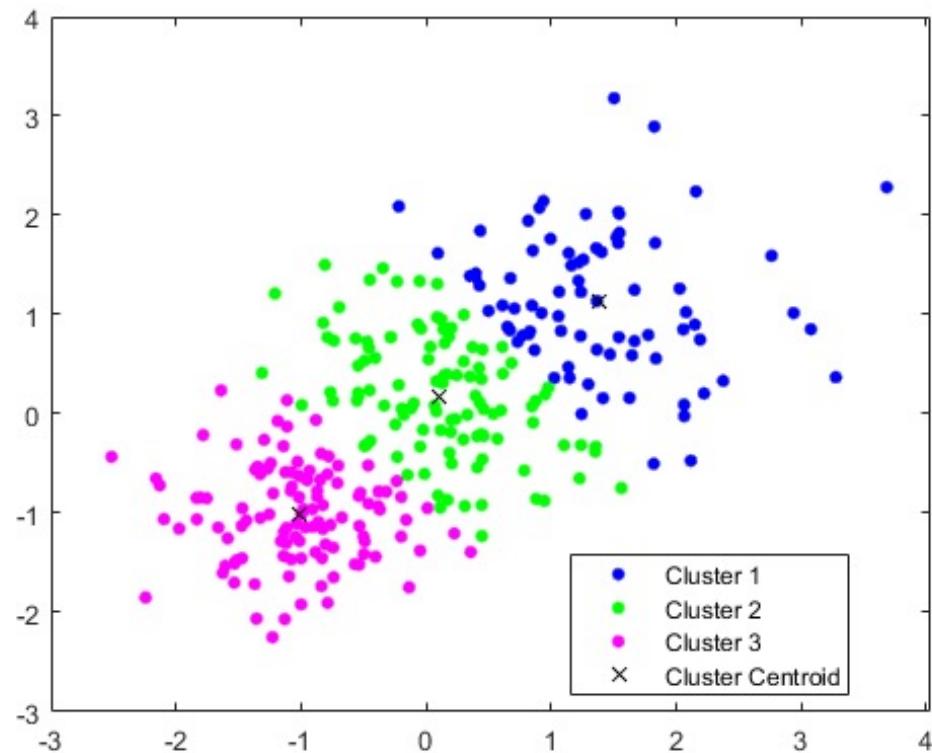
Eg. Big Data Visualization



Clustering Particional

El **algoritmo particional**, conocido también como de particiones, organiza los elementos dentro de k clústeres. Tiene ventajas en aplicaciones que involucran gran cantidad de datos.

Particional



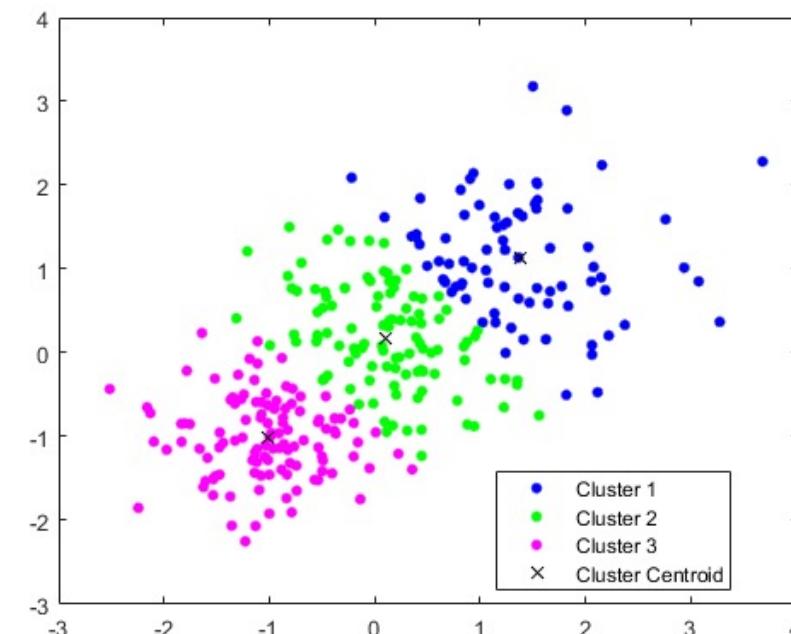
Clustering Particional

K-means

- Es uno de los algoritmos utilizados en la industria para crear **k** clústeres a partir de un conjunto de elementos (objetos), de modo que los miembros de un grupo sean similares.
- Ejemplo: Analizar pacientes por su situación de salud: edad, pulso, presión arterial, colesterol, entre otros.

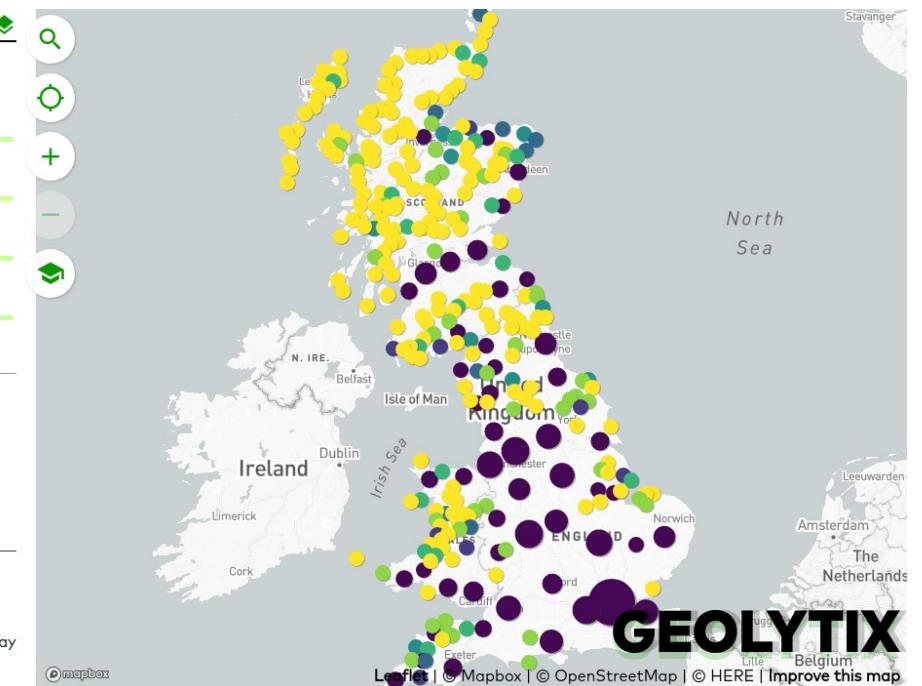
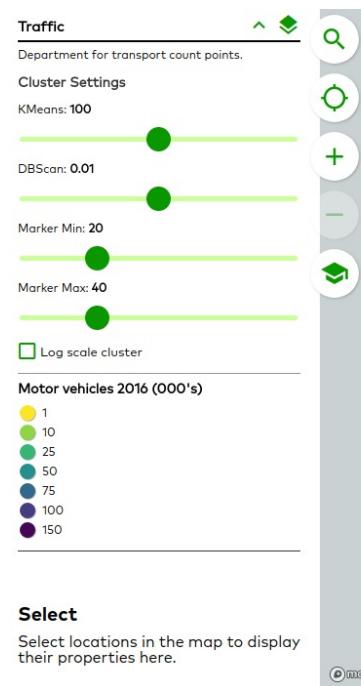
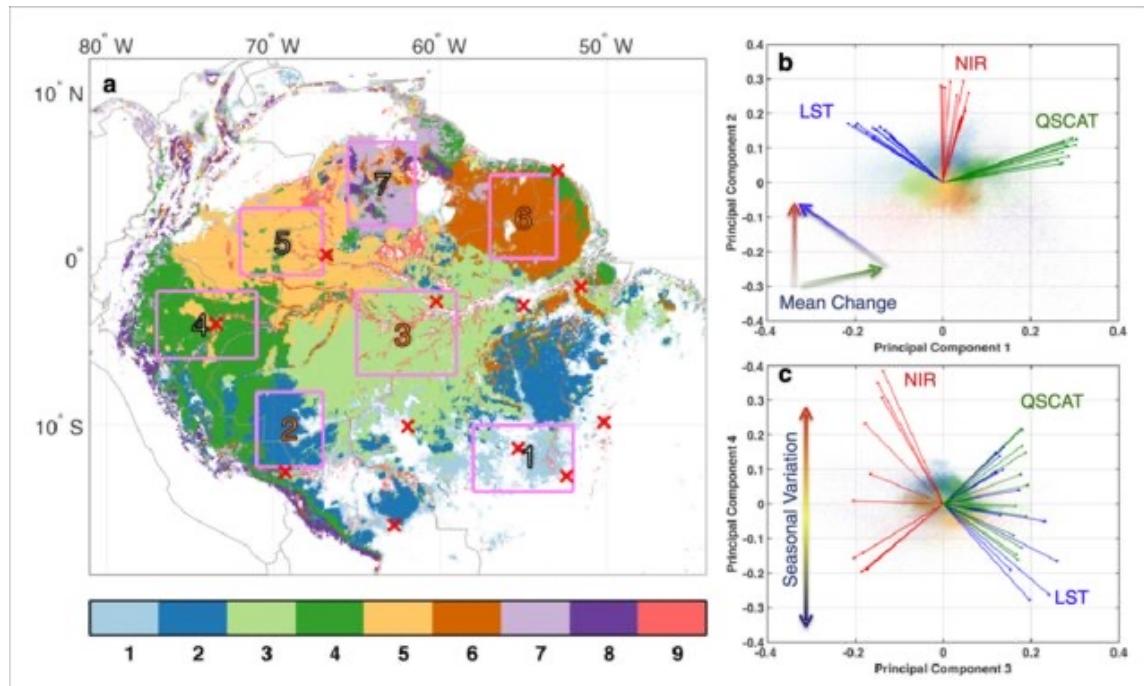
Pulso
Presión arterial
Colesterol
...

Estas mediciones sobre el paciente representan un vector de datos.



Clustering Particional

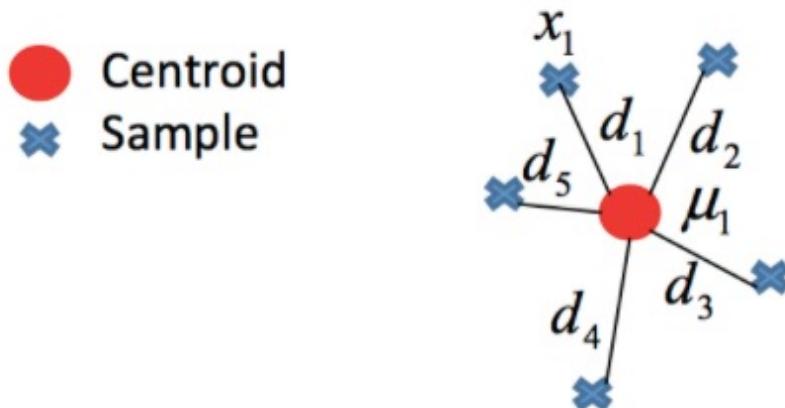
El algoritmo k-means resuelve **problemas de optimización**, dado que la función es minimizar (optimizar) la suma de las distancias de cada elemento al centroide de un clúster.



Clustering Particional

Centroide:

- El centroide es el punto que ocupa la posición media en un cluster.
- Al inicio, cuando se empieza a definir el cluster, es probable que el centroide no tenga relación con algunos de los elementos.
- Posteriormente, la ubicación del centroide se calcula de manera iterativa.



Clustering Particional

Pseudocódigo

- 1 **Inicio:** Se establecen k centroides para la formación de k grupos. Estos centroides (elementos) se eligen aleatoriamente.
- 2 **Asignación:** Cada elemento es asignado al centroide más cercano.
- 3 **Actualización:** Se actualiza la posición del centroide con base en la media de los elementos asignados en el cluster.
- 4 **Repetir:** Se repiten los pasos 2 y 3 de manera iterativa hasta que los centroides no cambien más.

Cuando se trabaja con **clustering**, dado que son algoritmos basados en distancias, es fundamental estandarizar los datos para que cada una de las variables contribuyan por igual en el análisis.

$$z = \frac{\text{value} - \text{mean}}{\text{standard deviation}}$$

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

La razón es que si existen diferencias entre los rangos de las variables, aquellas con valores más grandes predominarán sobre las que tienen rangos pequeños. Por ejemplo, un rango entre [0 y 100] dominará sobre otras que oscilan entre [0 y 1]. Esto dará lugar a resultados sesgados.

Clustering Particional

Pseudocódigo

- 1 **Inicio:** Se establecen k centroides para la formación de k grupos. Estos centroides (elementos) se eligen aleatoriamente.
- 2 **Asignación:** Cada elemento es asignado al centroide más cercano.
- 3 **Actualización:** Se actualiza la posición del centroide con base en la media de los elementos asignados en el cluster.
- 4 **Repetir:** Se repiten los pasos 2 y 3 de manera iterativa hasta que los centroides no cambien más.

K-MEANS(P, k)

Input: a dataset of points $P = \{p_1, \dots, p_n\}$, a number of clusters k
Output: centers $\{c_1, \dots, c_k\}$ implicitly dividing P into k clusters

```
1 choose  $k$  initial centers  $C = \{c_1, \dots, c_k\}$ 
2 while stopping criterion has not been met
3   do ▷ assignment step:
4     for  $i = 1, \dots, N$ 
5       do find closest center  $c_k \in C$  to instance  $p_i$ 
6         assign instance  $p_i$  to set  $C_k$ 
7   ▷ update step:
8   for  $i = 1, \dots, k$ 
9     do set  $c_i$  to be the center of mass of all points in  $C_i$ 
```

Clustering Particional

Para la asignación: Se asigna cada elemento al clúster más cercano, aplicando alguna medida de distancia (por ejemplo, Euclidiana, Manhattan, Chebyshev, o Minkowsky) entre el objeto y el centroide del clúster.

$$dist(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

$$d_{Cheb}(p, q) = \max |p_i - q_i|$$

$$d_{Manh}(p, q) = \sum_{i=1}^n |p_i - q_i|$$

$$d_{Mink}(q, p) = \lambda \sqrt{\sum_{i=1}^n (q_i - p_i)^\lambda}$$

Para la actualización: Se calcula los nuevos centroides con base en la media de los elementos asignados en el clúster.

$$\mu = \frac{1}{N} \sum_{j=1}^N x_j$$

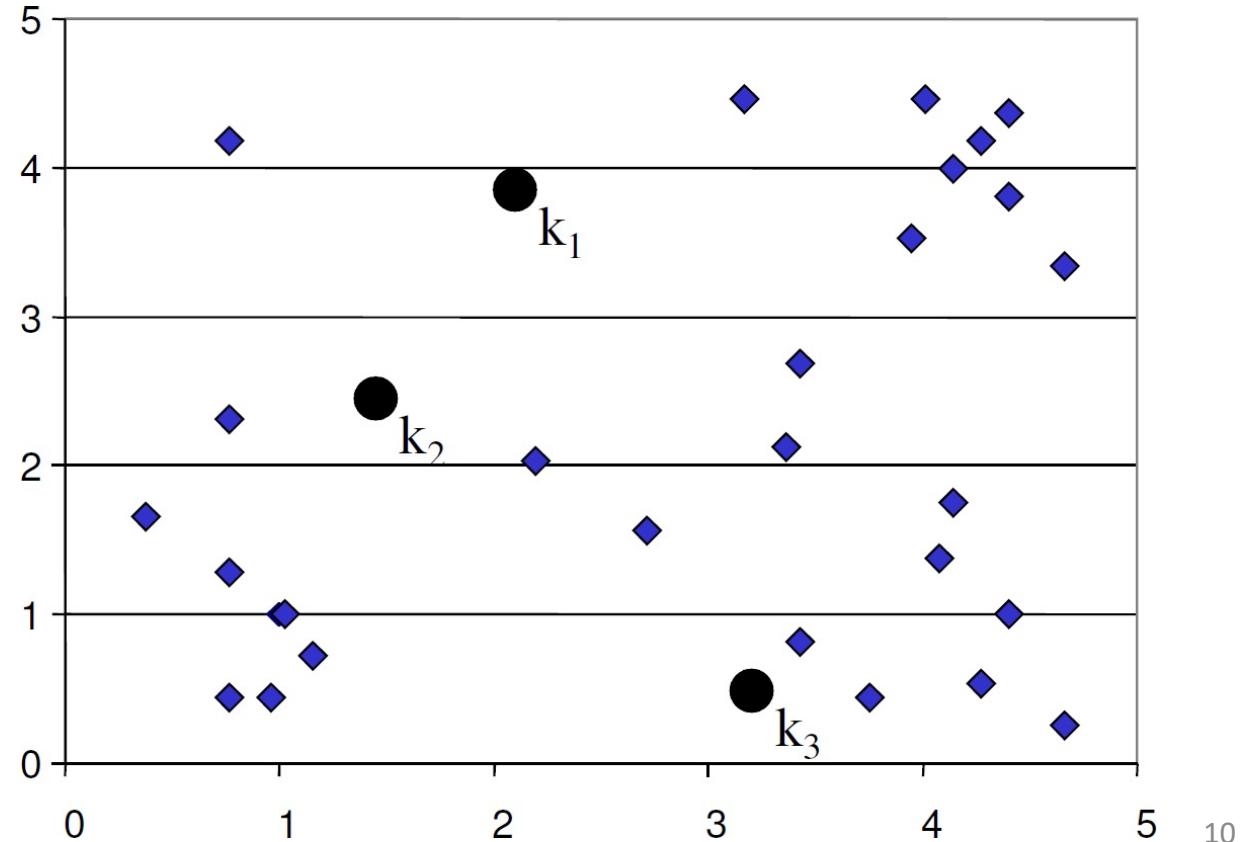
Clustering Particional

Procedimiento

Paso previo: Se elije el número de **K de grupos** en los que se asignarán los elementos.

1

Paso 1: Seleccionar k centroides aleatoriamente. Estos serán los centros iniciales en los k grupos. Por ejemplo, 3 centroides (elementos).

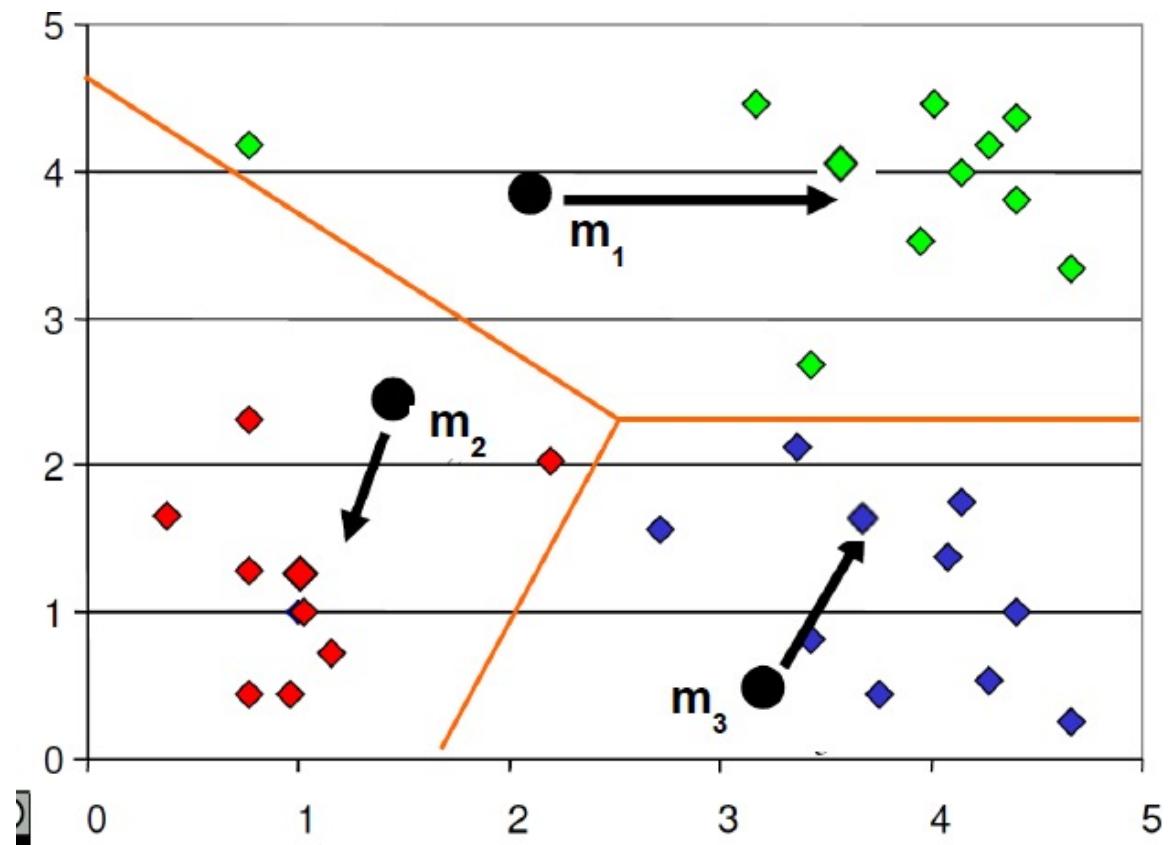


Clustering Particional

Procedimiento

2

Paso 2: Se asigna cada elemento al centroide más cercano, creando así k clústeres.
Para la asignación se utiliza mediciones de **distancia mínima** entre el elemento y el centroide.

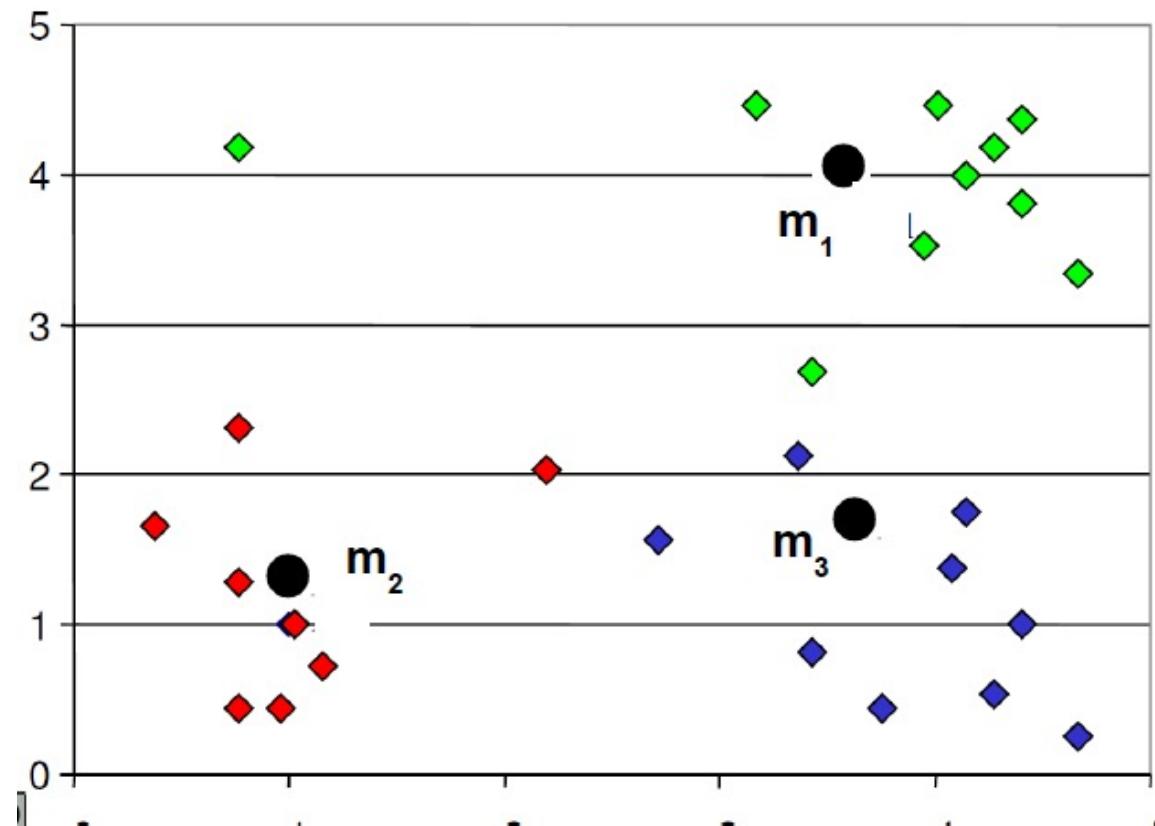


Clustering Particional

Procedimiento

3

Paso 3: Una vez asignados todos los elementos, se actualiza la posición de los **centroides**, tomando como nuevo centro la posición del promedio de los elementos pertenecientes a cada clúster.

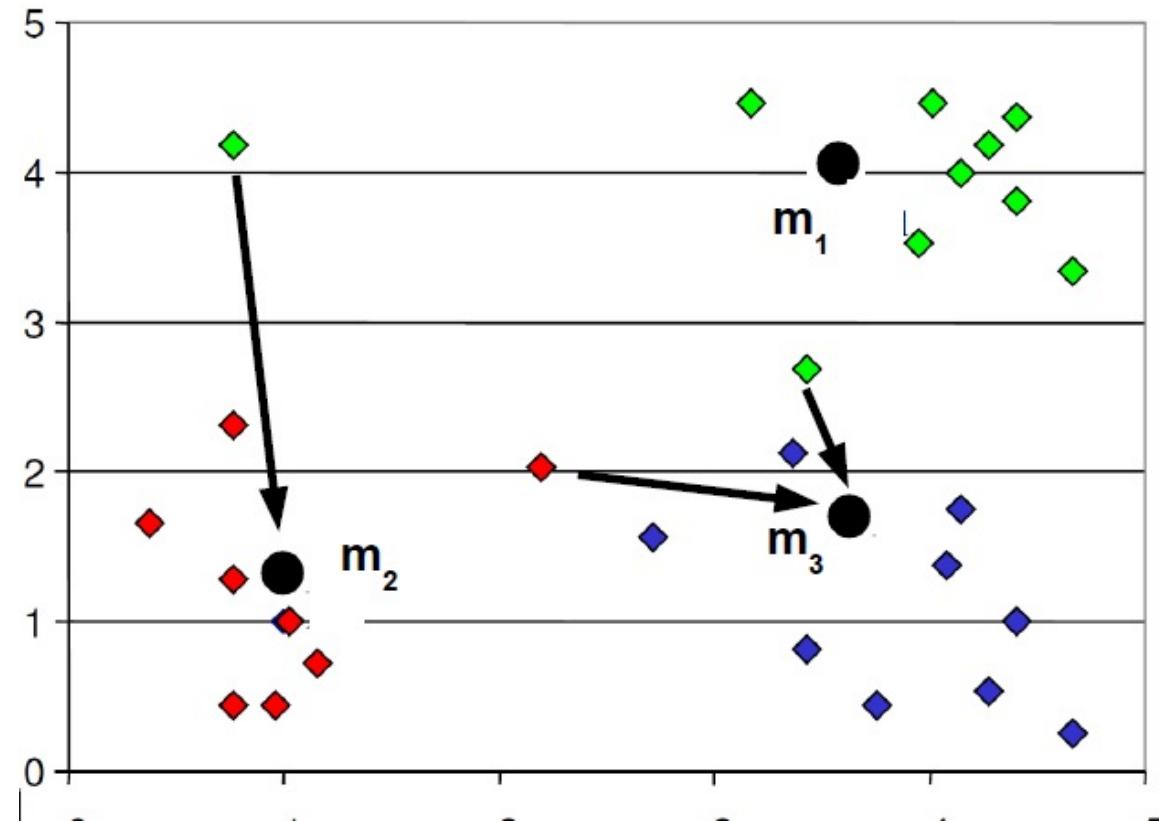


Clustering Particional

Procedimiento

4

Paso 4: Se repiten los pasos **2 y 3**, se vuelven a asignar los elementos y se recalculan los centroides, hasta que éstos (centroides) no se modifiquen más, o se alcance un número máximo de iteraciones.



Clustering Particional

Procedimiento con una matriz de datos

Se quiere dividir una población de usuarios de un determinado sitio Web (Netflix) con base en sus edades: **n = 18**

ID	x _i
1	15
2	15
3	16
4	19
5	19
6	20
7	20
8	21
9	22
10	28
11	35
12	40
13	41
14	42
15	43
16	44
17	60
18	61

Distancia de Manhattan:

$$d_{Manh}(p, q) = \sum_{i=1}^n |p_i - q_i|$$

Clustering Particional

Procedimiento con una tabla de datos

Iteración 1

ID	x_i	c_1	c_2	c_3	Distancia 1	Distancia 2	Distancia 3	Cluster Cercano	Nuevo Centroide
1	15	16	22	60	1	7	45	1	
2	15	16	22	60	1	7	45	1	
3	16	16	22	60	0	6	44	1	16.8
4	19	16	22	60	3	3	41	1	
5	19	16	22	60	3	3	41	1	
6	20	16	22	60	4	2	40	2	
7	20	16	22	60	4	2	40	2	
8	21	16	22	60	5	1	39	2	
9	22	16	22	60	6	0	38	2	
10	28	16	22	60	12	6	32	2	
11	35	16	22	60	19	13	25	2	
12	40	16	22	60	24	18	20	2	
13	41	16	22	60	25	19	19	2	
14	42	16	22	60	26	20	18	3	
15	43	16	22	60	27	21	17	3	
16	44	16	22	60	28	22	16	3	
17	60	16	22	60	44	38	0	3	
18	61	16	22	60	45	39	1	3	

Se elije el número de **K de grupos** en los que se asignarán los elementos.

$$k = 3$$

$$c_1 = 16$$

$$c_2 = 22$$

$$c_3 = 60$$

$$d_{Manh}(p, q) = \sum_{i=1}^n |p_i - q_i|$$

$$\text{Distancia 1} = |x_i - c_1| = |15 - 16| = 1$$

$$\text{Distancia 2} = |x_i - c_2| = |15 - 22| = 7$$

$$\text{Distancia 3} = |x_i - c_3| = |15 - 60| = 45$$

Clustering Particional

Procedimiento con una tabla de datos

Iteración 2

ID	x _i	Centroide			Distancia	Distancia	Distancia	Cluster Cercano	Nuevo Centroide
		c ₁	c ₂	c ₃	1	2	3		
1	15	16.8	28.4	50	1.8	13.4	35	1	18.6
2	15	16.8	28.4	50	1.8	13.4	35		
3	16	16.8	28.4	50	0.8	12.4	34		
4	19	16.8	28.4	50	2.2	9.4	31		
5	19	16.8	28.4	50	2.2	9.4	31		
6	20	16.8	28.4	50	3.2	8.4	30		
7	20	16.8	28.4	50	3.2	8.4	30		
8	21	16.8	28.4	50	4.2	7.4	29		
9	22	16.8	28.4	50	5.2	6.4	28		
10	28	16.8	28.4	50	11.2	0.4	22		
11	35	16.8	28.4	50	18.2	6.6	15	2	31.5
12	40	16.8	28.4	50	23.2	11.6	10		
13	41	16.8	28.4	50	24.2	12.6	9		
14	42	16.8	28.4	50	25.2	13.6	8		
15	43	16.8	28.4	50	26.2	14.6	7		
16	44	16.8	28.4	50	27.2	15.6	6		
17	60	16.8	28.4	50	43.2	31.6	10		
18	61	16.8	28.4	50	44.2	32.6	11		

Se actualiza la posición de los **centroides**, y nuevamente se asignan los elementos.

$$k = 3$$

$$c_1 = 16.8$$

$$c_2 = 28.4$$

$$c_3 = 50.0$$

$$d_{Manh}(p, q) = \sum_{i=1}^n |p_i - q_i|$$

$$\text{Distancia 1} = |x_i - c_1|$$

$$\text{Distancia 2} = |x_i - c_2|$$

$$\text{Distancia 3} = |x_i - c_3|$$

Clustering Particional

Procedimiento con una tabla de datos

Iteración 3

ID	x _i	Centroide			Distancia 1	Distancia 2	Distancia 3	Cluster Cercano	Nuevo Centroide
		c ₁	c ₂	c ₃					
1	15	18.6	31.5	47.3	3.6	16.5	32.3	1	18.6
2	15	18.6	31.5	47.3	3.6	16.5	32.3	1	
3	16	18.6	31.5	47.3	2.6	15.5	31.3	1	
4	19	18.6	31.5	47.3	0.4	12.5	28.3	1	
5	19	18.6	31.5	47.3	0.4	12.5	28.3	1	
6	20	18.6	31.5	47.3	1.4	11.5	27.3	1	
7	20	18.6	31.5	47.3	1.4	11.5	27.3	1	
8	21	18.6	31.5	47.3	2.4	10.5	26.3	1	
9	22	18.6	31.5	47.3	3.4	9.5	25.3	1	
10	28	18.6	31.5	47.3	9.4	3.5	19.3	2	31.5
11	35	18.6	31.5	47.3	16.4	3.5	12.3	2	
12	40	18.6	31.5	47.3	21.4	8.5	7.3	3	
13	41	18.6	31.5	47.3	22.4	9.5	6.3	3	
14	42	18.6	31.5	47.3	23.4	10.5	5.3	3	
15	43	18.6	31.5	47.3	24.4	11.5	4.3	3	47.3
16	44	18.6	31.5	47.3	25.4	12.5	3.3	3	
17	60	18.6	31.5	47.3	41.4	28.5	12.7	3	
18	61	18.6	31.5	47.3	42.4	29.5	13.7	3	

Se actualiza la posición de los **centroides**, y nuevamente se asignan los elementos.

$$k = 3$$

$$c_1 = 18.6$$

$$c_2 = 31.5$$

$$c_3 = 47.3$$

$$d_{Manh}(p, q) = \sum_{i=1}^n |p_i - q_i|$$

$$\text{Distancia 1} = |x_i - c_1|$$

$$\text{Distancia 2} = |x_i - c_2|$$

$$\text{Distancia 3} = |x_i - c_3|$$

Clustering Particional

Procedimiento con una tabla de datos

Iteración 4

ID	x_i	c_1	c_2	c_3	Distancia 1	Distancia 2	Distancia 3	Cluster Cercano	Nuevo Centroide
1	15	18.6	31.5	47.3	3.6	16.5	32.3	1	
2	15	18.6	31.5	47.3	3.6	16.5	32.3	1	
3	16	18.6	31.5	47.3	2.6	15.5	31.3	1	
4	19	18.6	31.5	47.3	0.4	12.5	28.3	1	
5	19	18.6	31.5	47.3	0.4	12.5	28.3	1	
6	20	18.6	31.5	47.3	1.4	11.5	27.3	1	
7	20	18.6	31.5	47.3	1.4	11.5	27.3	1	
8	21	18.6	31.5	47.3	2.4	10.5	26.3	1	
9	22	18.6	31.5	47.3	3.4	9.5	25.3	1	
10	28	18.6	31.5	47.3	9.4	3.5	19.3	2	
11	35	18.6	31.5	47.3	16.4	3.5	12.3	2	
12	40	18.6	31.5	47.3	21.4	8.5	7.3	3	
13	41	18.6	31.5	47.3	22.4	9.5	6.3	3	
14	42	18.6	31.5	47.3	23.4	10.5	5.3	3	
15	43	18.6	31.5	47.3	24.4	11.5	4.3	3	
16	44	18.6	31.5	47.3	25.4	12.5	3.3	3	
17	60	18.6	31.5	47.3	41.4	28.5	12.7	3	
18	61	18.6	31.5	47.3	42.4	29.5	13.7	3	

$k = 3$

$$c_1 = 18.6$$

$$c_2 = 31.5$$

$$c_3 = 47.3$$

$$d_{Manh}(p, q) = \sum_{i=1}^n |p_i - q_i|$$

$$\text{Distancia 1} = |x_i - c_1|$$

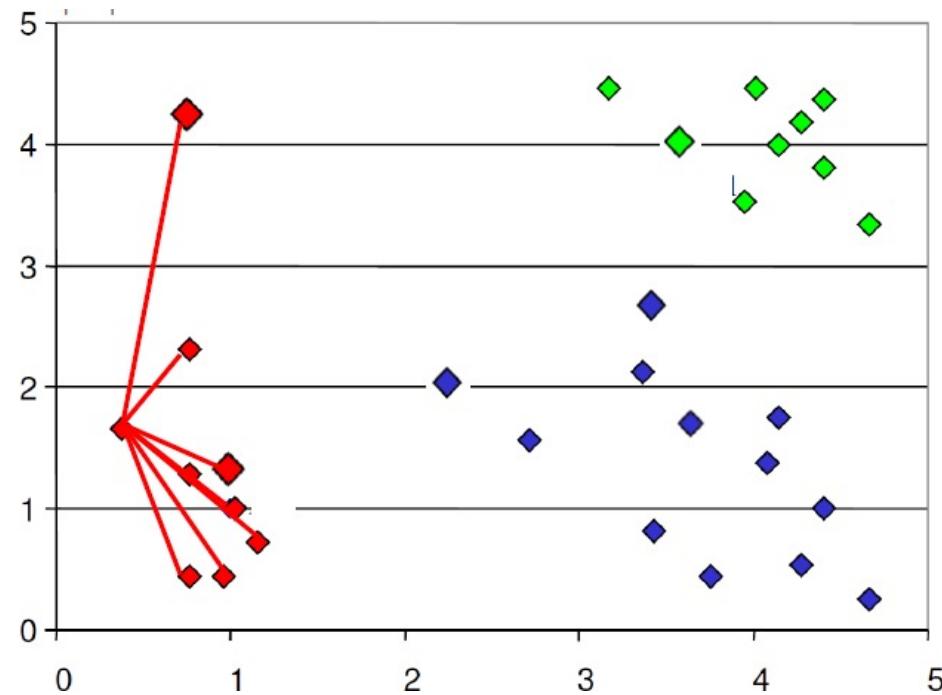
$$\text{Distancia 2} = |x_i - c_2|$$

$$\text{Distancia 3} = |x_i - c_3|$$

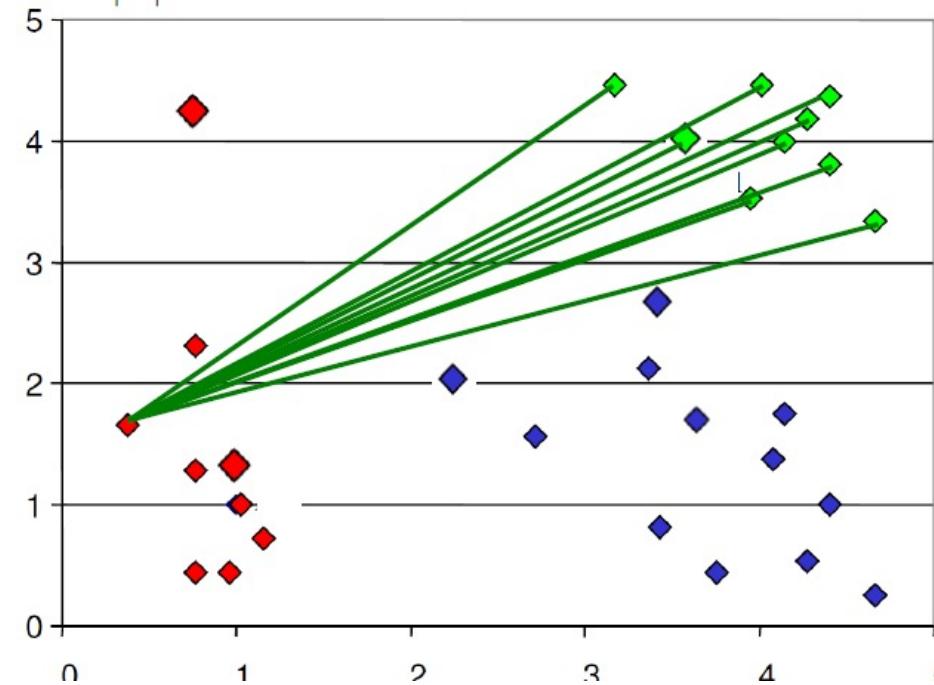
Clustering Particional

Lo que se busca

La similitud entre los elementos del mismo clúster sea alta. **Similitud intraclúster alta.**



La similitud entre los elementos de distintos clústeres sea baja. **Similitud interclúster baja.**

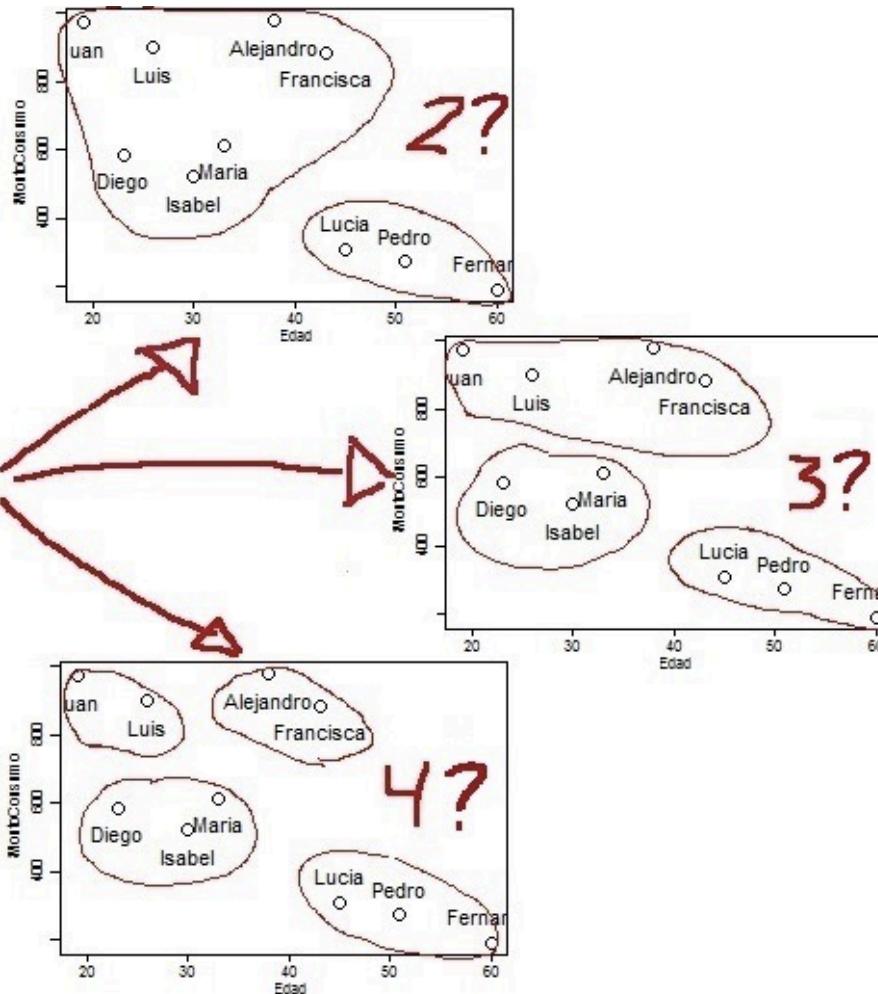


Clustering Particional

La idea básica de los algoritmos de partición, como K-means, es definir el número de grupos.

Nombre	Edad	MontoConsumo
Juan	19	971
Pedro	51	271
Maria	33	614
Isabel	30	521
Diego	23	585
Luis	26	898
Lucia	45	310
Francisca	43	884
Alejandro	38	979
Fernando	60	189

Cuántos Grupos?



Método para decidir la cantidad de grupos

Elbow method

Es una herramienta gráfica útil para estimar el número adecuado de grupos. El propósito es identificar el valor de k, donde la distorsión (efecto del codo) cambia de manera significativa.

Para esto

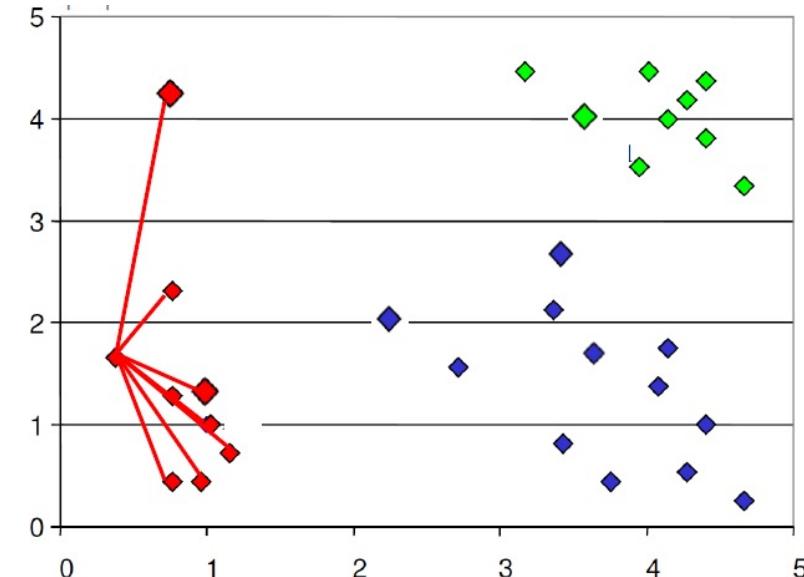
- Se debe calcular **SSE –tot.withiness–** (suma de la distancia al cuadrado entre cada elemento del clúster y su centroide) para varias configuraciones de k.
Por ejemplo, k = 2, 3, 4, 5, 6, 7, 8 ... n.

$$SSE = \sum_{k=1}^k \text{dist}(x_i, u_k) = \sum_{k=1}^k \sum_{x_i \in C_k} (x_i - u_k)^2$$

Número de clúster

Elemento del clúster

Centroide



Elbow method

SSE

$$SSE = \sum_{k=1}^k \text{dist}(x_i, u_k) = \sum_{k=1}^k \sum_{x_i \in C_k} (x_i - u_k)^2$$

ID	x_i	c_1	c_2	c_3	Distancia 1	Distancia 2	Distancia 3	Cluster Cercano	Nuevo Centroide	$ x_i - u_k $	$ x_i - u_k ^2$	SSE
1	15	18.6	31.5	47.3	3.6	16.5	32.3	1	18.6	3.6	12.96	54.24
2	15	18.6	31.5	47.3	3.6	16.5	32.3	1		3.6	12.96	
3	16	18.6	31.5	47.3	2.6	15.5	31.3	1		2.6	6.76	
4	19	18.6	31.5	47.3	0.4	12.5	28.3	1		0.4	0.16	
5	19	18.6	31.5	47.3	0.4	12.5	28.3	1		0.4	0.16	
6	20	18.6	31.5	47.3	1.4	11.5	27.3	1		1.4	1.96	
7	20	18.6	31.5	47.3	1.4	11.5	27.3	1		1.4	1.96	
8	21	18.6	31.5	47.3	2.4	10.5	26.3	1		2.4	5.76	
9	22	18.6	31.5	47.3	3.4	9.5	25.3	1		3.4	11.56	
10	28	18.6	31.5	47.3	9.4	3.5	19.3	2	31.5	3.5	12.25	24.50
11	35	18.6	31.5	47.3	16.4	3.5	12.3	2		3.5	12.25	
12	40	18.6	31.5	47.3	21.4	8.5	7.3	3	47.3	7.3	53.29	499.4
13	41	18.6	31.5	47.3	22.4	9.5	6.3	3		6.3	39.69	
14	42	18.6	31.5	47.3	23.4	10.5	5.3	3		5.3	28.09	
15	43	18.6	31.5	47.3	24.4	11.5	4.3	3		4.3	18.49	
16	44	18.6	31.5	47.3	25.4	12.5	3.3	3		3.3	10.89	
17	60	18.6	31.5	47.3	41.4	28.5	12.7	3		12.7	161.29	
18	61	18.6	31.5	47.3	42.4	29.5	13.7	3		13.7	187.69	

578.17

Algoritmo

1. Calcular el agrupamiento para **diferentes valores de k**. Por ejemplo, k de 2 a 12 grupos.
2. Para cada k , calcular la suma total de la distancia al cuadrado dentro de cada grupo (**SSE**, conocido también como **WSS**).
3. **Trazar la curva de SSE** de acuerdo con el número de grupos k .
4. La ubicación de una curva (efecto del codo) en el gráfico se considera como un indicador del número adecuado de grupos.

$$SSE = \text{tot.withiness} = \sum_{k=1}^k \text{dist}(x_i, u_k) = \sum_{k=1}^k \sum_{x_i \in C_k} (x_i - u_k)^2$$

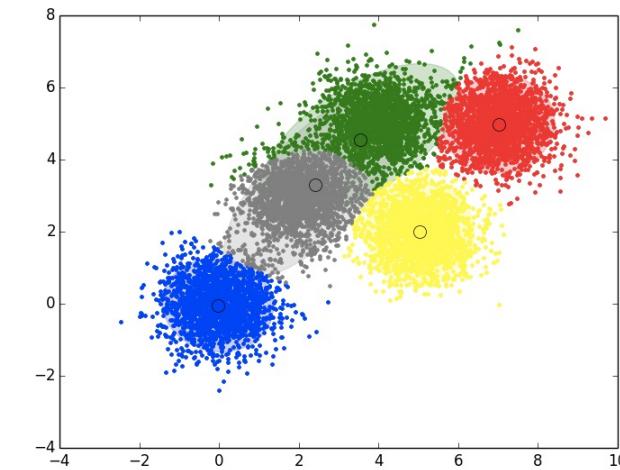
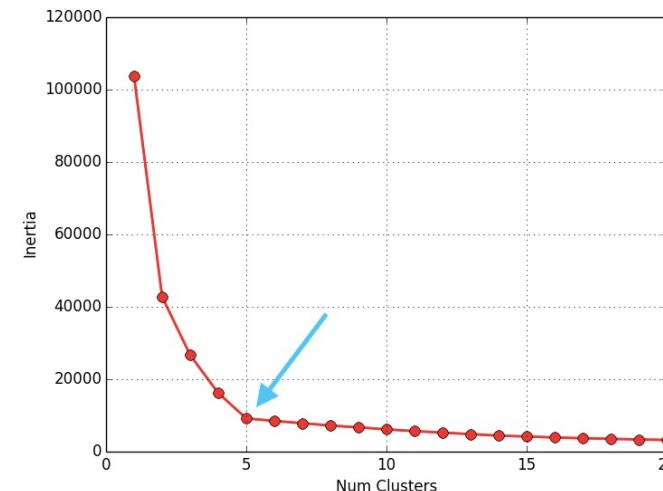
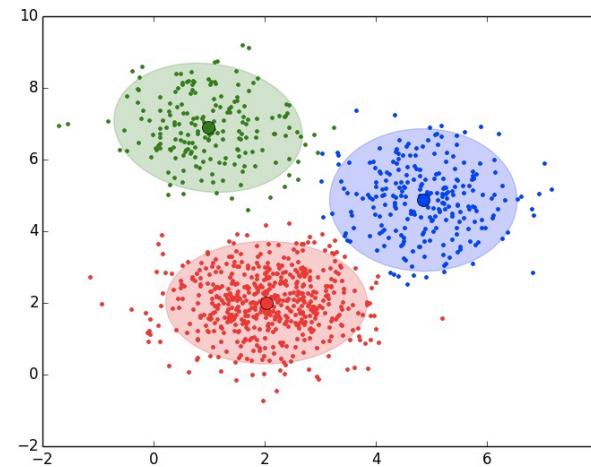
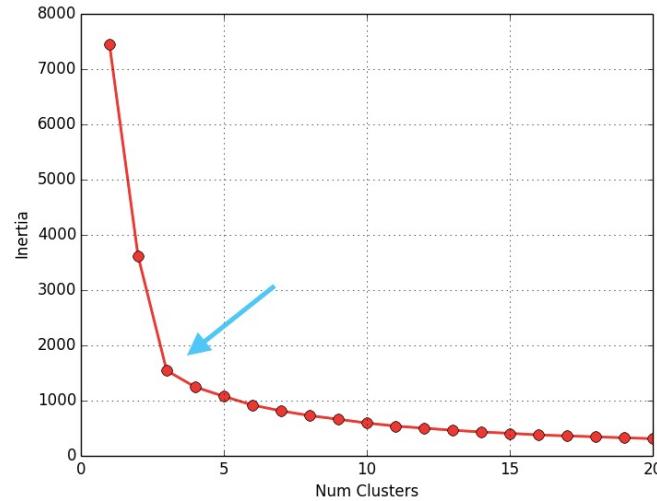
Número de clúster

Elemento del clúster

Centroide

Elbow method

El punto de inflexión se considera como un indicador del número adecuado de grupos.



Elbow method

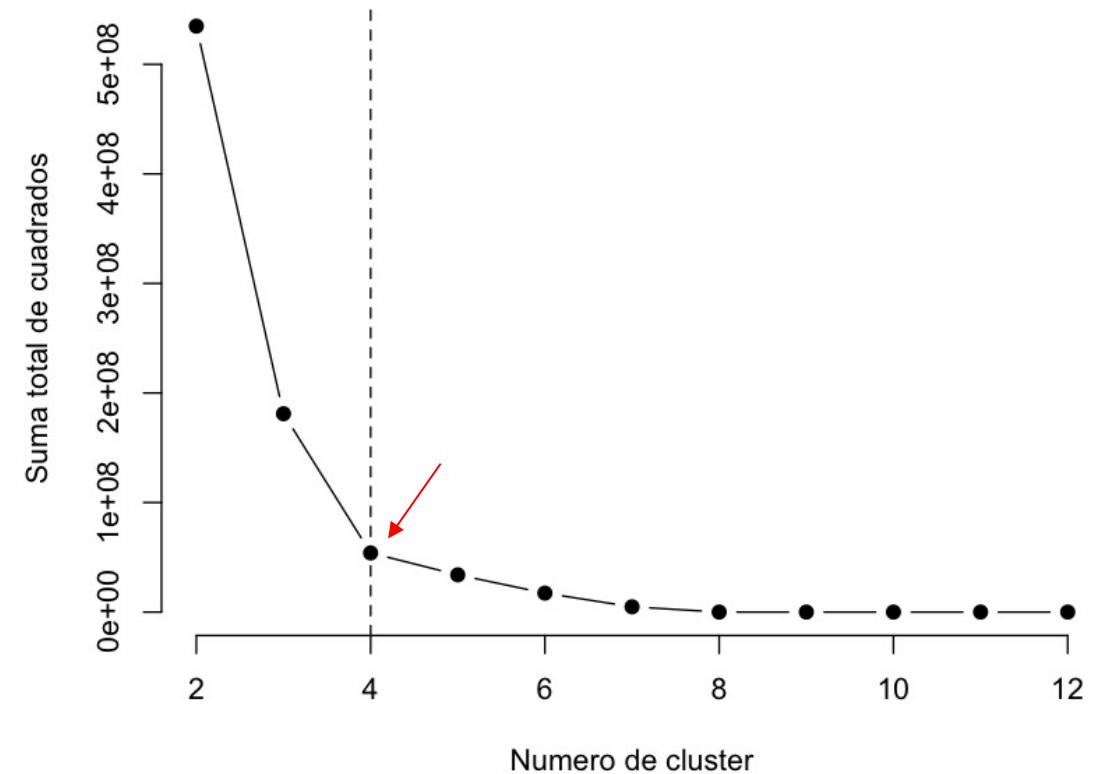
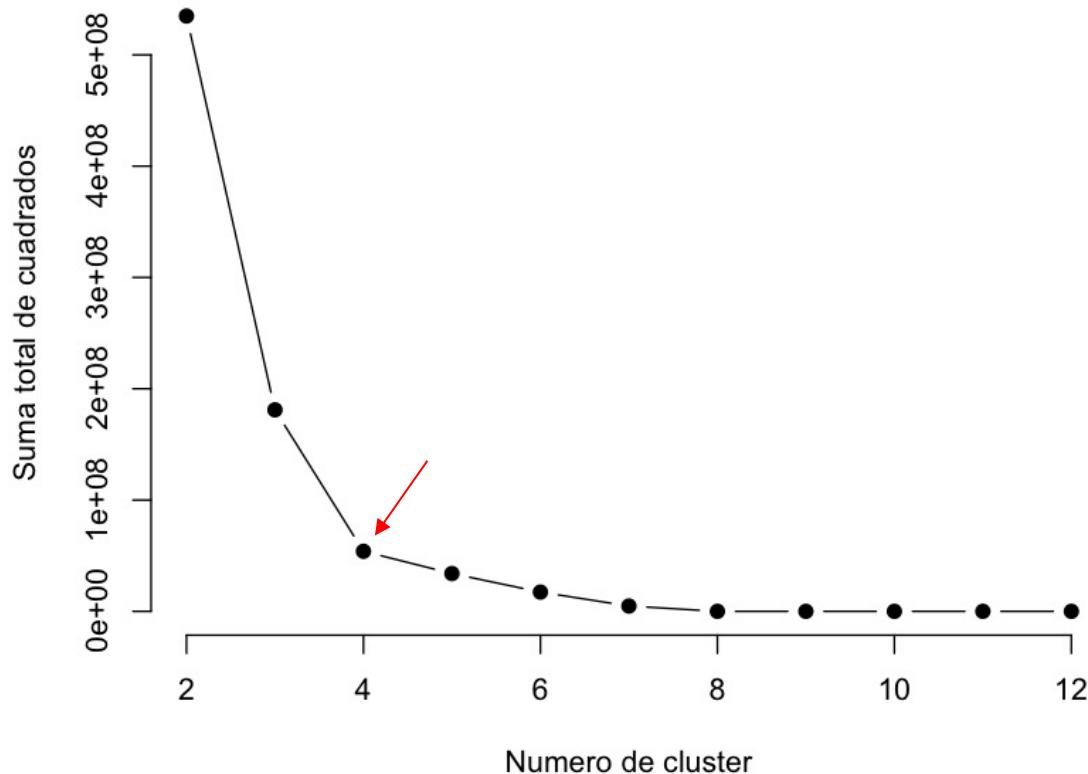
Ejemplo ilustrativo

	ID	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	FaltasAño	Antiguedad	Sexo
1	E1	10000	1	0	0	0	0	7	15	1
2	E2	20000	0	1	1	0	1	3	3	0
3	E3	15000	1	1	2	1	1	5	10	1
4	E4	30000	1	1	1	0	0	15	7	0
5	E5	10000	1	1	0	1	1	1	6	1
6	E6	40000	0	1	0	0	1	3	16	0
7	E7	25000	0	0	0	0	1	0	8	1
8	E8	20000	0	1	0	1	1	2	6	0
9	E9	20000	1	1	3	1	0	7	5	1
10	E10	30000	1	1	2	1	0	1	20	1
11	E11	45000	0	0	0	0	0	2	12	0
12	E12	8000	1	1	2	1	0	3	1	1
13	E13	20000	0	0	0	0	0	27	5	0
14	E14	10000	0	1	0	0	1	0	7	1
15	E15	8000	0	1	0	0	0	3	2	1

Elbow method

1

Obtención de número de grupos (Elbow method)



2

Obtención de los grupos

K-means clustering with 4 clusters of sizes 3, 2, 5, 5

Cluster means:

	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	FaltasAnno	Antiguedad	Sexo
1	28333.33	0.6666667	0.6666667	1.0	0.3333333	0.3333333	5.333333	11.66667	0.6666667
2	42500.00	0.0000000	0.5000000	0.0	0.0000000	0.5000000	2.500000	14.00000	0.0000000
3	9200.00	0.6000000	0.8000000	0.4	0.4000000	0.4000000	2.800000	6.20000	1.0000000
4	19000.00	0.4000000	0.8000000	1.2	0.6000000	0.6000000	8.800000	5.80000	0.4000000

Clustering vector:

[1] 3 4 4 1 3 2 1 4 4 1 2 3 4 3 3

Within cluster sum of squares by cluster:

[1] 16666917 12500010 4800159 20000468

3 Interpretación

	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	FaltasAnno	Antiguedad	Sexo
1	28333.33	0.6666667	0.6666667	1.0	0.3333333	0.3333333	5.333333	11.66667	0.6666667
2	42500.00	0.0000000	0.5000000	0.0	0.0000000	0.5000000	2.500000	14.00000	0.0000000
3	9200.00	0.6000000	0.8000000	0.4	0.4000000	0.4000000	2.800000	6.20000	1.0000000
4	19000.00	0.4000000	0.8000000	1.2	0.6000000	0.6000000	8.800000	5.80000	0.4000000

Cluster 1: 3 empleados

Salario : 28333
 Casado : Si = 0.67 / No = 0.33
 Coche : Si = 0.67 / No = 0.33
 Hijos : 1
 Vivienda : Prop = 0.33
 Alquiler = 0.67
 Sindicato : Si = 0.33 / No = 0.67
 Faltas/Año : 5.3 (5)
 Antigüedad : 11.6 (12)
 Sexo : M = 0.67 / F = 0.33

...

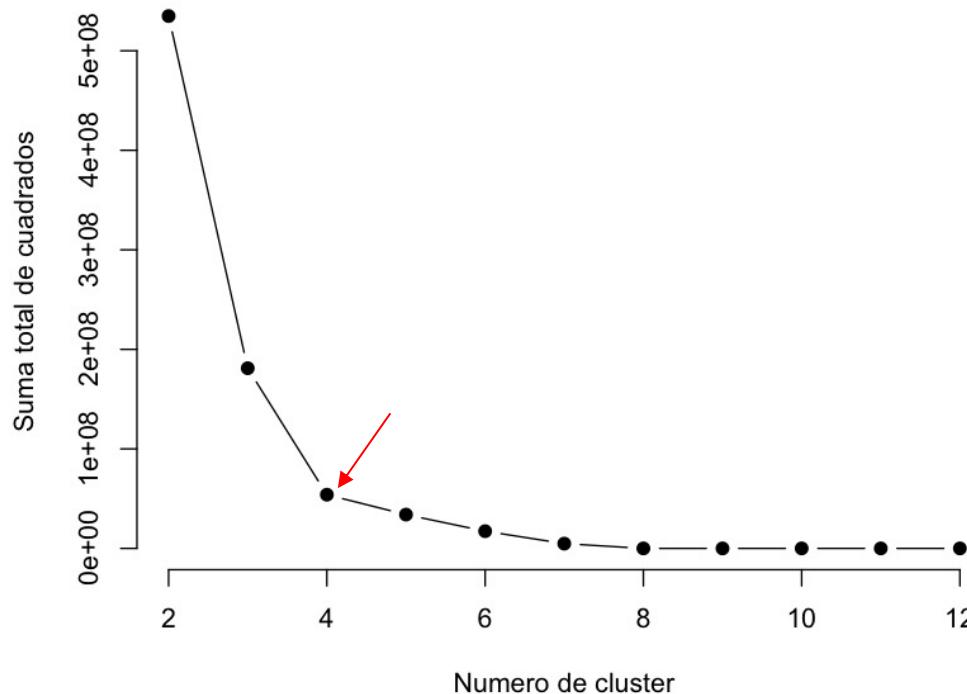
ID	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	FaltasAño	Antiguedad	Sexo
1	E1	10000	1	0	0	0	0	7	15 1
2	E2	20000	0	1	1	0	1	3	3 0
3	E3	15000	1	1	2	1	1	5	10 1
4	E4	30000	1	1	1	0	0	15	7 0
5	E5	10000	1	1	0	1	1	1	6 1
6	E6	40000	0	1	0	0	1	3	16 0
7	E7	25000	0	0	0	0	1	0	8 1
8	E8	20000	0	1	0	1	1	2	6 0
9	E9	20000	1	1	3	1	0	7	5 1
10	E10	30000	1	1	2	1	0	1	20 1
11	E11	45000	0	0	0	0	0	2	12 0
12	E12	8000	1	1	2	1	0	3	1 1
13	E13	20000	0	0	0	0	0	27	5 0
14	E14	10000	0	1	0	0	1	0	7 1
15	E15	8000	0	1	0	0	0	3	2 1

- **Cluster 1 [3 elementos –4, 7, 10–].** Empleados con salario promedio de \$28333, casados en su mayoría (67%), con coche en su mayoría (67%) y con un hijo. No tienen vivienda propia en su mayoría (67%), no sindicalizados en su mayoría (67%), con varias faltas al año (5), con una antigüedad promedio de 12 años y la mayoría varones (67%).

Elbow method

K-means

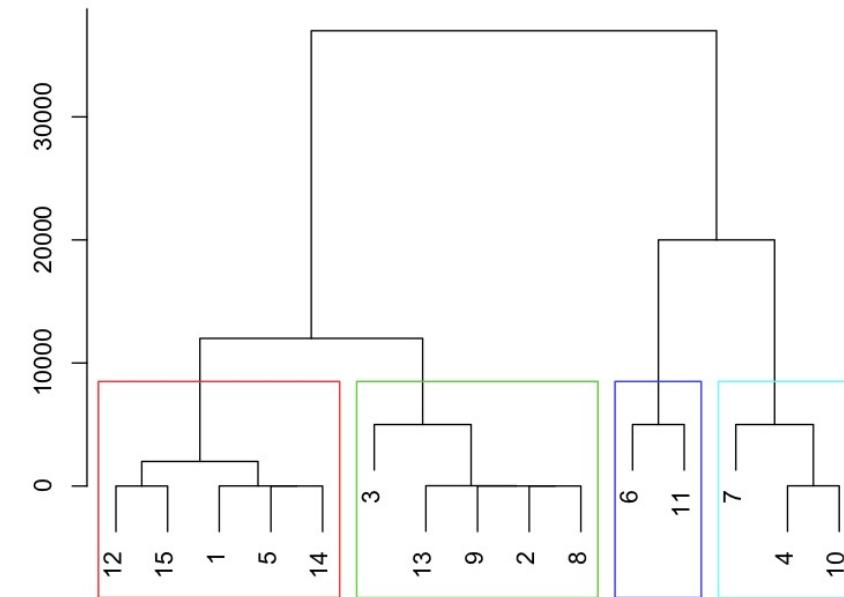
	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	FaltasAnno	Antiguedad	Sexo
1	28333.33	0.6666667	0.6666667	1.0	0.3333333	0.3333333	5.333333	11.66667	0.6666667
2	42500.00	0.0000000	0.5000000	0.0	0.0000000	0.5000000	2.500000	14.00000	0.0000000
3	9200.00	0.6000000	0.8000000	0.4	0.4000000	0.4000000	2.800000	6.20000	1.0000000
4	19000.00	0.4000000	0.8000000	1.2	0.6000000	0.6000000	8.800000	5.80000	0.4000000



Jerárquico Ascendente

	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	FaltasAnno	Antiguedad	Sexo
[1,]	9200.00	0.6000000	0.8000000	0.4	0.4000000	0.4000000	2.800000	6.20000	1.0000000
[2,]	19000.00	0.4000000	0.8000000	1.2	0.6000000	0.6000000	8.800000	5.80000	0.4000000
[3,]	28333.33	0.6666667	0.6666667	1.0	0.3333333	0.3333333	5.333333	11.66667	0.6666667
[4,]	42500.00	0.0000000	0.5000000	0.0	0.0000000	0.5000000	0.0	0.0000000	0.5000000

Cluster Dendrogram



Consideraciones finales

- Aumentar la cantidad de **clústeres** mejorará naturalmente el ajuste (se hará una mejor explicación de la variación). Sin embargo, se puede caer en un sobre ajuste, ya que se está dividiendo en múltiples grupos.
- En la práctica, puede que no exista un codo afilado (codo agudo) y, como método heurístico, ese "codo" no siempre puede identificarse sin ambigüedades.



Universidad Nacional Autónoma de México
Facultad de Ingeniería

Clustering Particional

Práctica 5

Guillermo Molero-Castillo
guillermo.molero@ingenieria.unam.edu

Octubre, 2021

Classical Machine Learning

Supervised Learning

(Pre Categorized Data)

Classification

(Divide the socks by Color)

Eg. Identity Fraud Detection

Regression

(Divide the Ties by Length)

Eg. Market Forecasting

Unsupervised Learning

(Unlabelled Data)

Clustering

(Divide by Similarity)

Eg. Targeted Marketing

Association

(Identify Sequences)

Eg. Customer Recommendation

Dimensionality Reduction

(Wider Dependencies)

Eg. Big Data Visualization

Task Driven

Data Driven

</div

Fuente de datos

- ingresos: son ingresos mensuales de 1 o 2 personas, si están casados.
- gastos_comunes: son gastos mensuales de 1 o 2 personas, si están casados.
- pago_coche
- gastos_otros
- ahorros
- vivienda: valor de la vivienda.
- estado_civil: 0-soltero, 1-casado, 2-divorciado
- hijos: cantidad de hijos menores (no trabajan).
- trabajo: 0-sin trabajo, 1-autonomo, 2-asalariado, 3-empresario, 4-autonomos, 5-asalariados, 6-autonomo y asalariado, 7-empresario y autonomo, 8-empresarios o empresario y autónomo
- comprar: 0-alquilar, 1-comprar casa a través de crédito hipotecario con tasa fija a 30 años.

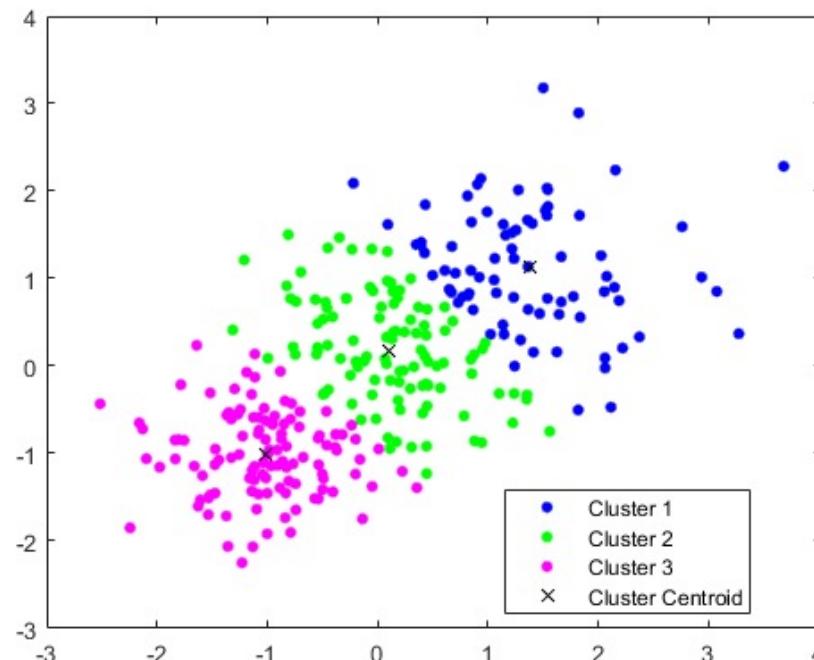
Práctica

Fuente de datos

ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo	comprar
6000	1000	0	600	50000	400000	0	2	2	1
6745	944	123	429	43240	636897	1	3	6	0
6455	1033	98	795	57463	321779	2	1	8	1
7098	1278	15	254	54506	660933	0	0	3	0
6167	863	223	520	41512	348932	0	0	3	1
5692	911	11	325	50875	360863	1	4	5	1

Objetivo

Obtener clústeres de casos de usuarios, con características similares, evaluados para la adquisición de una casa a través de un crédito hipotecario con tasa fija a 30 años.



1. Importar las bibliotecas y los datos

```
▶ import pandas as pd          # Para la manipulación y análisis de datos  
    import numpy as np          # Para crear vectores y matrices n dimensionales  
    import matplotlib.pyplot as plt # Para la generación de gráficas a partir de los datos  
    import seaborn as sns         # Para la visualización de datos basado en matplotlib  
    %matplotlib inline  
  
▶ from google.colab import files  
files.upload()  
  
#from google.colab import drive  
#drive.mount('/content/drive')  
  
□ Elegir archivos Hipoteca.csv  
• Hipoteca.csv(text/csv) - 8014 bytes, last modified: 1/4/2021 - 100% done  
Saving Hipoteca.csv to Hipoteca.csv  
{'Hipoteca.csv': b'ingresos,gastos_comunes,pago_coche,gastos_otros,ahorros,vivienda,estado_civil'}
```

Práctica

1. Importar las bibliotecas y los datos



```
Hipoteca = pd.read_csv("Hipoteca.csv")  
Hipoteca
```

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo	comprar
0	6000	1000	0	600	50000	400000	0	2	2	1
1	6745	944	123	429	43240	636897	1	3	6	0
2	6455	1033	98	795	57463	321779	2	1	8	1
3	7098	1278	15	254	54506	660933	0	0	3	0
4	6167	863	223	520	41512	348932	0	0	3	1
...
197	3831	690	352	488	10723	363120	0	0	2	0
198	3961	1030	270							
199	3184	955	276							
200	3334	867	369							
201	3988	1157	105							

202 rows × 10 columns



```
#'comprar' representa un valor obtenido de un análisis hipotecario preliminar  
print(Hipoteca.groupby('comprar').size())
```

comprar	size
0	135
1	67

dtype: int64

1. Importar las bibliotecas y los datos



Hipoteca.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 202 entries, 0 to 201
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   ingresos         202 non-null    int64  
 1   gastos_comunes  202 non-null    int64  
 2   pago_coche       202 non-null    int64  
 3   gastos_otros    202 non-null    int64  
 4   ahorros          202 non-null    int64  
 5   vivienda         202 non-null    int64  
 6   estado_civil    202 non-null    int64  
 7   hijos            202 non-null    int64  
 8   trabajo          202 non-null    int64  
 9   comprar          202 non-null    int64  
dtypes: int64(10)
memory usage: 15.9 KB
```



print(Hipoteca.groupby('comprar').size())

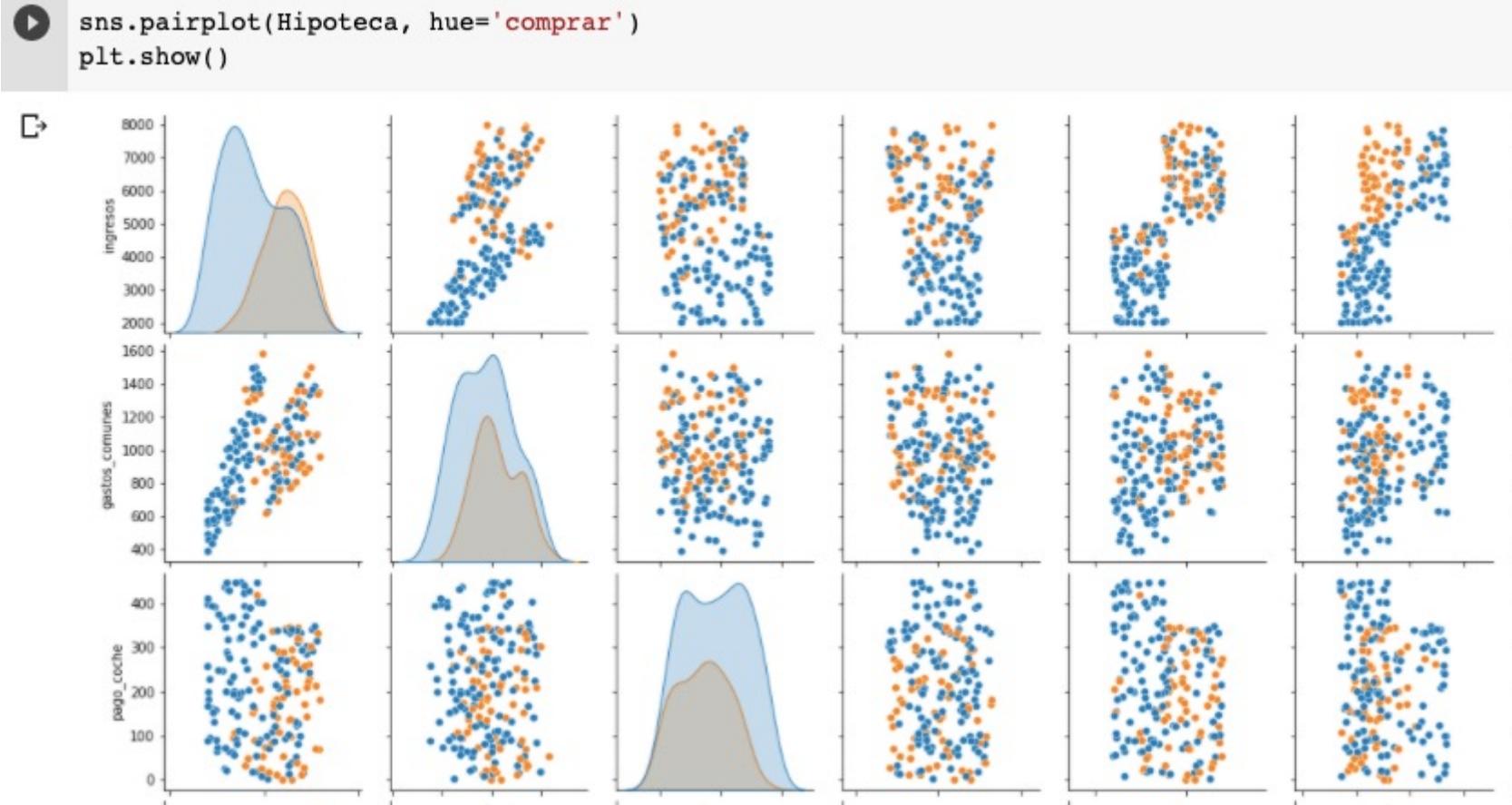
comprar

0	135
1	67

dtype: int64

2. Selección de características

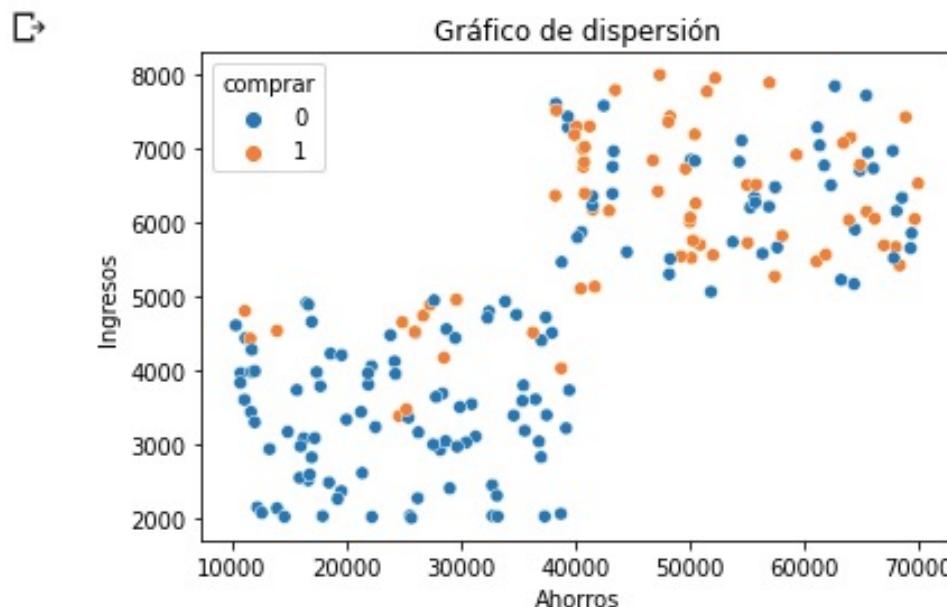
Evaluación visual



2. Selección de características

Evaluación visual

```
▶ sns.scatterplot(x='ahorros', y ='ingresos', data=Hipoteca, hue='comprar')
plt.title('Gráfico de dispersión')
plt.xlabel('Ahorros')
plt.ylabel('Ingresos')
plt.show()
```



2. Selección de características

Matriz de correlaciones

```
CorrHipoteca = Hipoteca.corr(method='pearson')
CorrHipoteca
```

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo	comprar
ingresos	1.000000	0.560211	-0.109780	-0.124105	0.712889	0.614721	-0.042556	-0.024483	-0.038852	0.467123
gastos_comunes	0.560211	1.000000	-0.054400	-0.099881	0.209414	0.204781	-0.057152	-0.072321	-0.079095	0.200191
pago_coche	-0.109780	-0.054400	1.000000	0.010602	-0.193299	-0.094631	0.052239	-0.044858	0.018946	-0.196468
gastos_otros	-0.124105	-0.099881	0.010602	1.000000	-0.064384	-0.054577	-0.020226	0.124845	0.047313	-0.110330
ahorros	0.712889	0.209414	-0.193299	-0.064384	1.000000	0.605836	-0.063039	0.001445	-0.023829	0.340778
vivienda	0.614721	0.204781	-0.094631	-0.054577	0.605836	1.000000	-0.113420	-0.141924	-0.211790	-0.146092
estado_civil	-0.042556	-0.057152	0.052239	-0.020226	-0.063039	-0.113420	1.000000	0.507609	0.589512	0.142799
hijos	-0.024483	-0.072321	-0.044858	0.124845	0.001445	-0.141924	0.507609	1.000000	0.699916	0.272883
trabajo	-0.038852	-0.079095	0.018946	0.047313	-0.023829	-0.211790	0.589512	0.699916	1.000000	0.341537
comprar	0.467123	0.200191	-0.196468	-0.110330	0.340778	-0.146092	0.142799	0.272883	0.341537	1.000000

2. Selección de características

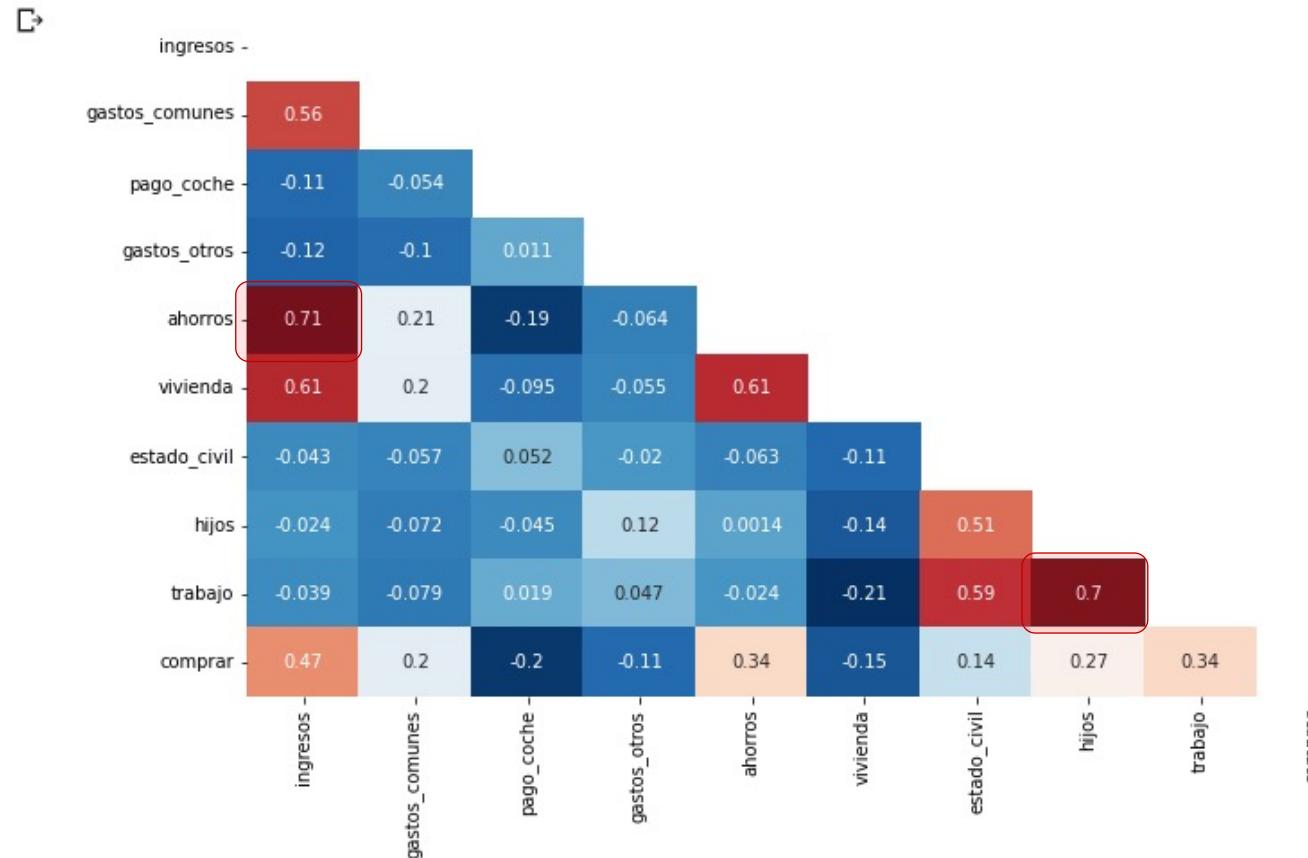
Matriz de correlaciones

```
▶ print(CorrHipoteca['ingresos'].sort_values(ascending=False)[:10], '\n') #Top 10 valores
```

	ingresos	ahorros	vivienda	gastos_comunes	comprar	hijos	trabajo	estado_civil	pago_coche	gastos_otros
	1.000000	0.712889	0.614721	0.560211	0.467123	-0.024483	-0.038852	-0.042556	-0.109780	-0.124105
Name:	ingresos									
dtype:	float64									

2. Selección de características

```
plt.figure(figsize=(14,7))
MatrizInf = np.triu(CorrHipoteca)
sns.heatmap(CorrHipoteca, cmap='RdBu_r', annot=True, mask=MatrizInf)
plt.show()
```



Selección de variables:

- A pesar de existir 2 correlaciones altas, entre 'ingresos' y 'ahorros' (0.71) y 'trabajo' e 'hijos' (0.69); éstas se tomarán en cuenta para obtener una segmentación que combine todas las variables.
- Se suprimirá la variable 'comprar' debido a que representa inherentemente un agrupamiento, y fue un campo calculado con base a un análisis hipotecario preliminar.



2. Selección de características

Elección de variables

```
▶ MatrizHipoteca = np.array(Hipoteca[['ingresos', 'gastos_comunes', ' pago_coche', 'gastos_otros', 'ahorros', 'vivienda',  
pd.DataFrame(MatrizHipoteca)  
#MatrizHipoteca = Hipoteca.iloc[:, 0:9].values      #iloc para seleccionar filas y columnas según su posición
```

	0	1	2	3	4	5	6	7	8
0	6000	1000	0	600	50000	400000	0	2	2
1	6745	944	123	429	43240	636897	1	3	6
2	6455	1033	98	795	57463	321779	2	1	8
3	7098	1278	15	254	54506	660933	0	0	3
4	6167	863	223	520	41512	348932	0	0	3
...

3) Aplicación del algoritmo: K-Means

Estandarización de datos

```
▶ from sklearn.preprocessing import StandardScaler, MinMaxScaler  
estandarizar = StandardScaler() # Se instancia el objeto StandardScaler o MinMaxScaler  
MEstandarizada = estandarizar.fit_transform(MatrizHipoteca) # Se calculan la media y desviación y se escalan los datos  
  
▶ pd.DataFrame(MEstandarizada)  
  
→

|     | 0        | 1         | 2         | 3         | 4        | 5         | 6         | 7         | 8         |
|-----|----------|-----------|-----------|-----------|----------|-----------|-----------|-----------|-----------|
| 0   | 0.620129 | 0.104689  | -1.698954 | 0.504359  | 0.649475 | 0.195910  | -1.227088 | 0.562374  | -0.984420 |
| 1   | 1.063927 | -0.101625 | -0.712042 | -0.515401 | 0.259224 | 1.937370  | -0.029640 | 1.295273  | 0.596915  |
| 2   | 0.891173 | 0.226266  | -0.912634 | 1.667244  | 1.080309 | -0.379102 | 1.167809  | -0.170526 | 1.387582  |
| 3   | 1.274209 | 1.128886  | -1.578599 | -1.559015 | 0.909604 | 2.114062  | -1.227088 | -0.903426 | -0.589086 |
| 4   | 0.719611 | -0.400042 | 0.090326  | 0.027279  | 0.159468 | -0.179497 | -1.227088 | -0.903426 | -0.589086 |
| ... | ...      | ...       | ...       | ...       | ...      | ...       | ...       | ...       | ...       |


```

3) Aplicación del algoritmo: K-means

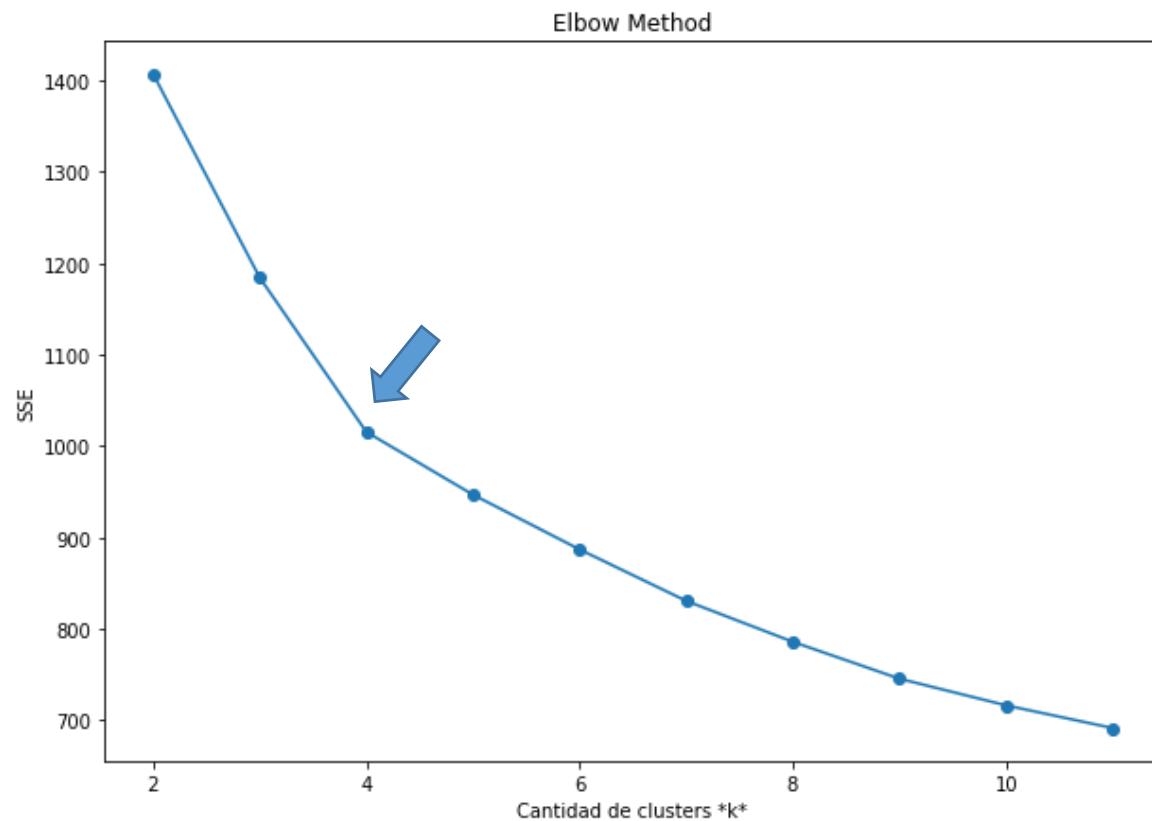
```
▶ #Se importan las bibliotecas
    from sklearn.cluster import KMeans
    from sklearn.metrics import pairwise_distances_argmin_min

▶ #Definición de k clusters para K-means
    #Se utiliza random_state para inicializar el generador interno de números aleatorios
    SSE = []
    for i in range(2, 12):
        km = KMeans(n_clusters=i, random_state=0)
        km.fit(MEstandarizada)
        SSE.append(km.inertia_)

    #Se grafica SSE en función de k
    plt.figure(figsize=(10, 7))
    plt.plot(range(2, 12), SSE, marker='o')
    plt.xlabel('Cantidad de clusters *k*')
    plt.ylabel('SSE')
    plt.title('Elbow Method')
    plt.show()
```

3) Aplicación del algoritmo: K-means

Método del codo



Observación:

En la práctica, puede que no exista un codo afilado (agudo) y, como método heurístico, ese "codo" no siempre puede identificarse sin ambigüedades.

3) Aplicación del algoritmo: K-means

Método del codo

```
▶ !pip install kneed
```

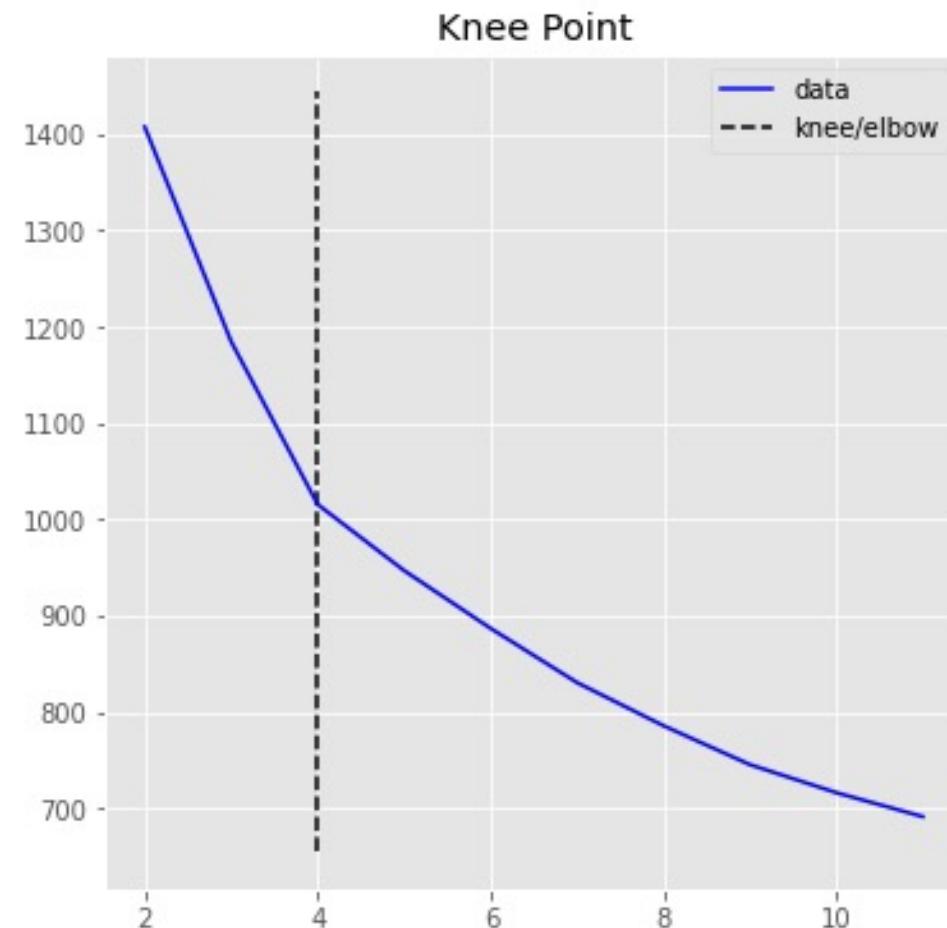
```
▷ Collecting kneed
  Downloading kneed-0.7.0-py2.py3-none-any.whl (9.4 kB)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from kneed)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from k
Requirement already satisfied: numpy>=1.14.2 in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: pyparsing!=2.0.4,!>=2.1.2,!>=2.1.6,>=2.0.1 in /usr/local/lib/p
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from cycler>=
Installing collected packages: kneed
Successfully installed kneed-0.7.0
```

```
▶ from kneed import KneeLocator
kl = KneeLocator(range(2, 12), SSE, curve="convex", direction="decreasing")
kl.elbow
```

3) Aplicación del algoritmo: K-means

Método del codo

```
plt.style.use('ggplot')  
kl.plot_knee()
```



3) Aplicación del algoritmo: K-means

Se crean las etiquetas en los clústeres

▶ #Se crean las etiquetas de los elementos en los clusters
MParticional = KMeans(n_clusters=4, random_state=0).fit(MEstandarizada)
MParticional.predict(MEstandarizada)
MParticional.labels_

↳ array([0, 2, 2, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 0, 2,
0, 2, 0, 0, 2, 0, 2, 0, 2, 0, 0, 2, 0, 2, 0, 2, 2, 2, 2, 0, 2, 2, 2,
0, 0, 3, 2, 2, 0, 1, 1, 1, 1, 3, 1, 3, 3, 3, 3, 3, 1, 1, 3, 1, 3, 1, 3, 1,
1, 3, 1, 3, 1, 1, 1, 3, 1, 3, 1, 1, 3, 1, 0, 3, 3, 1, 1, 3, 1, 3, 1,
1, 3, 3, 1, 1, 3, 3, 1, 3, 3, 1, 3, 1, 2, 0, 2, 2, 0, 0, 2, 0, 2, 0, 2, 0, 2,
2, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 0,
0, 0, 0, 0, 2, 2, 2, 0, 2, 0, 2, 0, 0, 2, 2, 0, 0, 0, 3, 1, 3,
0, 3, 0, 1, 1, 3, 1, 1, 1, 3, 1, 1, 3, 1, 1, 1, 3, 3, 1, 3, 1, 1,
3, 3, 1, 3, 1, 1, 1, 3, 1, 3, 1, 0, 3, 1, 3, 3, 1, 1, 1, 3, 3, 1, 1,
1, 1, 1, 3], dtype=int32)

3) Aplicación del algoritmo: K-means

Se crean las etiquetas en los clústeres

```
▶ Hipoteca = Hipoteca.drop(columns=['comprar'])  
Hipoteca['clusterP'] = MParticional.labels_  
Hipoteca
```

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo	clusterP
0	6000	1000	0	600	50000	400000	0	2	2	0
1	6745	944	123	429	43240	636897	1	3	6	2
2	6455	1033	98	795	57463	321779	2	1	8	2
3	7098	1278	15	254	54506	660933	0	0	3	0
4	6167	863	223	520	41512	348932	0	0	3	0
...
197	3831	#Cantidad de elementos en los clusters Hipoteca.groupby(['clusterP'])['clusterP'].count()								
198	3961	clusterP 0 49								
199	3184	1 56								
200	3334	2 54								
201	3988	3 43								

202 rows × 10 columns

Name: clusterP, dtype: int64

3) Aplicación del algoritmo: K-means

Se crean las etiquetas en los clústeres

▶ Hipoteca[Hipoteca.clusterP == 0]

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo	clusterP
0	6000	1000	0	600	50000	400000	0	2	2	0
3	7098	1278	15	254	54506	660933	0	0	3	0
4	6167	863	223	520	41512	348932	0	0	3	0
6	6830	1298	345	309	46761	429812	1	1	5	0
7	6470	1035	39	782	57439	606291	0	0	1	0
8	6251	1250	209	571	50503	291010	0	0	3	0
10	7273	1455	303	201	39340	577972	0	0	0	0
11	5058	1012	74	463	51836	427334	2	0	2	0
18	7705	1387	348	366	65410	597411	0	0	2	0
20	6840	889	127	263	50080	455906	2	0	0	0

3) Aplicación del algoritmo: K-means

Obtención de los centroides



```
CentroidesP = Hipoteca.groupby('clusterP').mean()  
CentroidesP
```

clusterP	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo
0	6358.959184	1117.306122	190.755102	465.653061	50687.081633	497262.265306	0.448980	0.061224	2.122449
1	3472.482143	905.607143	224.732143	536.589286	23957.642857	272010.535714	1.625000	2.250000	6.660714
2	6389.685185	998.851852	190.203704	524.148148	54899.722222	430860.092593	1.462963	2.222222	6.296296
3	3502.930233	857.209302	245.790698	533.627907	24129.139535	291900.953488	0.348837	0.000000	2.093023

4) Interpretación

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo
clusterP									
0	6358.959184	1117.306122	190.755102	465.653061	50687.081633	497262.265306	0.448980	0.061224	2.122449
1	3472.482143	905.607143	224.732143	536.589286	23957.642857	272010.535714	1.625000	2.250000	6.660714
2	6389.685185	998.851852	190.203704	524.148148	54899.722222	430860.092593	1.462963	2.222222	6.296296
3	3502.930233	857.209302	245.790698	533.627907	24129.139535	291900.953488	0.348837	0.000000	2.093023

Clúster 0: Conformado por 49 casos de una evaluación hipotecaria, con un ingreso promedio mensual de 6358 USD, con gastos comunes de 1117 USD, otros gastos de 465 USD y un pago mensual de coche de 190 USD. Estos gastos en promedio representan menos de la tercera parte del salario mensual (1772 USD). Por otro lado, este grupo de usuarios tienen un ahorro promedio de 50687 USD, y un valor promedio de vivienda (a comprar o hipotecar) de 497262 USD. Además, en su mayoría son solteros (0-soltero), casi sin hijos menores y tienen un tipo de trabajo, en su mayoría, asalariado (2-asalariado).

```
#Cantidad de elementos en los clusters
Hipoteca.groupby(['clusterP'])['clusterP'].count()

clusterP
0    49
1    56
2    54
3    43
Name: clusterP, dtype: int64
```

Práctica

4) Interpretación

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo
clusterP									
0	6358.959184	1117.306122	190.755102	465.653061	50687.081633	497262.265306	0.448980	0.061224	2.122449
1	3472.482143	905.607143	224.732143	536.589286	23957.642857	272010.535714	1.625000	2.250000	6.660714
2	6389.685185	998.851852	190.203704	524.148148	54899.722222	430860.092593	1.462963	2.222222	6.296296
3	3502.930233	857.209302	245.790698	533.627907	24129.139535	291900.953488	0.348837	0.000000	2.093023

Clúster 3: Es un segmento de clientes conformado 43 usuarios, con un ingreso promedio mensual de 3502 USD, con gastos comunes de 857 USD, otros gastos de 533 USD y un pago mensual de coche de 245 USD. Estos gastos en promedio representan casi la mitad del salario mensual (1635 USD). Por otro lado, este grupo de usuarios tienen un ahorro promedio de 24129 USD, y un valor promedio de vivienda (a comprar o hipotecar) de 291900 USD. Además, en su mayoría son solteros (0-soltero), sin hijos y tienen un tipo de trabajo asalariado (2-asalariado).

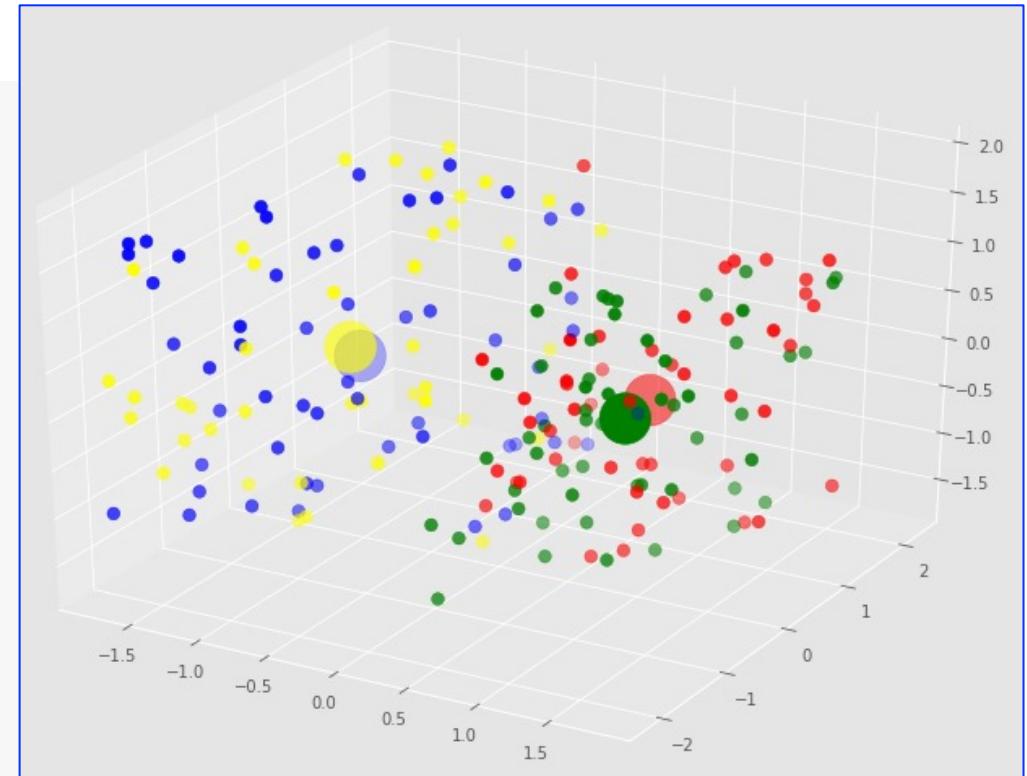
```
#Cantidad de elementos en los clusters
Hipoteca.groupby(['clusterP'])['clusterP'].count()

clusterP
0    49
1    56
2    54
3    43
Name: clusterP, dtype: int64
```

4) Interpretación

```
# Gráfica de los elementos y los centros de los clusters
from mpl_toolkits.mplot3d import Axes3D
plt.rcParams['figure.figsize'] = (10, 7)
plt.style.use('ggplot')
colores=['red', 'blue', 'green', 'yellow']
asignar=[]
for row in MParticional.labels_:
    asignar.append(colores[row])

fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(MEstandarizada[:, 0],
           MEstandarizada[:, 1],
           MEstandarizada[:, 2], marker='o', c=asignar, s=60)
ax.scatter(MParticional.cluster_centers_[:, 0],
           MParticional.cluster_centers_[:, 1],
           MParticional.cluster_centers_[:, 2], marker='o', c=colores, s=1000)
plt.show()
```





Universidad Nacional Autónoma de México
Facultad de Ingeniería

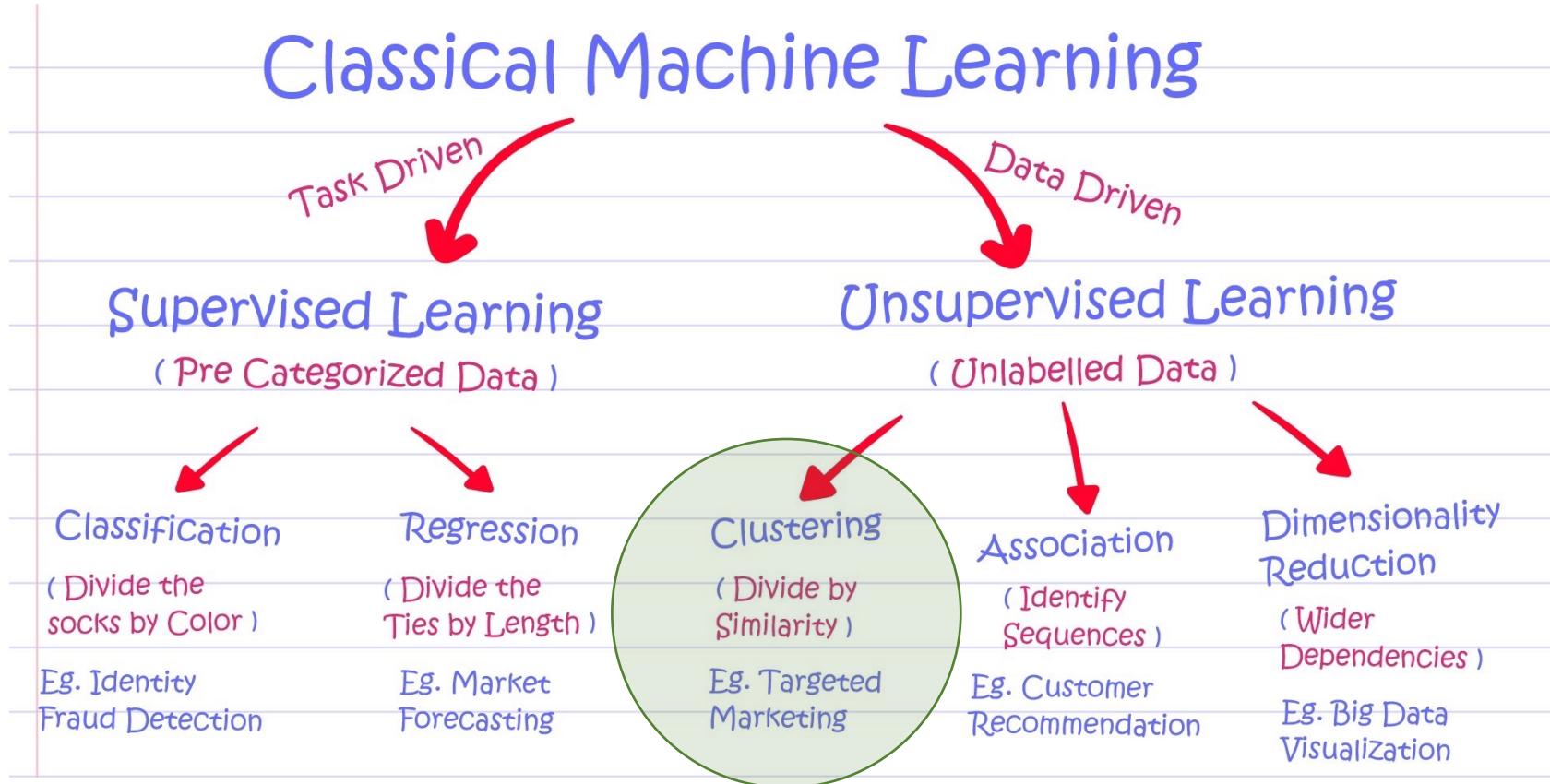
Clustering: Jerárquico y Particional Práctica 6

Guillermo Molero-Castillo

guillermo.molero@ingenieria.unam.edu

Octubre, 2021

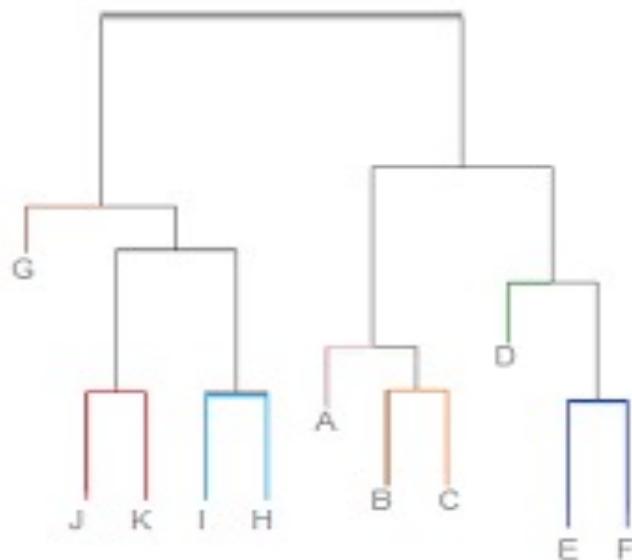
Classical Machine Learning



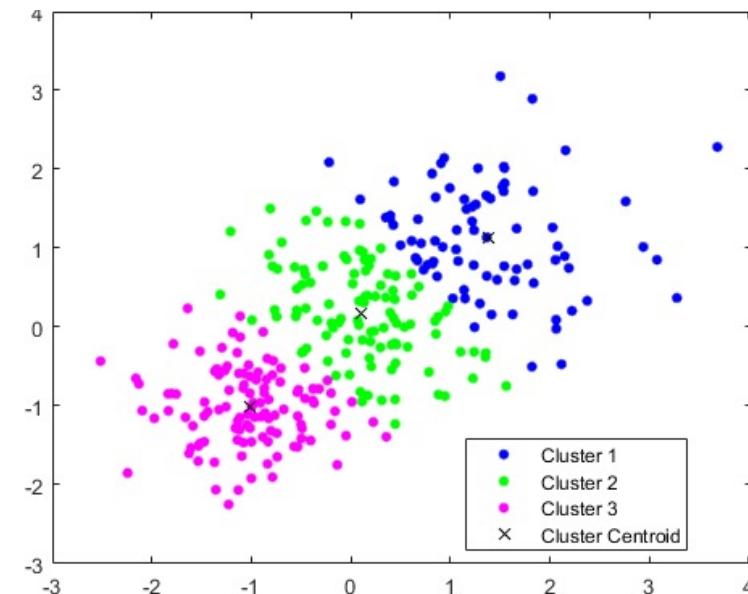
Objetivo

Obtener grupos de pacientes con características similares, diagnosticadas con un tumor de mama, a través de clustering jerárquico y particional.

Jerárquico



Particional



Fuente de datos

Estudios clínicos a partir de imágenes digitalizadas de pacientes con cáncer de mama de Wisconsin (WDBC, Wisconsin Diagnostic Breast Cancer).

Variable	Descripción	Tipo
ID number	Identifica al paciente	Discreto
Diagnosis	Diagnóstico (M=maligno, B=benigno)	Booleano
Radius	Media de las distancias del centro y puntos del perímetro	Continuo
Texture	Desviación estándar de la escala de grises	Continuo
Perimeter	Valor del perímetro del cáncer de mama	Continuo
Area	Valor del área del cáncer de mama	Continuo
Smoothness	Variación de la longitud del radio	Continuo
Compactness	Perímetro \wedge 2 /Area - 1	Continuo
Concavity	Caída o gravedad de las curvas de nivel	Continuo
Concave points	Número de sectores de contorno cóncavo	Continuo
Symmetry	Simetría de la imagen	Continuo
Fractal dimension	“Aproximación de frontera” - 1	Continuo

Fuente: [https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))

1. Importar las bibliotecas y los datos

```
▶ import pandas as pd          # Para la manipulación y análisis de datos  
    import numpy as np          # Para crear vectores y matrices n dimensionales  
    import matplotlib.pyplot as plt # Para la generación de gráficas a partir de los datos  
    import seaborn as sns         # Para la visualización de datos basado en matplotlib  
    %matplotlib inline
```

```
▶ from google.colab import files  
files.upload()  
  
#from google.colab import drive  
#drive.mount('/content/drive')
```

1. Importar las bibliotecas y los datos

```
▶ BCancer = pd.read_csv('WDBCOriginal.csv')  
BCancer
```

	IDNumber	Diagnosis	Radius	Texture	Perimeter	Area	Smoothness	Compactness	Concavity	ConcavePoints	Symmetry	FractalDimension
0	P-842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	0.2419	0.07871
1	P-842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	0.1812	0.05667
2	P-84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	0.2069	0.05999
3	P-84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	0.2597	0.09744
4	P-84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	0.1809	0.05883
...
564	P-926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	0.1726	0.05623
565	P-926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	0.1752	0.05533
566	P-926954	M	16.60	28.08	108.30	1035.0	0.12100	0.13200	0.15600	0.10800	0.1590	0.05648
567	P-927241	M	20.60	29.33	140.10	1354.0	0.10400	0.11800	0.15800	0.09500	0.2397	0.07016
568	P-92751	B	7.76	24.54	47.92	185.0	0.07870	0.08450	0.09620	0.04920	0.1587	0.05884

569 rows × 12 columns

```
▶ print(BCancer.groupby('Diagnosis').size())  
Diagnosis  
B    357  
M    212  
dtype: int64
```

1. Importar las bibliotecas y los datos



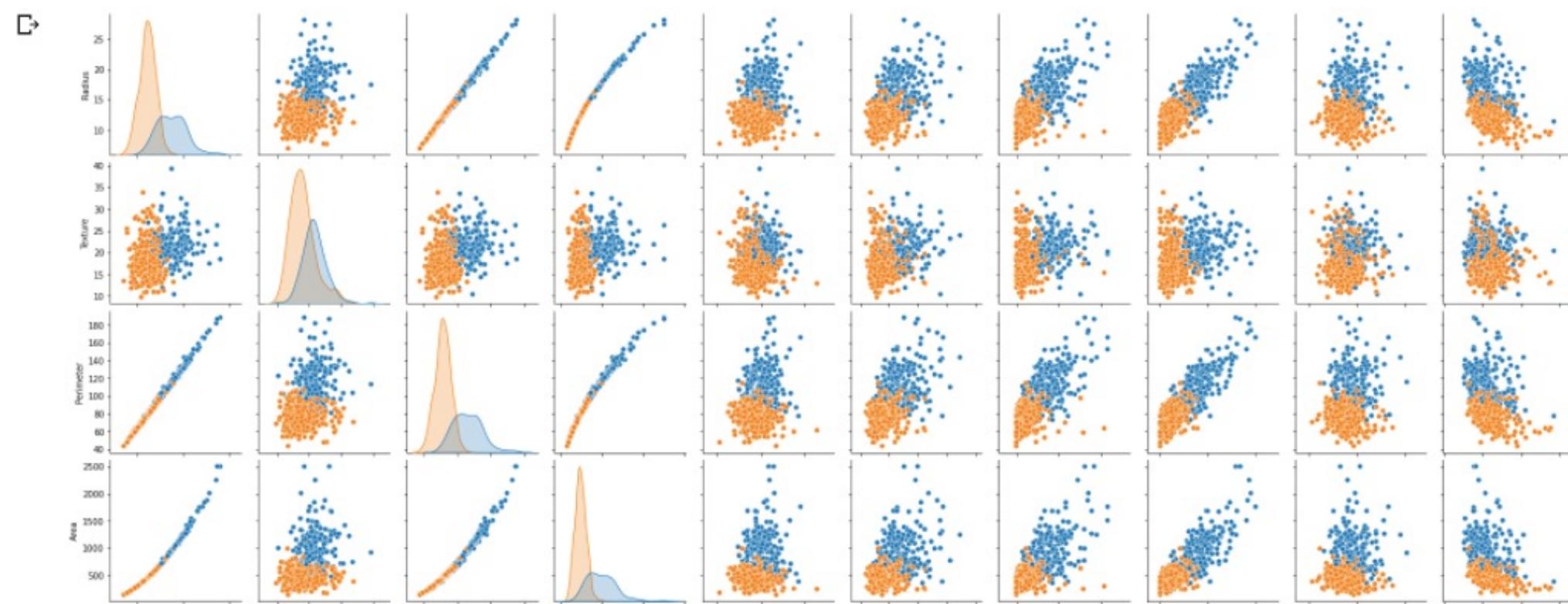
```
BCancer.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 12 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   IDNumber          569 non-null    object  
 1   Diagnosis         569 non-null    object  
 2   Radius             569 non-null    float64 
 3   Texture            569 non-null    float64 
 4   Perimeter          569 non-null    float64 
 5   Area               569 non-null    float64 
 6   Smoothness         569 non-null    float64 
 7   Compactness        569 non-null    float64 
 8   Concavity          569 non-null    float64 
 9   ConcavePoints      569 non-null    float64 
 10  Symmetry           569 non-null    float64 
 11  FractalDimension  569 non-null    float64 
dtypes: float64(10), object(2)
memory usage: 53.5+ KB
```

2. Selección de características

Evaluación visual

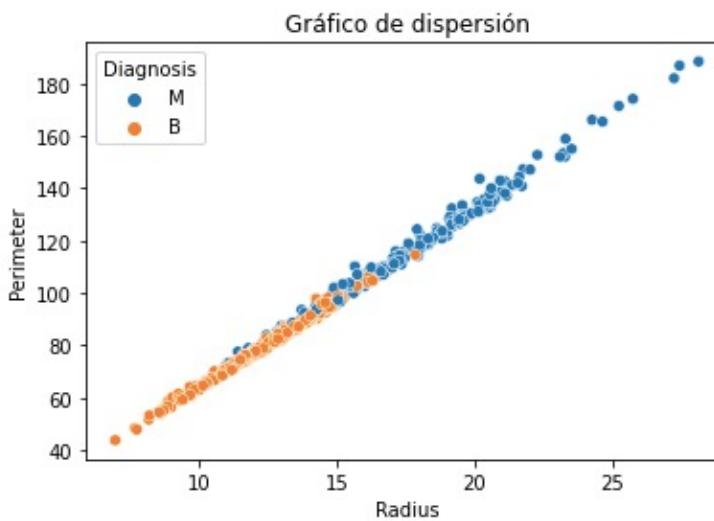
```
▶ sns.pairplot(BCancer, hue='Diagnosis')  
plt.show()
```



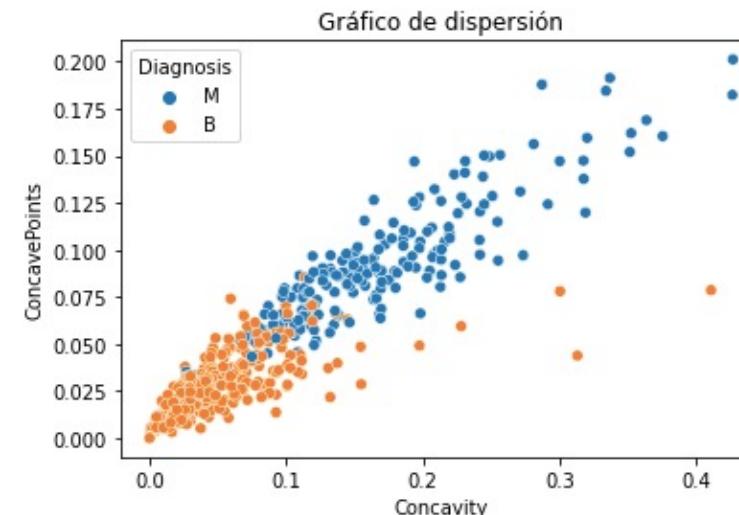
2. Selección de características

Evaluación visual


```
sns.scatterplot(x='Radius', y ='Perimeter', data=BCancer, hue='Diagnosis')
plt.title('Gráfico de dispersión')
plt.xlabel('Radius')
plt.ylabel('Perimeter')
plt.show()
```



```
sns.scatterplot(x='Concavity', y ='ConcavePoints', data=BCancer, hue='Diagnosis')
plt.title('Gráfico de dispersión')
plt.xlabel('Concavity')
plt.ylabel('ConcavePoints')
plt.show()
```



2. Selección de características

Matriz de correlaciones

```
▶ CorrBCancer = BCancer.corr(method='pearson')
CorrBCancer
```

	Radius	Texture	Perimeter	Area	Smoothness	Compactness	Concavity	ConcavePoints	Symmetry	FractalDimension
Radius	1.000000	0.323782	0.997855	0.987357	0.170581	0.506124	0.676764	0.822529	0.147741	-0.311631
Texture	0.323782	1.000000	0.329533	0.321086	-0.023389	0.236702	0.302418	0.293464	0.071401	-0.076437
Perimeter	0.997855	0.329533	1.000000	0.986507	0.207278	0.556936	0.716136	0.850977	0.183027	-0.261477
Area	0.987357	0.321086	0.986507	1.000000	0.177028	0.498502	0.685983	0.823269	0.151293	-0.283110
Smoothness	0.170581	-0.023389	0.207278	0.177028	1.000000	0.659123	0.521984	0.553695	0.557775	0.584792
Compactness	0.506124	0.236702	0.556936	0.498502	0.659123	1.000000	0.883121	0.831135	0.602641	0.565369
Concavity	0.676764	0.302418	0.716136	0.685983	0.521984	0.883121	1.000000	0.921391	0.500667	0.336783
ConcavePoints	0.822529	0.293464	0.850977	0.823269	0.553695	0.831135	0.921391	1.000000	0.462497	0.166917
Symmetry	0.147741	0.071401	0.183027	0.151293	0.557775	0.602641	0.500667	0.462497	1.000000	0.479921
FractalDimension	-0.311631	-0.076437	-0.261477	-0.283110	0.584792	0.565369	0.336783	0.166917	0.479921	1.000000

2. Selección de características

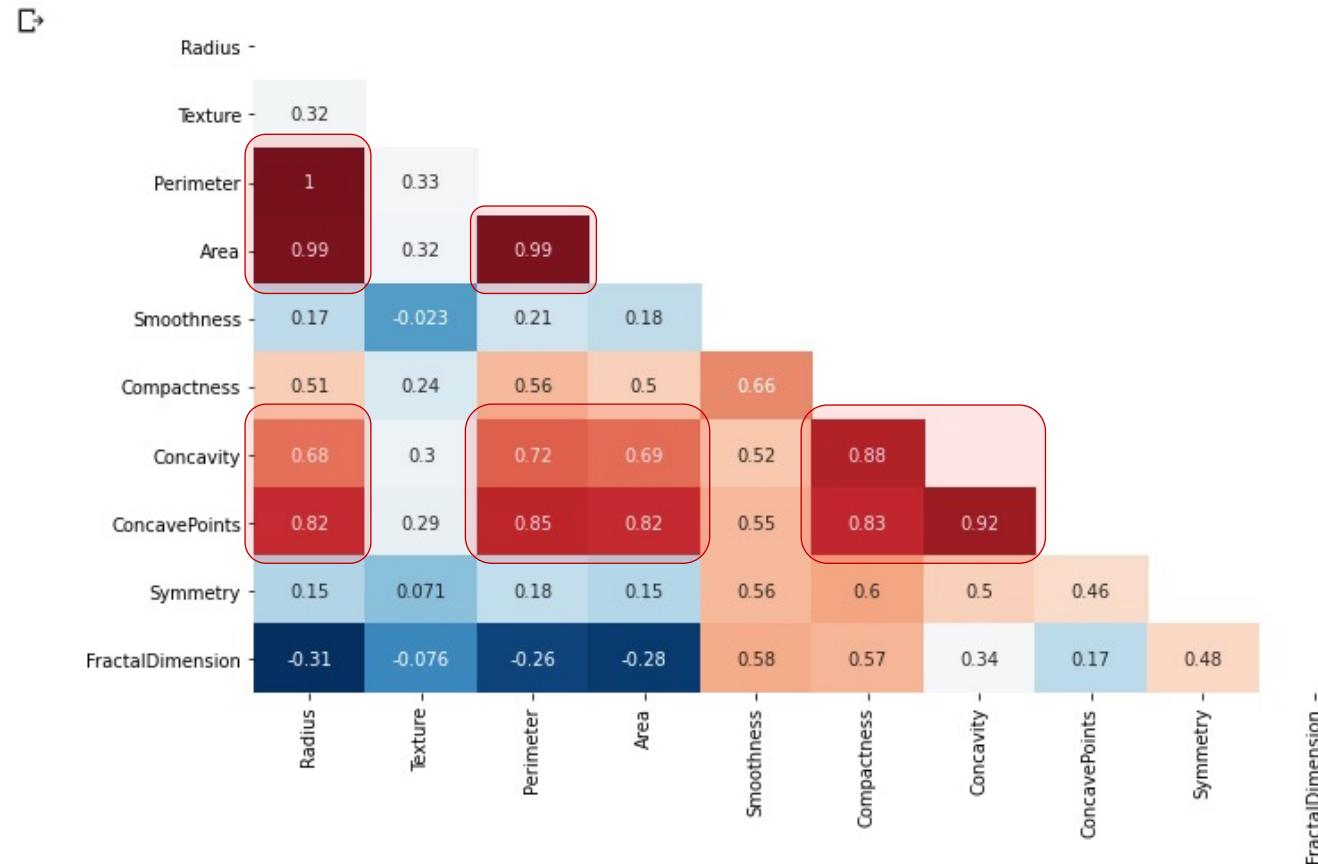
Matriz de correlaciones

```
▶ print(CorrBCancer['Radius'].sort_values(ascending=False)[:10], '\n') #Top 10 valores  
Radius          1.000000  
Perimeter      0.997855  
Area            0.987357  
ConcavePoints   0.822529  
Concavity       0.676764  
Compactness     0.506124  
Texture          0.323782  
Smoothness       0.170581  
Symmetry         0.147741  
FractalDimension -0.311631  
Name: Radius, dtype: float64
```

Práctica

2. Selección de características

```
▶ plt.figure(figsize=(14,7))
MatrizInf = np.triu(CorrBCancer)
sns.heatmap(CorrBCancer, cmap='RdBu_r', annot=True, mask=MatrizInf)
plt.show()
```



Variables seleccionadas:

- 1) Textura [Posición 3]
- 2) Área [Posición 5]
- 3) Smoothness [Pos. 6]
- 4) Compactness [Pos. 7]
- 5) Symmetry [Posición 10]
- 6) FractalDimension [Pos. 11]

2. Selección de características

Elección de variables

```
▶ MatrizVariables = np.array(BCancer[['Texture', 'Area', 'Smoothness', 'Compactness', 'Symmetry', 'FractalDimension']])
pd.DataFrame(MatrizVariables)
#MatrizVariables = BCancer.iloc[:, [3, 5, 6, 7, 10, 11]].values #iloc para seleccionar filas y columnas
```

	0	1	2	3	4	5
0	10.38	1001.0	0.11840	0.27760	0.2419	0.07871
1	17.77	1326.0	0.08474	0.07864	0.1812	0.05667
2	21.25	1203.0	0.10960	0.15990	0.2069	0.05999
3	20.38	386.1	0.14250	0.28390	0.2597	0.09744
4	14.34	1297.0	0.10030	0.13280	0.1809	0.05883
...
564	22.39	1479.0	0.11100	0.11590	0.1726	0.05623
565	28.25	1261.0	0.09780	0.10340	0.1752	0.05533
566	28.08	858.1	0.08455	0.10230	0.1590	0.05648
567	29.33	1265.0	0.11780	0.27700	0.2397	0.07016
568	24.54	181.0	0.05263	0.04362	0.1587	0.05884
569 rows × 6 columns						

3. Estandarización de datos

```
▶ from sklearn.preprocessing import StandardScaler, MinMaxScaler  
estandarizar = StandardScaler()  
MEstandarizada = estandarizar.fit_transform(MatrizVariables)      # Se instancia el objeto StandardScaler o MinMaxScaler  
pd.DataFrame(MEstandarizada)                                     # Sescalán los datos
```

	0	1	2	3	4	5
0	-2.073335	0.984375	1.568466	3.283515	2.217515	2.255747
1	-0.353632	1.908708	-0.826962	-0.487072	0.001392	-0.868652
2	0.456187	1.558884	0.942210	1.052926	0.939685	-0.398008
3	0.253732	-0.764464	3.283553	3.402909	2.867383	4.910919
4	-1.151816	1.826229	0.280372	0.539340	-0.009560	-0.562450
...
564	0.721473	2.343856	1.041842	0.219060	-0.312589	-0.931027
565	2.085134	1.723842	0.102458	-0.017833	-0.217664	-1.058611
566	2.045574	0.577953	-0.840484	-0.038680	-0.809117	-0.895587
567	2.336457	1.735218	1.525767	3.272144	2.137194	1.043695
568	1.221792	-1.347789	-3.112085	-1.150752	-0.820070	-0.561032

569 rows × 6 columns

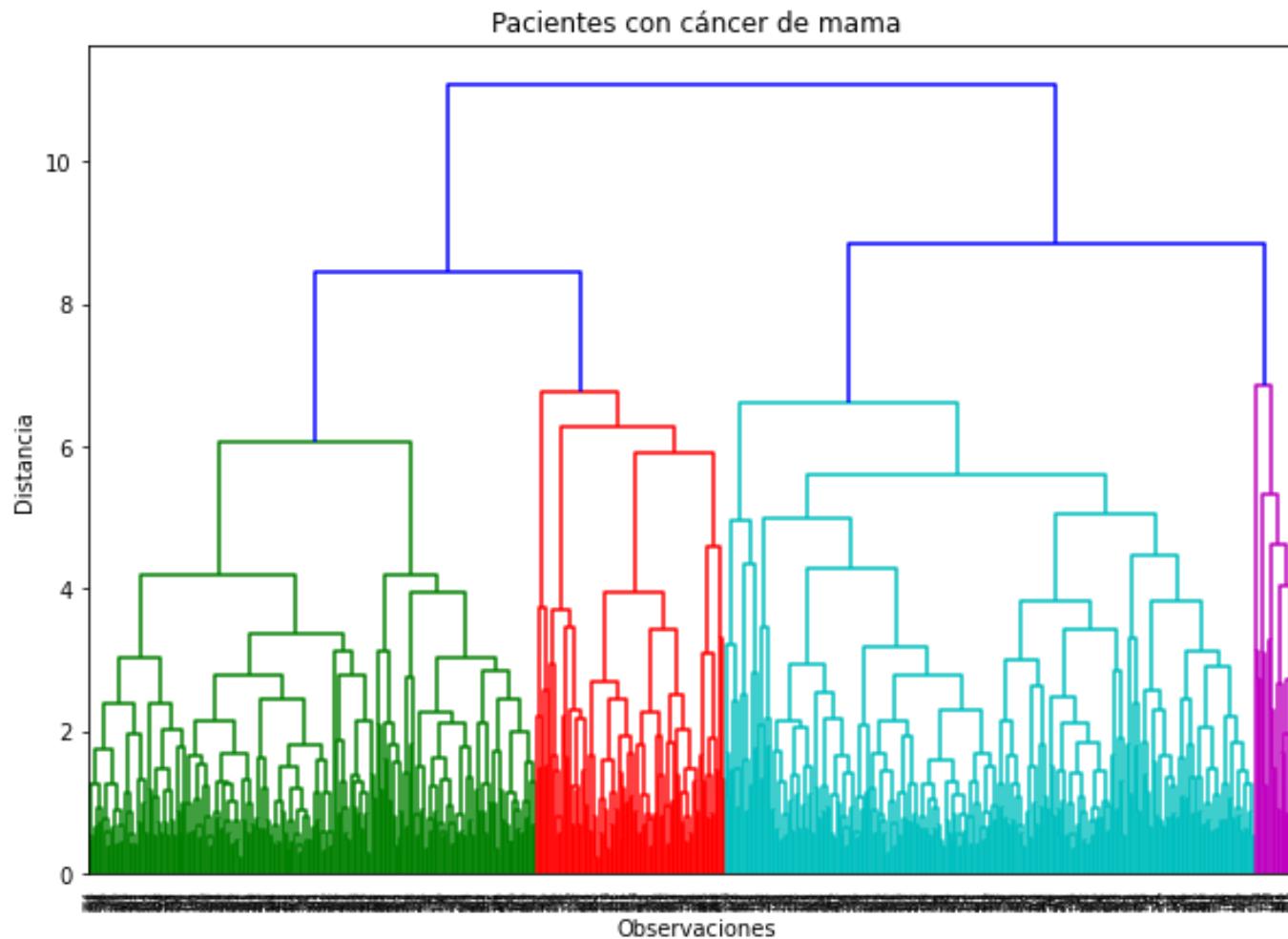
Clustering Jerárquico

Algoritmo: Ascendente Jerárquico

4. Algoritmo: Ascendente Jerárquico

```
▶ #Se importan las bibliotecas de clustering jerárquico para crear el árbol
import scipy.cluster.hierarchy as shc
from sklearn.cluster import AgglomerativeClustering
plt.figure(figsize=(10, 7))
plt.title("Pacientes con cáncer de mama")
plt.xlabel('Observaciones')
plt.ylabel('Distancia')
Arbol = shc.dendrogram(shc.linkage(MEstandarizada, method='complete', metric='euclidean'))
#plt.axhline(y=7, color='orange', linestyle='--')
#Probar con otras mediciones de distancia (euclidean, chebyshev, cityblock)
```

4. Algoritmo: Ascendente Jerárquico



4. Algoritmo: Ascendente Jerárquico

Se crean las etiquetas en los clústeres



#Se crean las etiquetas de los elementos en los clusters

```
MJerarquico = AgglomerativeClustering(n_clusters=4, linkage='complete', affinity='euclidean')  
MJerarquico.fit_predict(MEstandarizada)  
MJerarquico.labels_
```

```
array([0, 1, 1, 0, 1, 2, 1, 2, 0, 0, 3, 2, 1, 3, 0, 2, 3, 2, 1, 2, 2, 2,  
0, 1, 2, 0, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2, 2, 3, 3, 2, 3, 2, 1, 2,  
2, 2, 3, 2, 2, 3, 3, 3, 2, 3, 3, 1, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2,  
2, 3, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 0, 2, 2, 2, 1, 1, 2, 2, 2, 2,  
3, 2, 3, 3, 3, 2, 2, 2, 2, 2, 3, 2, 3, 2, 3, 2, 2, 2, 2, 2, 3, 0, 3,  
2, 2, 2, 2, 2, 3, 2, 2, 2, 1, 3, 2, 0, 2, 3, 3, 3, 1, 2, 1, 2, 2,  
2, 2, 1, 3, 3, 2, 2, 3, 2, 2, 3, 2, 0, 3, 2, 3, 2, 2, 2, 2, 2, 2,  
2, 3, 2, 1, 3, 3, 2, 1, 2, 3, 1, 3, 3, 1, 1, 2, 2, 2, 3, 2, 2, 2, 3,  
2, 2, 3, 3, 1, 0, 1, 3, 3, 3, 1, 2, 2, 3, 0, 3, 3, 2, 2, 2, 3, 2, 1,  
1, 2, 2, 1, 1, 0, 3, 3, 2, 1, 2, 3, 1, 3, 1, 1, 2, 2, 2, 3, 3, 2, 1,  
3, 2, 2, 2, 3, 2, 2, 2, 3, 2, 2, 3, 1, 3, 3, 1, 1, 3, 1, 2, 3, 2, 3,  
2, 3, 2, 2, 3, 2, 2, 2, 1, 3, 1, 1, 1, 2, 1, 0, 0, 1, 1, 3, 2, 3,  
2, 1, 3, 3, 2, 2, 3, 2, 1, 3, 3, 2, 3, 1, 3, 2, 1, 3, 1, 2, 3, 3, 3,  
3, 3, 2, 3, 2, 2, 3, 2, 3, 3, 2, 3, 1, 3, 1, 2, 3, 3, 3, 2, 3, 3, 3,  
3, 3, 3, 3, 3, 3, 2, 3, 3, 1, 2, 3, 2, 1, 2, 1, 3, 2, 3, 3, 2, 2, 2,  
2, 2, 2, 3, 3, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 2, 3, 0,  
1, 1, 3, 3, 2, 3, 2, 2, 3, 3, 2, 1, 3, 1, 1, 2, 1, 1, 2, 3, 1, 1, 1,  
3, 2, 2, 3, 3, 0, 2, 3, 3, 2, 3, 3, 3, 2, 2, 2, 2, 2, 2, 1, 2, 3, 2,  
2, 3, 3, 3, 0, 3, 3, 2, 3, 2, 2, 3, 2, 3, 3, 2, 2, 3, 2, 3, 2, 2, 2,  
3, 3, 3, 2, 2, 3, 2, 3, 2, 3, 3, 3, 2, 2, 1, 2, 3, 2, 3, 3, 3, 3, 3,  
2, 3, 3, 2, 2, 2, 1, 2, 3, 1, 3, 1, 2, 3, 3, 3, 3, 3, 3, 1, 1,  
3, 3, 3, 3, 3, 3, 2, 2, 3, 3, 3, 2, 2, 3, 3, 2, 2, 3, 3, 3, 3, 2, 2,  
2, 2, 3, 1, 2, 1, 3, 3, 2, 3, 3, 3, 2, 2, 3, 2, 1, 2, 2, 2, 2, 1, 0, 0,  
2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 1, 1, 2, 2, 2, 2, 1, 3, 2, 2, 2, 2, 2, 3,  
2, 2, 2, 2, 2, 2, 1, 2, 0, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,  
3, 2, 3, 3, 3, 3, 2, 3, 3, 1, 3, 1, 1, 1, 1, 3, 0, 3])
```

4. Algoritmo: Ascendente Jerárquico

Se crean las etiquetas en los clústeres

```
BCancer['clusterH'] = MJerarquico.labels_
BCancer
```

	IDNumber	Diagnosis	Radius	Texture	Perimeter	Area	Smoothness	Compactness	Concavity	ConcavePoints	Symmetry	FractalDimension	clusterH
0	P-842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	0.2419	0.07871	0
1	P-842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	0.1812	0.05667	1
2	P-84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	0.2069	0.05999	1
3	P-84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	0.2597	0.09744	0
4	P-84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	0.1809	0.05883	1
...
564	P-926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	0.1726	0.05623	1
565	P-926682	M	20.13	28.25							0.1752	0.05533	1
566	P-926954	M	16.60	28.08							0.1590	0.05648	3
567	P-927241	M	20.60	29.33							0.2397	0.07016	0
568	P-92751	B	7.76	24.54							0.1587	0.05884	3

569 rows × 13 columns

```
#Cantidad de elementos en los clusters
BCancer.groupby(['clusterH'])['clusterH'].count()
```

clusterH	0	23
1	88	
2	248	
3	210	

4. Algoritmo: Ascendente Jerárquico

Se crean las etiquetas en los clústeres

```
BCancer[BCancer.clusterH == 0]
```

	IDNumber	Diagnosis	Radius	Texture	Perimeter	Area	Smoothness	Compactness	Concavity	ConcavePoints	Symmetry	FractalDimension	clusterH
0	P-842302	M	17.990	10.38	122.80	1001.0	0.1184	0.2776	0.30010	0.14710	0.2419	0.07871	0
3	P-84348301	M	11.420	20.38	77.58	386.1	0.1425	0.2839	0.24140	0.10520	0.2597	0.09744	0
8	P-844981	M	13.000	21.82	87.50	519.8	0.1273	0.1932	0.18590	0.09353	0.2350	0.07389	0
9	P-84501001	M	12.460	24.04	83.97	475.9	0.1186	0.2396	0.22730	0.08543	0.2030	0.08243	0
14	P-84667401	M	13.730	22.61	93.60	578.3	0.1131	0.2293	0.21280	0.08025	0.2069	0.07682	0
22	P-8511133	M	15.340	14.26	102.50	704.4	0.1073	0.2135	0.20770	0.09756	0.2521	0.07032	0
25	P-852631	M	17.140	16.40	116.00	912.7	0.1186	0.2276	0.22290	0.14010	0.3040	0.07413	0
78	P-8610862	M	20.180	23.97	143.70	1245.0	0.1286	0.3454	0.37540	0.16040	0.2906	0.08142	0
108	P-86355	M	22.270	19.67	152.80	1509.0	0.1326	0.2768	0.42640	0.18230	0.2556	0.07039	0
122	P-865423	M	24.250	20.20	166.20	1761.0	0.1447	0.2867	0.42680	0.20120	0.2655	0.06877	0
146	P-869691	M	11.800	16.58	78.99	432.0	0.1091	0.1700	0.16590	0.07415	0.2678	0.07371	0
181	P-873593	M	21.090	26.57	142.70	1311.0	0.1141	0.2832	0.24870	0.14960	0.2395	0.07398	0

4. Algoritmo: Ascendente Jerárquico

Obtención de los centroides

```
CentroidesH = BCancer.groupby(['clusterH'])['Texture', 'Area', 'Smoothness', 'Compactness', 'Symmetry', 'FractalDimension'].mean()
CentroidesH
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Indexing with multiple keys (implicitly converted to
""Entry point for launching an IPython kernel.

	Texture	Area	Smoothness	Compactness	Symmetry	FractalDimension
clusterH						
0	20.133478	775.543478	0.124274	0.242200	0.240830	0.077839
1	22.540568	1243.728409	0.098441	0.137140	0.182560	0.058889
2	18.167540	561.336694	0.103316	0.114235	0.190486	0.065737
3	19.160095	505.403810	0.084217	0.063813	0.163030	0.059317

4. Algoritmo: Ascendente Jerárquico

Interpretación

	Texture	Area	Smoothness	Compactness	Symmetry	FractalDimension
clusterH						
0	20.133478	775.543478	0.124274	0.242200	0.240830	0.077839
1	22.540568	1243.728409	0.098441	0.137140	0.182560	0.058889
2	18.167540	561.336694	0.103316	0.114235	0.190486	0.065737
3	19.160095	505.403810	0.084217	0.063813	0.163030	0.059317

Clúster 0: Conformado por 23 pacientes con indicios de cáncer maligno por el tamaño del tumor, con un área promedio de tumor de 775 pixeles y una desviación estándar de textura de 20 pixeles. Aparentemente es un tumor compacto (0.24 pixeles), cuya suavidad alcanza 0.12 pixeles, una simetría de 0.24 y una aproximación de frontera, dimensión fractal, promedio de 0.077 pixeles.

```
#Cantidad de elementos en los clusters
BCancer.groupby(['clusterH'])['clusterH'].count()

clusterH
0    23
1    88
2   248
3   210
```

...

4. Algoritmo: Ascendente Jerárquico

Interpretación

	Texture	Area	Smoothness	Compactness	Symmetry	FractalDimension
clusterH						
0	20.133478	775.543478	0.124274	0.242200	0.240830	0.077839
1	22.540568	1243.728409	0.098441	0.137140	0.182560	0.058889
2	18.167540	561.336694	0.103316	0.114235	0.190486	0.065737
3	19.160095	505.403810	0.084217	0.063813	0.163030	0.059317

Clúster 3: Es un grupo formado por 210 pacientes con el menor tamaño de tumor (posiblemente benigno), con un área promedio de tumor de 505 pixeles y una desviación estándar de textura de 19 pixeles. Es un tumor compacto (0.06 pixeles), cuya suavidad alcanza 0.08 pixeles, una simetría de 0.16 y una aproximación de frontera, dimensión fractal, promedio de 0.059 pixeles.

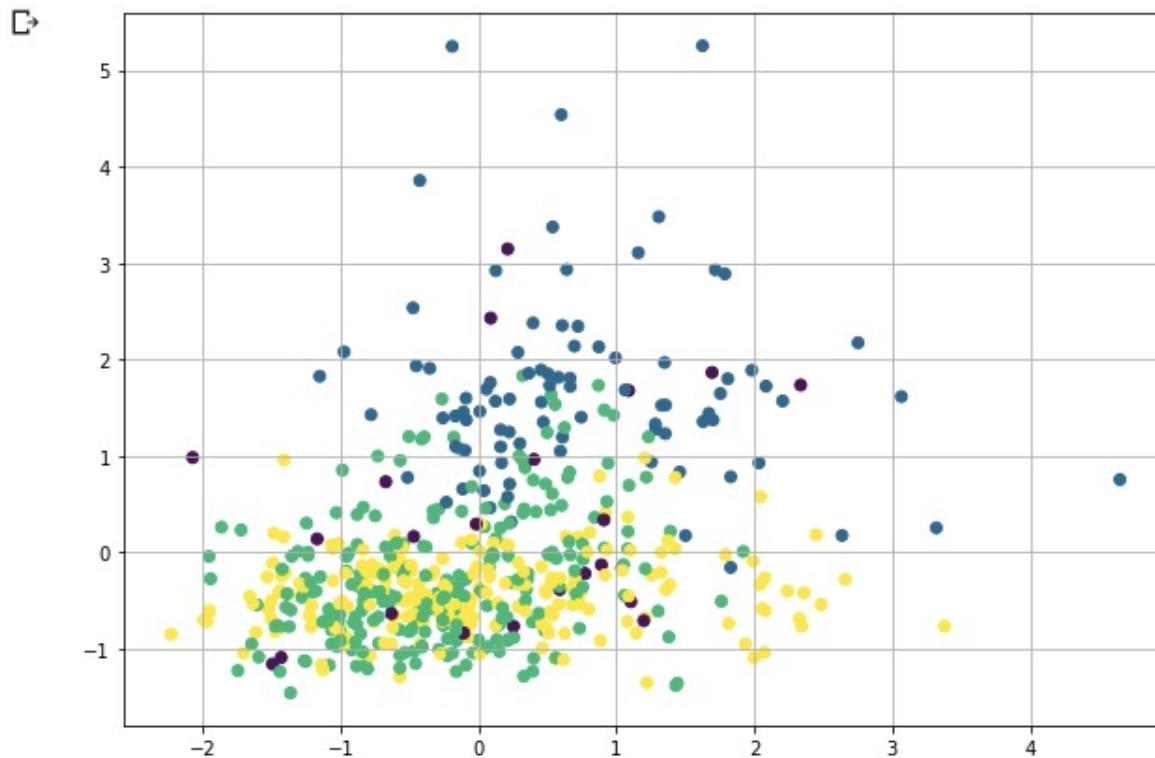
```
#Cantidad de elementos en los clusters
BCancer.groupby(['clusterH'])['clusterH'].count()

clusterH
0    23
1    88
2   248
3   210
```

...

4. Algoritmo: Ascendente Jerárquico

```
plt.figure(figsize=(10, 7))
plt.scatter(MEstandarizada[:,0], MEstandarizada[:,1], c=MJerarquico.labels_)
plt.grid()
plt.show()
```



Clustering Particional

Algoritmo: K-means

5) Algoritmo: K-means

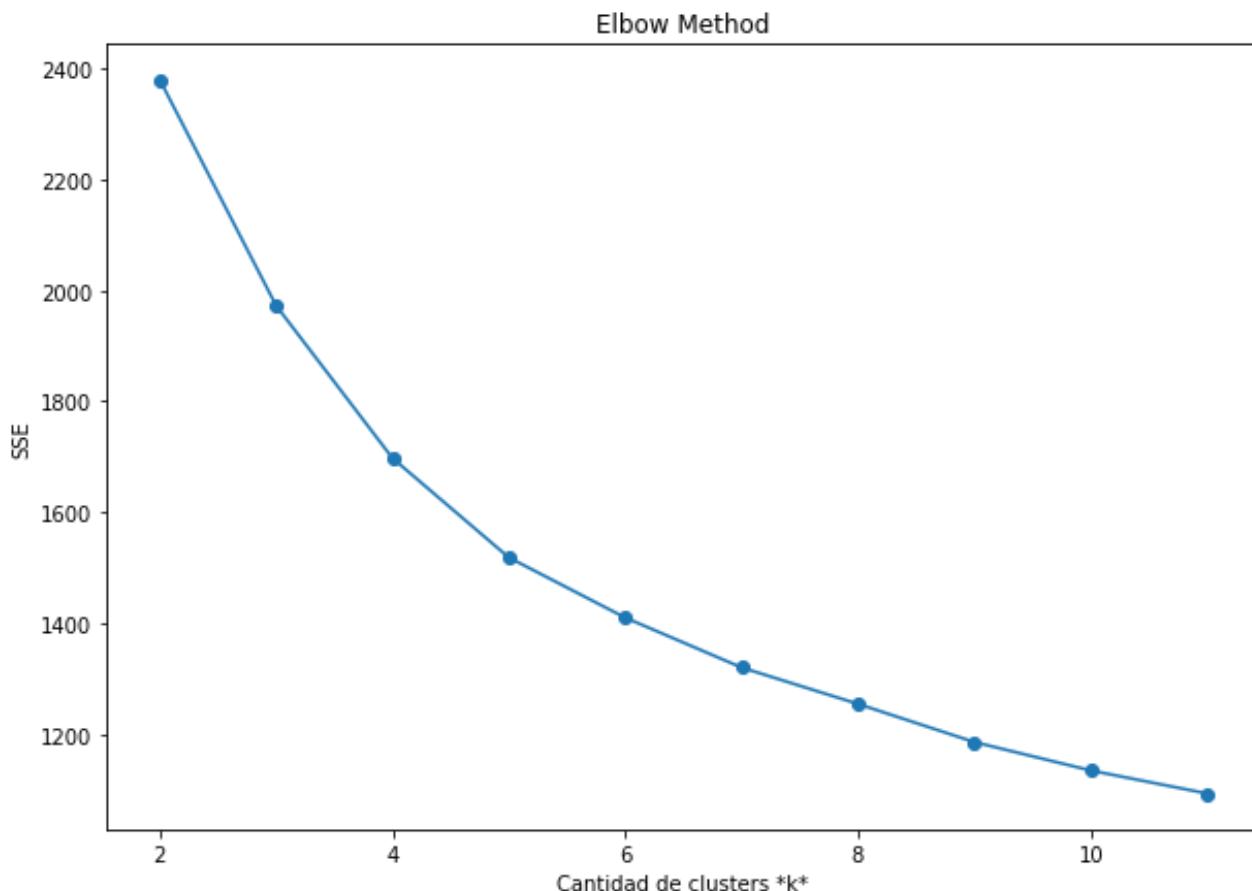
```
▶ #Se importan las bibliotecas
from sklearn.cluster import KMeans
from sklearn.metrics import pairwise_distances_argmin_min

▶ #Definición de k clusters para K-means
#Se utiliza random_state para inicializar el generador interno de números aleatorios
SSE = []
for i in range(2, 12):
    km = KMeans(n_clusters=i, random_state=0)
    km.fit(MEstandarizada)
    SSE.append(km.inertia_)

#Se grafica SSE en función de k
plt.figure(figsize=(10, 7))
plt.plot(range(2, 12), SSE, marker='o')
plt.xlabel('Cantidad de clusters *k*')
plt.ylabel('SSE')
plt.title('Elbow Method')
plt.show()
```

5) Algoritmo: K-means

Método del codo



Observación:

En la práctica, puede que no exista un codo afilado (agudo) y, como método heurístico, ese "codo" no siempre puede identificarse sin ambigüedades.

5) Algoritmo: K-means

Método del codo

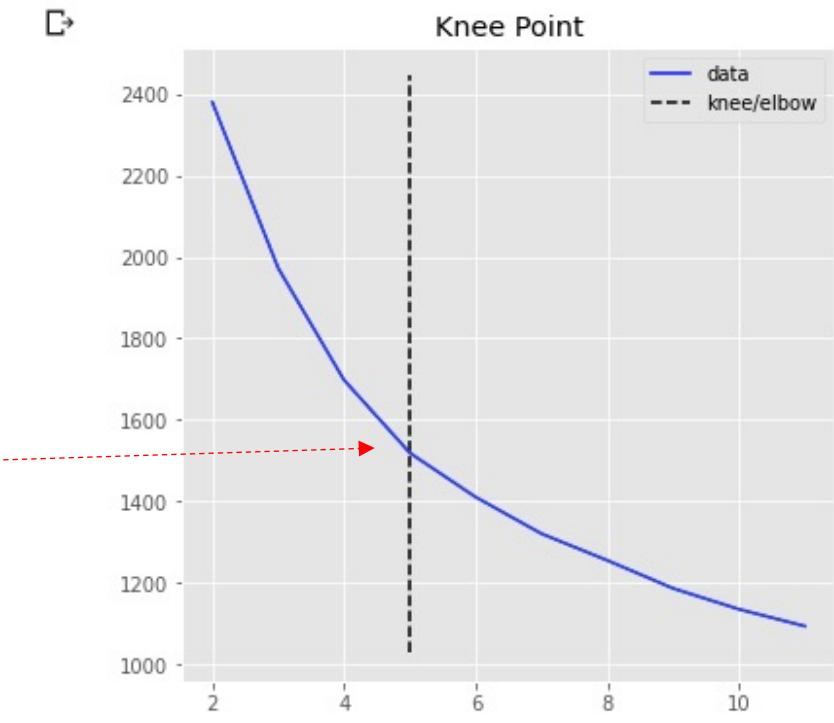
```
▶ !pip install kneed
```

```
▷ Collecting kneed
```

```
  Downloading kneed-0.7.0-py2.py3-none-any.whl (9.4 kB)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from kneed)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from kneed)
Requirement already satisfied: numpy>=1.14.2 in /usr/local/lib/python3.7/dist-packages (from kneed)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from kneed)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from kneed)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from kneed)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from kneed)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from cycler>=0.10)
Installing collected packages: kneed
Successfully installed kneed-0.7.0
```

```
▶ from kneed import KneeLocator
kl = KneeLocator(range(2, 12), SSE, curve="convex", direction="decreasing")
kl.elbow
```

```
▶ plt.style.use('ggplot')
kl.plot_knee()
```



5) Algoritmo: K-means

Se crean las etiquetas en los clústeres



```
#Se crean las etiquetas de los elementos en los clusters  
MParticional = KMeans(n_clusters=5, random_state=0).fit(MEstandarizada)  
MParticional.predict(MEstandarizada)  
MParticional.labels_
```

```
[ ] array([2, 1, 1, 2, 1, 2, 1, 2, 2, 2, 4, 3, 2, 4, 2, 2, 2, 0, 2, 1, 0, 3, 3,  
        2, 1, 1, 2, 2, 1, 1, 3, 1, 2, 2, 1, 3, 1, 3, 0, 4, 3, 4, 3, 1, 3,  
        4, 1, 0, 3, 3, 4, 4, 0, 0, 1, 4, 0, 1, 3, 0, 3, 3, 3, 2, 3, 3, 3, 3,  
        3, 0, 2, 0, 1, 2, 1, 3, 0, 0, 3, 2, 2, 3, 3, 3, 1, 1, 3, 1, 4, 1,  
        4, 3, 4, 4, 0, 0, 3, 1, 3, 3, 0, 3, 4, 3, 0, 3, 3, 2, 3, 0, 2, 4,  
        3, 3, 2, 3, 3, 4, 3, 2, 2, 1, 0, 1, 2, 3, 0, 0, 4, 1, 3, 1, 3, 3,  
        1, 0, 1, 4, 0, 0, 3, 3, 0, 3, 3, 0, 0, 3, 2, 0, 3, 0, 3, 2, 2, 0,  
        0, 0, 1, 0, 0, 0, 3, 1, 1, 3, 1, 0, 0, 0, 1, 0, 3, 0, 3, 0, 0, 0,  
        3, 1, 4, 0, 1, 2, 4, 0, 4, 0, 0, 0, 0, 2, 4, 0, 3, 3, 0, 3, 4,  
        1, 3, 3, 1, 1, 2, 3, 0, 3, 1, 3, 0, 1, 0, 1, 4, 3, 3, 3, 0, 1, 4,  
        0, 3, 3, 3, 0, 0, 3, 0, 4, 2, 1, 4, 4, 1, 0, 4, 1, 1, 4, 4, 0, 0,  
        3, 4, 1, 3, 0, 0, 4, 3, 1, 0, 1, 1, 1, 3, 1, 2, 2, 1, 1, 4, 1, 0,  
        1, 1, 3, 4, 0, 3, 0, 0, 1, 0, 4, 3, 0, 0, 0, 3, 1, 0, 1, 3, 0, 0,  
        4, 0, 3, 0, 3, 0, 3, 0, 0, 0, 0, 0, 4, 1, 0, 2, 3, 0, 4, 0, 0, 0,  
        0, 0, 0, 0, 0, 3, 0, 0, 1, 2, 0, 3, 1, 3, 2, 0, 3, 0, 0, 3, 3,  
        3, 3, 3, 0, 0, 1, 3, 1, 3, 1, 3, 3, 3, 1, 3, 3, 0, 0, 0, 3, 0, 2,  
        1, 4, 0, 0, 3, 0, 0, 3, 0, 4, 3, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1,  
        0, 3, 2, 4, 0, 2, 3, 0, 4, 3, 0, 4, 0, 0, 3, 1, 3, 3, 3, 1, 3, 0,  
        3, 0, 0, 0, 2, 0, 0, 3, 0, 3, 0, 4, 1, 0, 0, 3, 4, 4, 4, 3, 3, 2,  
        0, 4, 0, 2, 3, 3, 3, 4, 3, 4, 0, 0, 2, 3, 1, 1, 0, 3, 3, 0, 0, 0,  
        0, 4, 0, 0, 1, 3, 1, 0, 0, 1, 4, 1, 4, 3, 0, 4, 4, 4, 4, 4, 1, 1,  
        4, 0, 0, 4, 4, 0, 1, 3, 3, 4, 0, 4, 3, 0, 4, 0, 3, 2, 0, 0, 3, 0,  
        3, 3, 0, 1, 3, 4, 4, 0, 1, 0, 4, 0, 3, 0, 1, 1, 3, 2, 3, 1, 2, 2,  
        3, 3, 0, 2, 0, 0, 2, 0, 0, 3, 1, 1, 3, 3, 2, 1, 0, 3, 0, 3, 3, 0,  
        3, 3, 3, 0, 1, 3, 1, 3, 2, 4, 3, 3, 4, 4, 4, 4, 3, 4, 0, 0, 0, 4,  
        4, 3, 4, 4, 4, 4, 3, 4, 4, 4, 4, 2, 2, 1, 1, 4, 2, 4],  
        dtype=int32)
```

5) Algoritmo: K-means

Se crean las etiquetas en los clústeres

```
BCancer['clusterP'] = MParticional.labels_
BCancer
```

	IDNumber	Diagnosis	Radius	Texture	Perimeter	Area	Smoothness	Compactness	Concavity	ConcavePoints	Symmetry	FractalDimension	clusterH	clusterP	
0	P-842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	0.2419	0.07871	0	2	
1	P-842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	0.1812	0.05667	1	1	
2	P-84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	0.2069	0.05999	1	1	
3	P-84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	0.2597	0.09744	0	2	
4	P-84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	0.1809	0.05883	1	1	
...	
564	P-926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	0.1726	0.05623	1	1	
565	P-926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	0.1752	0.05533	1	1	
566	P-926954	M	16.60	28.08	BCancer.groupby(['clusterP'])['clusterP'].count()										
567	P-927241	M	20.60	29.33	clusterP										
568	P-92751	B	7.76	24.54	0 172 1 100 2 56 3 156 4 85										

569 rows x 14 columns

5) Algoritmo: K-means

Se crean las etiquetas en los clústeres

	IDNumber	Diagnosis	Radius	Texture	Perimeter	Area	Smoothness	Compactness	Concavity	ConcavePoints	Symmetry	FractalDimension	clusterH	clusterP
16	P-848406	M	14.680	20.13	94.74	684.5	0.09867	0.07200	0.07395	0.052590	0.1586	0.05922	3	0
19	P-8510426	B	13.540	14.36	87.46	566.3	0.09779	0.08129	0.06664	0.047810	0.1885	0.05766	2	0
37	P-854941	B	13.030	18.42	82.61	523.8	0.08983	0.03766	0.02562	0.029230	0.1467	0.05863	3	0
46	P-85713702	B	8.196	16.84	51.71	201.9	0.08600	0.05943	0.01588	0.005917	0.1769	0.06503	3	0
51	P-857373	B	13.640	16.34	87.21	571.8	0.07685	0.06059	0.01857	0.017230	0.1353	0.05953	3	0
...
527	P-91813702	B	12.340	12.27	78.94	468.5	0.09003	0.06307	0.02958	0.026470	0.1689	0.05808	3	0
532	P-91903902	B	13.680	16.33	87.76	575.5	0.09277	0.07255	0.01752	0.018800	0.1631	0.06155	2	0
546	P-922577	B	10.320	16.35	65.31	324.9	0.09434	0.04994	0.01012	0.005495	0.1885	0.06201	3	0
547	P-922840	B	10.260	16.58	65.85	320.8	0.08877	0.08066	0.04358	0.024380	0.1669	0.06714	3	0
548	P-923169	B	9.683	19.34	61.05	285.7	0.08491	0.05030	0.02337	0.009615	0.1580	0.06235	3	0

172 rows x 14 columns

5) Algoritmo: K-means

Obtención de los centroides

```
▶ CentroidesP = BCancer.groupby(['clusterP'])[['Texture', 'Area', 'Smoothness', 'Compactness', 'Symmetry', 'FractalDimension']].mean()  
CentroidesP  
↳ /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Indexing with multiple keys (implicitly converted to  
    """Entry point for launching an IPython kernel.  
  
          Texture      Area  Smoothness  Compactness   Symmetry  FractalDimension  
clusterP  
0      16.297442  514.286628     0.085941     0.062736    0.164908     0.059056  
1      21.837500 1228.067000     0.100036     0.140695    0.187407     0.059186  
2      20.364643  705.283929     0.115617     0.204721    0.226070     0.075936  
3      17.734615  476.337179     0.104744     0.107066    0.188042     0.066356  
4      24.492706  559.569412     0.085045     0.074626    0.164491     0.059430
```

5) Algoritmo: K-means

Interpretación

	Texture	Area	Smoothness	Compactness	Symmetry	FractalDimension
clusterP						
0	16.297442	514.286628	0.085941	0.062736	0.164908	0.059056
1	21.837500	1228.067000	0.100036	0.140695	0.187407	0.059186
2	20.364643	705.283929	0.115617	0.204721	0.226070	0.075936
3	17.734615	476.337179	0.104744	0.107066	0.188042	0.066356
4	24.492706	559.569412	0.085045	0.074626	0.164491	0.059430

Clúster 0: Conformado por 172 pacientes con alta probabilidad de tener un tumor benigno (por su tamaño), con un área promedio de tumor de 514 píxeles y una desviación estándar de textura de 16 píxeles. Aparentemente es un tumor compacto (0.06 píxeles), cuya suavidad alcanza 0.08 píxeles, una simetría de 0.16 y una aproximación de frontera, dimensión fractal, promedio de 0.059 píxeles.

...

```
BCancer.groupby(['clusterP'])['clusterP'].count()

clusterP
0    172
1    100
2     56
3    156
4     85
```

5) Algoritmo: K-means

Interpretación

	Texture	Area	Smoothness	Compactness	Symmetry	FractalDimension
clusterP						
0	16.297442	514.286628	0.085941	0.062736	0.164908	0.059056
1	21.837500	1228.067000	0.100036	0.140695	0.187407	0.059186
2	20.364643	705.283929	0.115617	0.204721	0.226070	0.075936
3	17.734615	476.337179	0.104744	0.107066	0.188042	0.066356
4	24.492706	559.569412	0.085045	0.074626	0.164491	0.059430

Clúster 4: Es un grupo formado por 85 pacientes con un menor tamaño de tumor (potencialmente benigno), con un área promedio de tumor de 559 pixeles y una desviación estándar de textura de 24 pixeles. Es un tumor compacto (0.07 pixeles), cuya suavidad alcanza 0.08 pixeles, una simetría de 0.16 y una aproximación de frontera, dimensión fractal, promedio de 0.059 pixeles.

```

▶ BCancer.groupby(['clusterP'])['clusterP'].count()

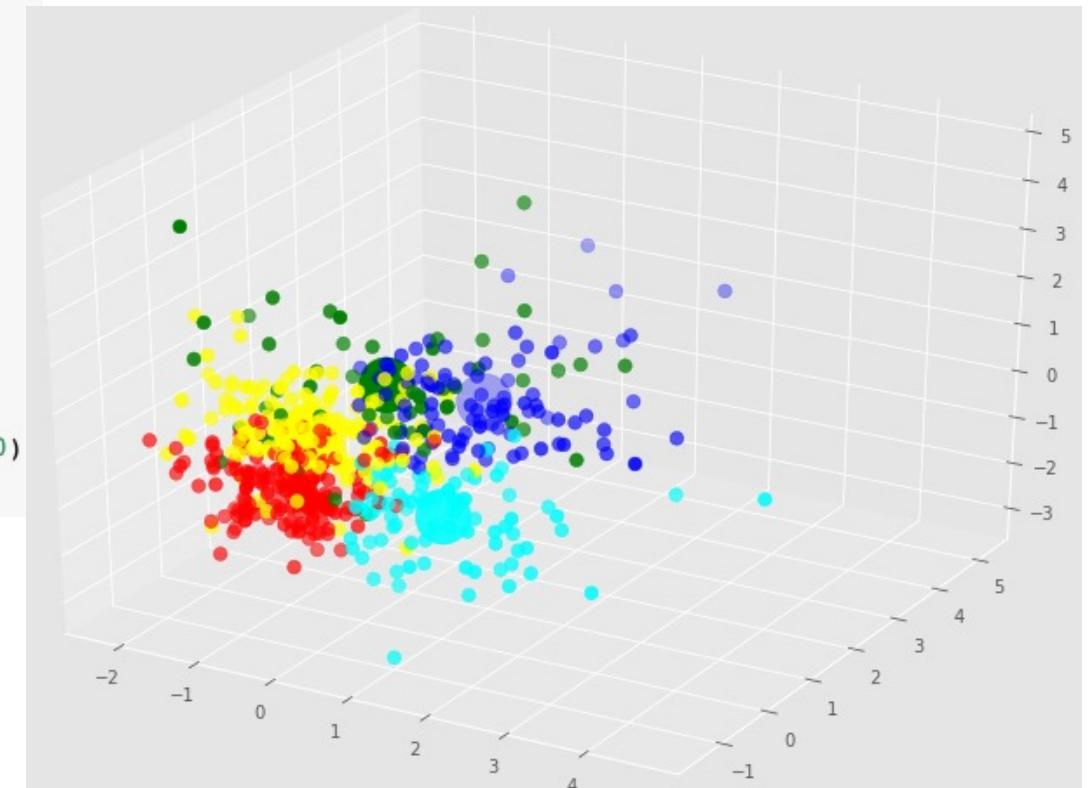
clusterP
0    172
1    100
2     56
3    156
4     85

```

5) Algoritmo: K-means

```
# Gráfica de los elementos y los centros de los clusters
from mpl_toolkits.mplot3d import Axes3D
plt.rcParams['figure.figsize'] = (10, 7)
plt.style.use('ggplot')
colores=['red', 'blue', 'green', 'yellow', 'cyan']
asignar=[]
for row in MParticional.labels_:
    asignar.append(colores[row])

fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(MEstandarizada[:, 0],
           MEstandarizada[:, 1],
           MEstandarizada[:, 2], marker='o', c=asignar, s=60)
ax.scatter(MParticional.cluster_centers_[:, 0],
           MParticional.cluster_centers_[:, 1],
           MParticional.cluster_centers_[:, 2], marker='o', c=colores, s=1000)
plt.show()
```



Consideraciones finales

- Aumentar la cantidad de **clusters** mejorará naturalmente el ajuste (se hará una mejor explicación de la variación). Sin embargo, se puede caer en un sobreajuste, ya que se está dividiendo en múltiples grupos.
- En la práctica, puede que no exista un codo afilado (codo agudo) y, como método heurístico, ese "codo" no siempre puede identificarse sin ambigüedades.



Universidad Nacional Autónoma de México
Facultad de Ingeniería

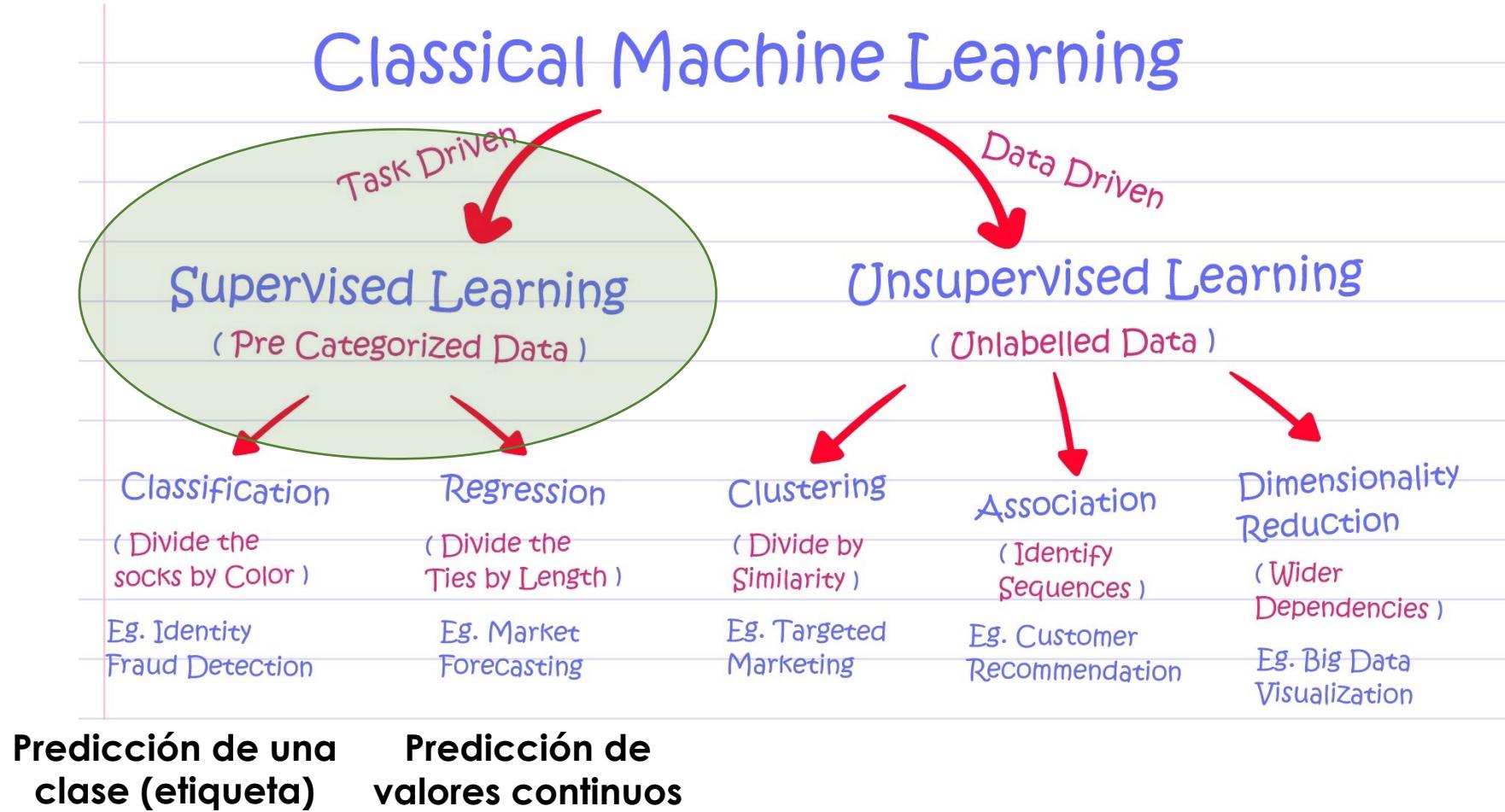
Pronóstico y clasificación Enfoques de aprendizaje supervisado

Guillermo Molero-Castillo
guillermo.molero@ingenieria.unam.edu

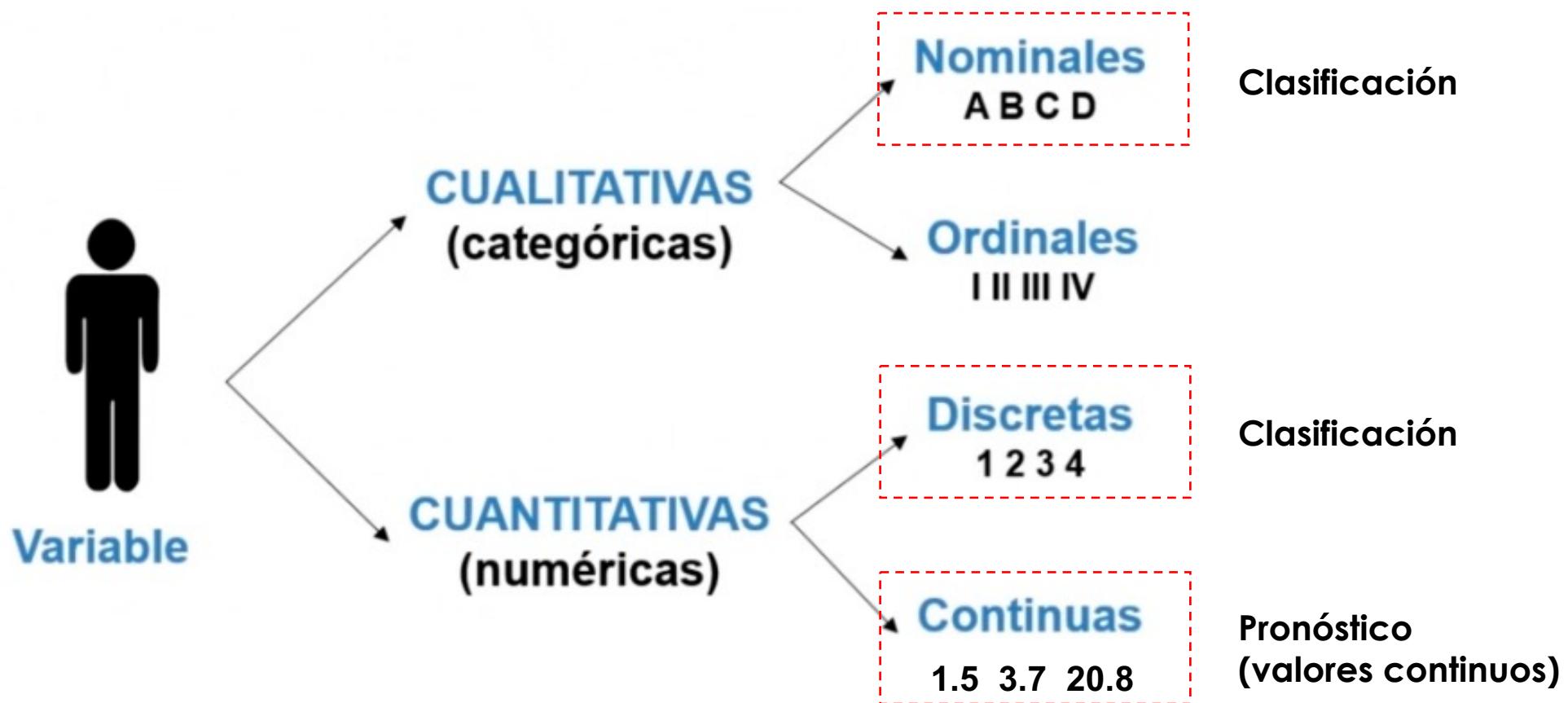
Octubre, 2021

Contexto





Tipos de variables



Aprender

Adquirir conocimientos (comprensión) mediante el análisis o la experiencia.

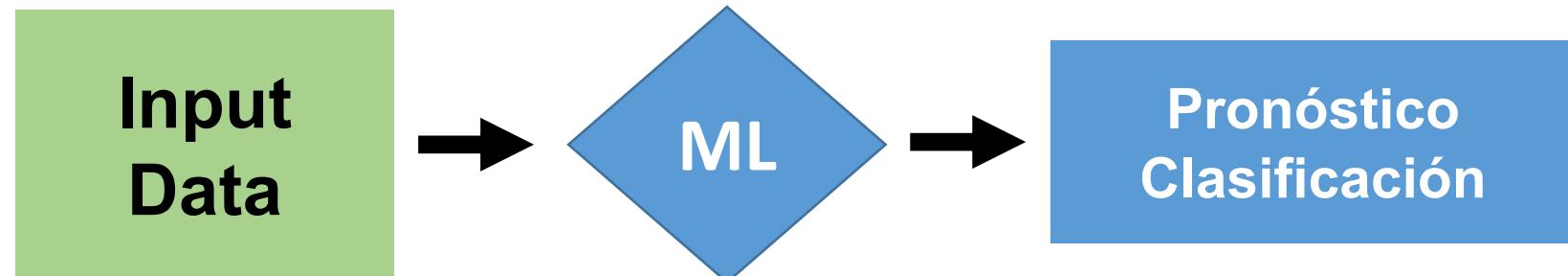
Aprendizaje automático

- Es aprender automáticamente a partir de los datos, a través de un proceso de inferencia, modelo o aprendizaje de ejemplos (reconocimiento de objetos o personas).
- Es ideal para áreas con gran cantidad de datos en ausencia de una teoría general.
- La extracción automática de información es útil mediante la construcción de buenos modelos inferenciales.

Contexto

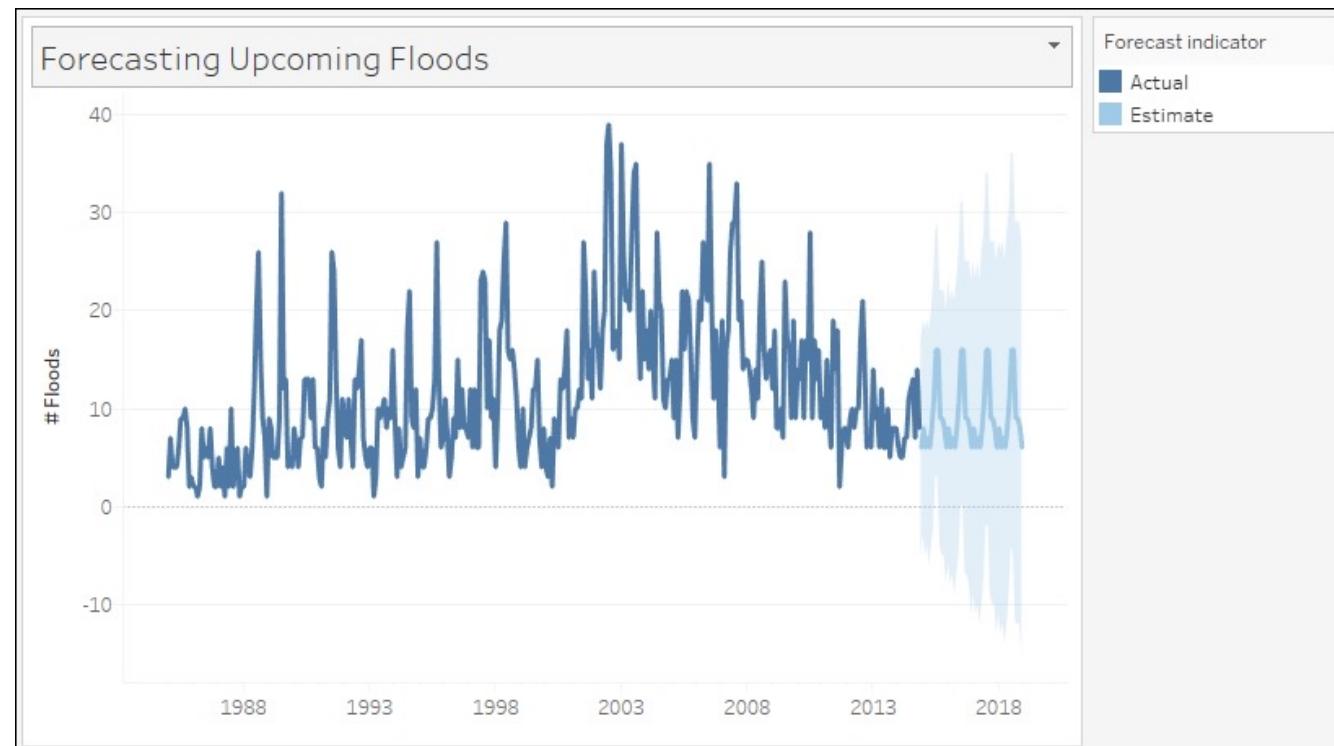
En la actualidad, los modelos de pronóstico se han vuelto predominantes. En diferentes disciplinas se trata de aplicar los algoritmos de aprendizaje automático:

- Los economistas para predecir los precios del mercado, obtener ganancias.
- Los médicos para diagnóstico, por ejemplo, clasificar si un tumor es maligno o benigno.
- Los meteorólogos para predecir el clima.
- En recursos humanos para verificar si el solicitante cumple con los criterios mínimos para el trabajo.
- Entre otros.



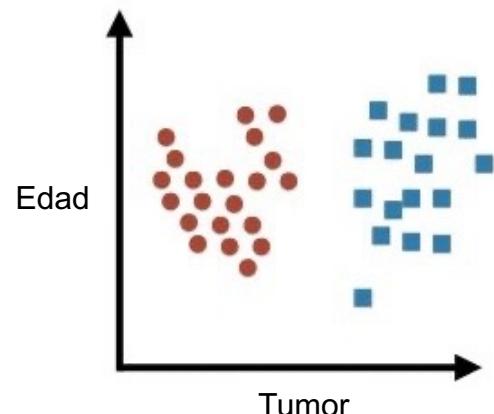
Pronóstico

- Modela funciones de valor continuo, es decir, predice valores desconocidos o faltantes.

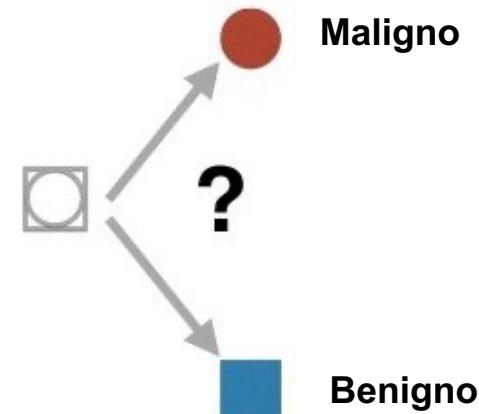


Clasificación

- Predice etiquetas de una o más clases de tipo discretas (0, 1, 2) o nominales (A, B, C; o positivo, negativo; y otros).
- Para esta clasificación se construye un modelo a través de un conjunto de entrenamiento (*training*).
- Se evalúa el modelo con un conjunto de prueba, que es independiente del entrenamiento. De lo contrario, se produce un sobre-ajuste (ajuste excesivo).

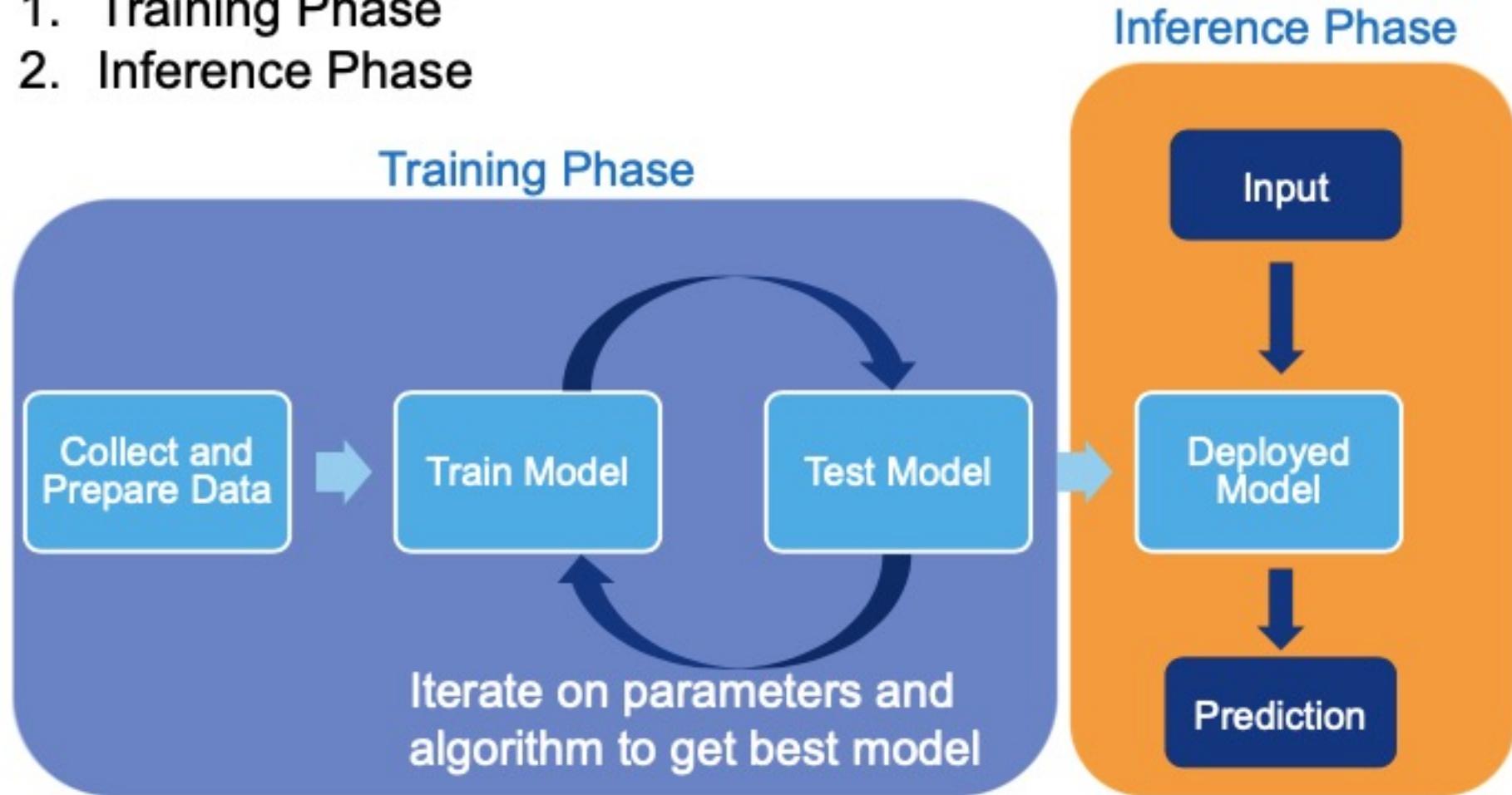


1) Aprender de los
datos de entrenamiento



2) Mapear nuevos
datos (nunca vistos)

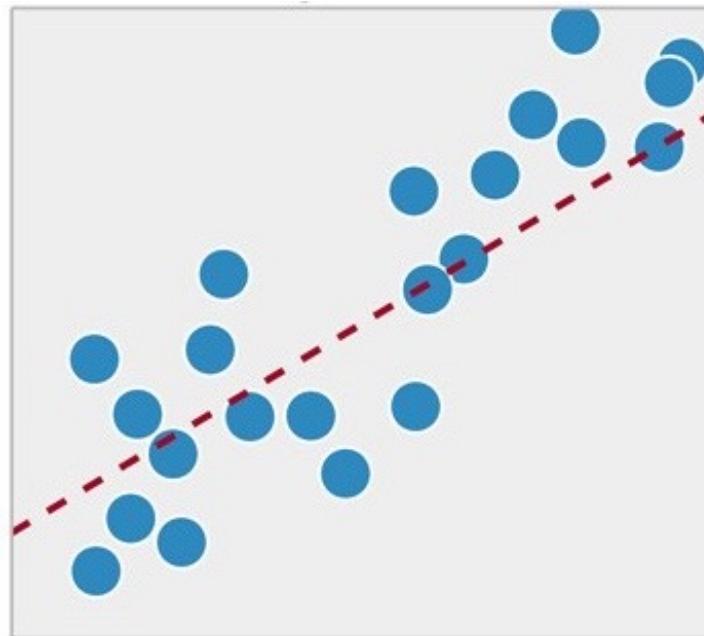
1. Training Phase
2. Inference Phase



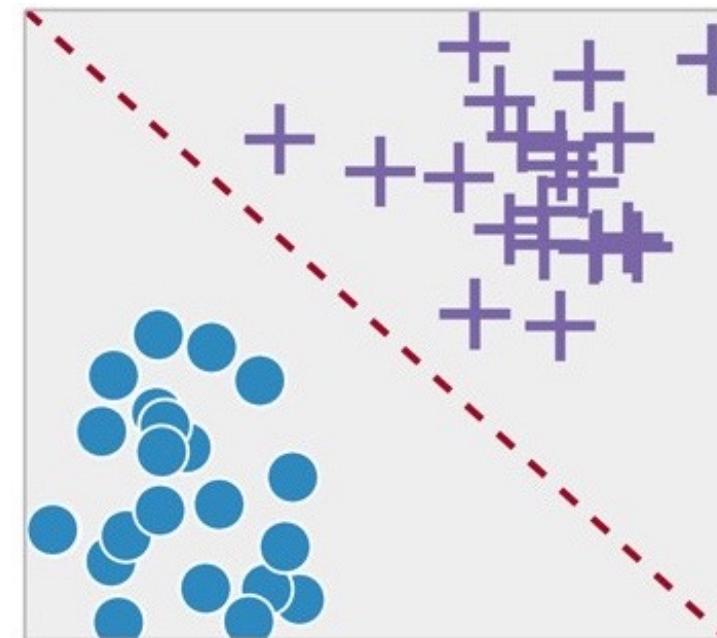
Contexto

En ambos casos, **pronóstico y clasificación**, si la precisión es aceptable, se utiliza el modelo para pronosticar o clasificar nuevos datos, cuyos valores o etiquetas no se conocen.

Pronóstico



Clasificación



Algoritmos

La predicción es importante, ayuda a automatizar actividades. Sin embargo, solo dice lo que sucederá, pero no lo que se debería hacer.

- Linear regression / Logistic regression
- Nearest Neighbor (kNN)
- Decision trees
- Support vector machines
- Artificial Neural Networks
- Bayesian methods
- ...

Los diferentes algoritmos tienen diferentes fortalezas y debilidades.
Se debe seleccionar el enfoque de predicción que sea adecuado para el problema.

Pronóstico Regresión Lineal

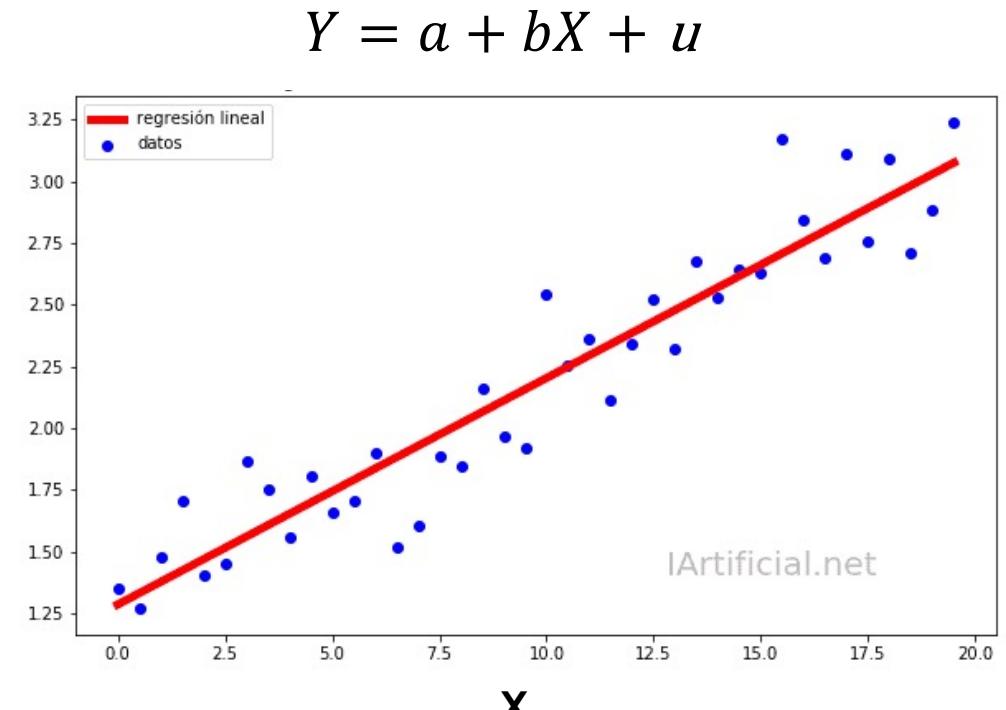
Regresión lineal

La regresión lineal es una forma básica de aprendizaje supervisado, cuyo objetivo es calcular una ecuación que minimiza la distancia entre la línea ajustada (recta) y todos los puntos de datos.

El propósito es proporcionar una base para desarrollar y aprender otros algoritmos de ML.

F1	F2	F3	F4	Class (Y)
1	0	1	1	0.5
0	1	1	0	2
1	0	1	0	0
0	1	0	1	1.2

Es la variable dependiente (output)



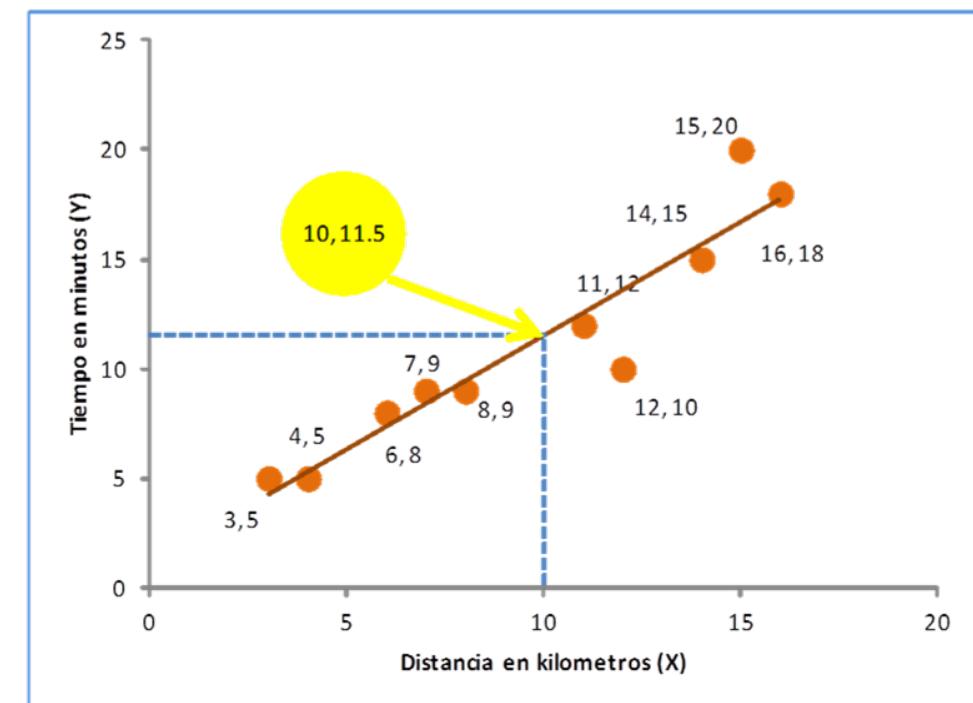
Es la variable independiente (input)

Regresión lineal

La regresión según el número de variables son:

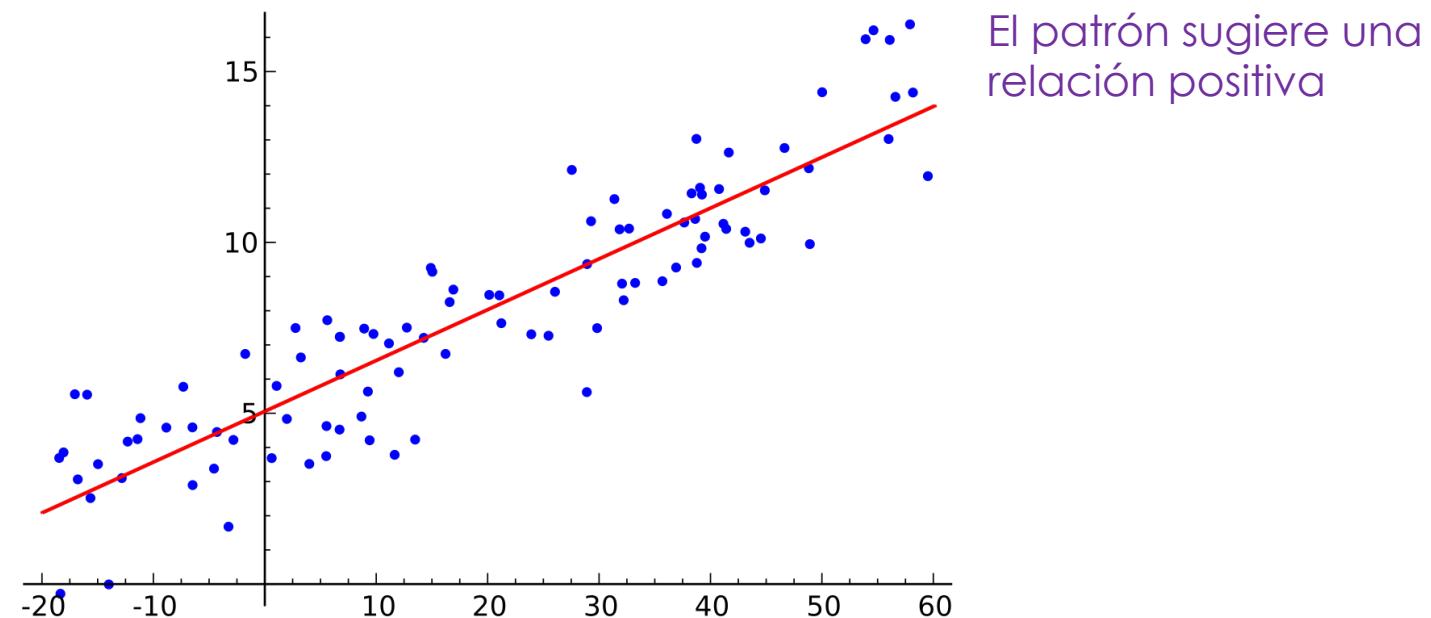
- Si se tiene dos variables se trata de un problema de **regresión lineal simple**.
- Si se tiene más de dos variables se trata de un problema de **regresión lineal múltiple**.

Se utiliza para pronosticar una variable dependiente (clase): **Y**, en función de una o más variables independientes: **X₁, X₂, X₃, ..., X_n**



Regresión lineal

- Un diagrama de dispersión ofrece una aproximación sobre el **tipo de relación** entre las variables.
- Con base en los datos, se traza una recta (línea) que modele mejor los puntos.
- A esto se conoce como la recta de mejor ajuste.



Regresión lineal simple

Regresión lineal simple

En una regresión lineal simple se evalúa una sola variable independiente (X), cuya ecuación lineal es:

$$Y = a + bX + u$$

Valores observados

1. Dada una entrada **X**, se calcula una salida **Y**:
2. Para esto se estiman los parámetros **a** y **b** (conocidos como coeficientes)

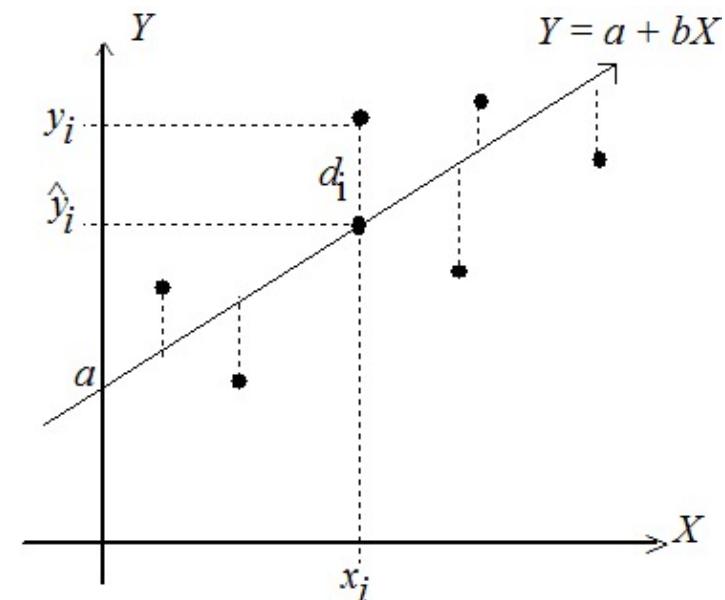
donde:

- **a** es el intercepto (corta el eje Y).

$$a = \bar{y} - b\bar{x}$$

- **b** es la pendiente de la recta.

$$b = \frac{S_{xy}}{S_x^2}$$



Regresión lineal múltiple

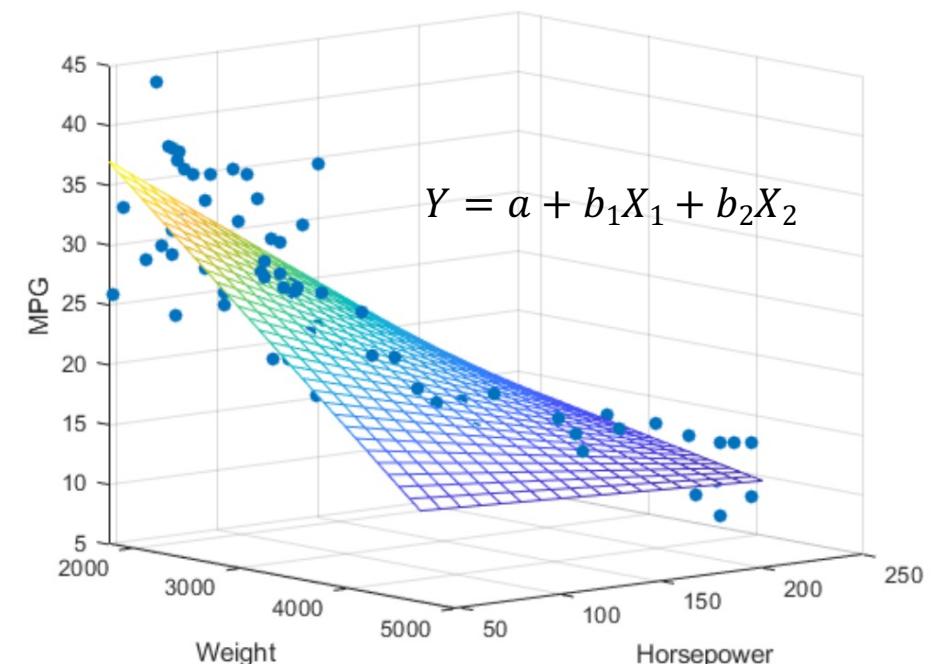
Regresión lineal múltiple

En una **regresión lineal múltiple** se evalúa dos o más variables independientes (X_1, X_2, \dots, X_n).

1. Se ajusta una regresión lineal: $Y = a + b_1X_1 + b_2X_2 \dots + b_nX_n + u$

Donde:

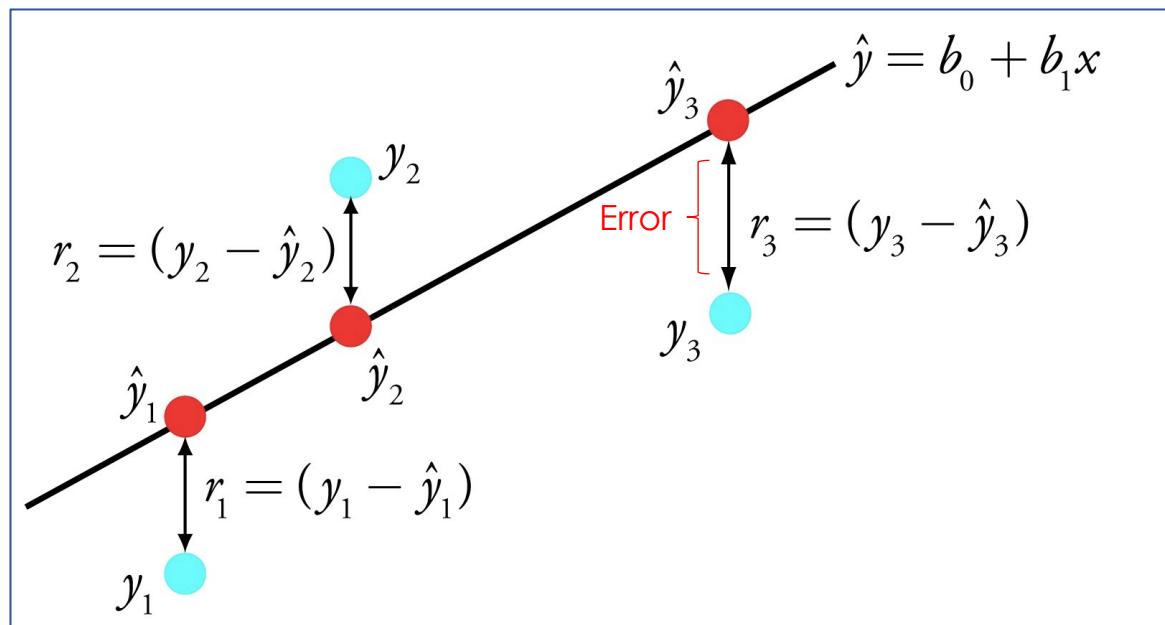
- a es el intercepto (corta el eje Y).
- b_1, b_2, \dots, b_n son valores de la pendiente.



Residuo

Residuo

- Al ajustar los datos en el hiperplano (recta) que "pasa por" los puntos, puede existir una diferencia entre el punto pronosticado y la observación real.
- A esto se conoce como **residuo**.



Residuo

La diferencia entre $Y - \hat{Y}$ es el error estimado.

- Y es el valor real.
- \hat{Y} es el valor pronosticado: $\hat{Y}_i = a + bX_i$

$$\text{Suma del error cuadrático} = SSE = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

A partir de **SSE** se calcula el **residuo (SEE)**, que mide la dispersión de los valores observados alrededor de la línea de regresión:

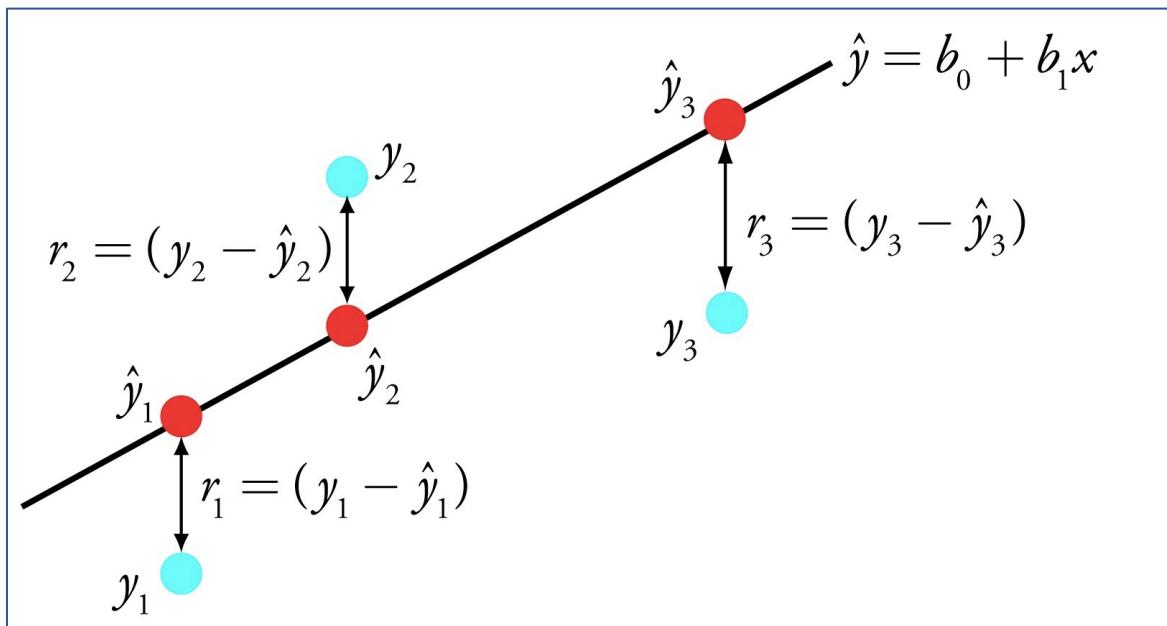
$$\text{Residuo} = \text{Error residual} = SEE = \sqrt{\frac{SSE}{n-2}} = \sqrt{\frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n-2}}$$

$$Y = a + b_i X_i + u$$

Error cuadrático medio

Error cuadrático medio

- El error cuadrático medio (MSE, por sus siglas en inglés) es el criterio de evaluación más usado para problemas de regresión (pronóstico).
- Se usa para estimar la diferencia (error) entre el valor real (azul) y el valor estimado (rojo).
- Se calcula el error al cuadrado, en lugar del error simple, para que el valor sea positivo.



$$\text{Error cuadrático} = (\text{real} - \text{estimado})^2$$

$$MSE = \frac{1}{M} \sum_{i=1}^M (\text{real}_i - \text{estimado}_i)^2$$

Es el total de puntos

- n (número de objetos)
- $n - 1$

Raíz del error cuadrático medio

- Una variante del **MSE** es la raíz cuadrada del error cuadrático medio (**RMSE**).
- En varios algoritmos supervisados se necesita calcular el error en cada iteración, para aprender de sus errores.
- Normalmente, se usa el MSE durante el proceso de aprendizaje y su raíz cuadrada al final, para dar una estimación en términos de la calidad de la predicción.

$$RMSE = \sqrt{\frac{1}{M} \sum_{i=1}^M (real_i - estimado_i)^2}$$

Bondad de ajuste

Bondad de ajuste

- La bondad de ajuste, conocido también como coeficiente de determinación (R^2 o Score), se utiliza para medir la precisión del modelo de regresión.
- Indica qué tan cerca están los datos de la línea de regresión ajustada.
- Representa un valor entre 0 y 1 (porcentaje).

$$\bar{R}^2 = 1 - \frac{n - 1}{n - k - 1} [1 - R^2]$$

Número de
elementos

Número de variables
independientes

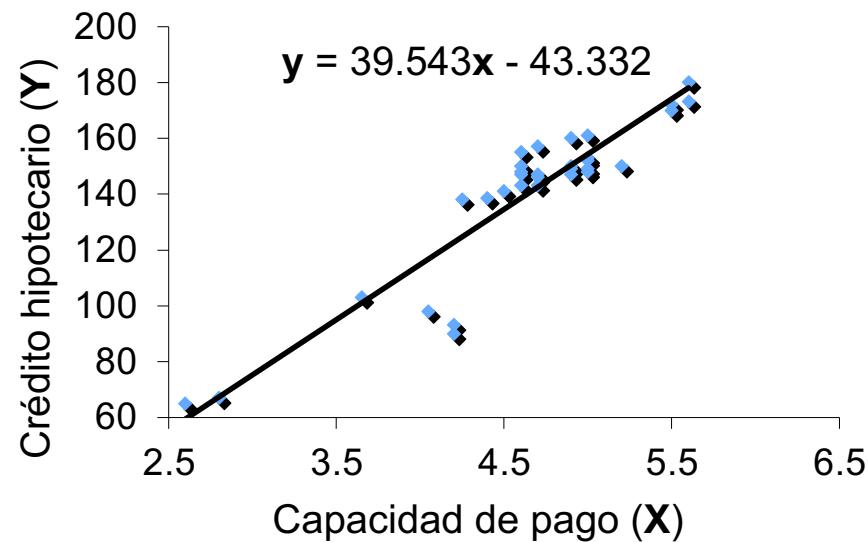
Coeficiente de correlación
(valor real y valores pronosticados)

$$\text{corr}(Y, \hat{Y})$$

Bondad de ajuste

Por ejemplo, Si \bar{R}^2 es 0.87, indica:

- Si se conoce la **capacidad de pago** (ingresos), entonces se puede lograr, con un 87% de efectividad, el pronóstico de algún **crédito hipotecario**.



- 0 cuando las variables son independientes.
- 1 cuando las variables tienen una relación perfecta.

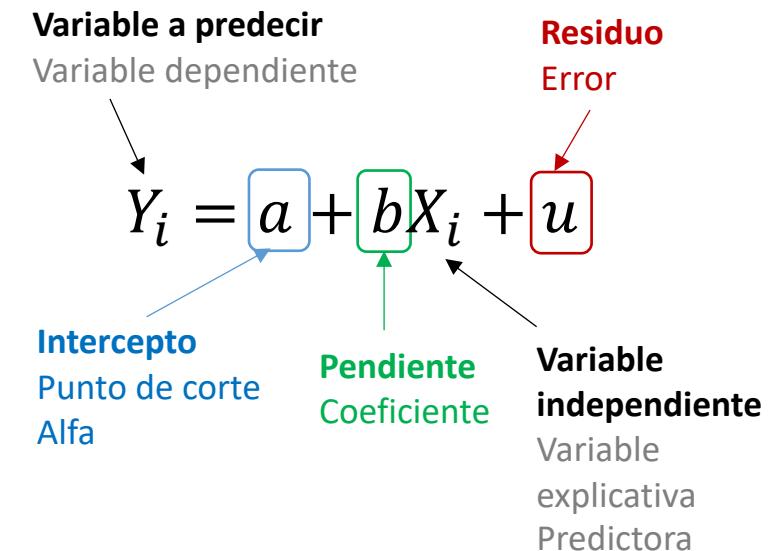
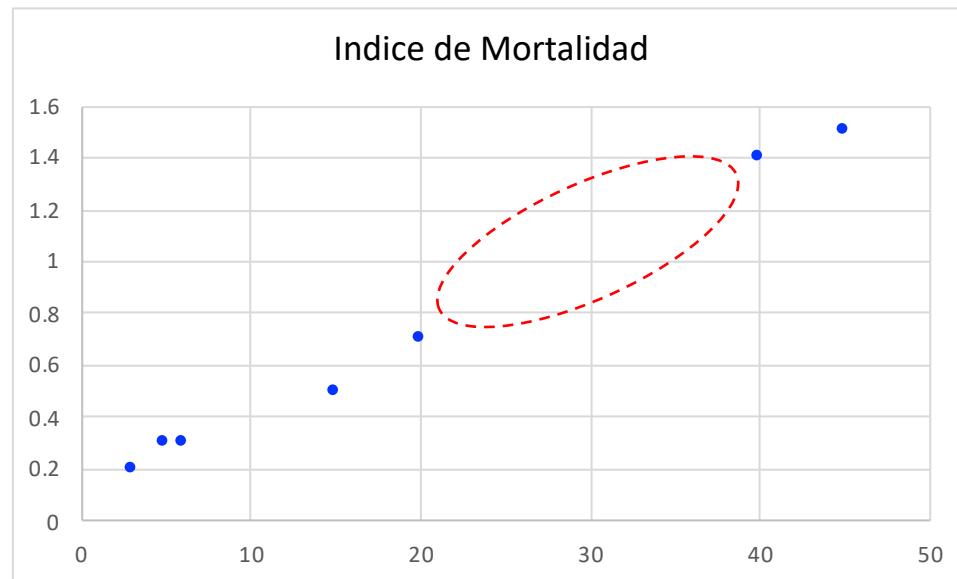
Ejemplo 1

Ejemplo

Sean dos variables: Consumo diario de cigarrillos (X) e índice de mortalidad (Y)

Nro	Cigarrillos	Mortalidad
1	3	0.2
2	5	0.3
3	6	0.3
4	15	0.5
5	20	0.7
6	40	1.4
7	45	1.5
8		
9		
10		
11		
12		

19.14 **0.70**



$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n} \quad \bar{Y} = \frac{\sum_{i=1}^n Y_i}{n}$$

Ejemplo

Solución:

Nro	Cigarrillos	Mortalidad
1	3	0.2
2	5	0.3
3	6	0.3
4	15	0.5
5	20	0.7
6	40	1.4
7	45	1.5
8		
9		
10		
11		
12		
	19.14	0.70

1

2

3

	S_x	S_y	S_{xy}
	260.59	0.25	8.07
	200.02	0.16	5.66
	172.73	0.16	5.26
	17.16	0.04	0.83
	0.73	0.00	0.00
	435.02	0.49	14.60
	668.59	0.64	20.69
	15.83	0.5	7.87

$$S_x = \sqrt{\sum_{i=1}^n \frac{(X_i - \bar{X})^2}{n}}$$

$$S_y = \sqrt{\sum_{i=1}^n \frac{(Y_i - \bar{Y})^2}{n}}$$

$$S_{xy} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n}$$

Ejemplo

Solución:

Nro	Cigarrillos	Mortalidad
1	3	0.2
2	5	0.3
3	6	0.3
4	15	0.5
5	20	0.7
6	40	1.4
7	45	1.5
8		
9		
10		
11		
12		
		19.14
		0.70

1

2

3

	S_x	S_y	S_{xy}
	260.59	0.25	8.07
	200.02	0.16	5.66
	172.73	0.16	5.26
	17.16	0.04	0.83
	0.73	0.00	0.00
	435.02	0.49	14.60
	668.59	0.64	20.69
		15.83	0.5
			7.87

4

Pendiente (b)

$$b = \frac{S_{xy}}{S_x^2}$$

$$b = \frac{7.87}{(15.83)^2} = \frac{7.87}{250.58}$$

$$\mathbf{b = 0.031}$$

Ejemplo

Solución:

Nro	Cigarrillos	Mortalidad
1	3	0.2
2	5	0.3
3	6	0.3
4	15	0.5
5	20	0.7
6	40	1.4
7	45	1.5
8		
9		
10		
11		
12		
		19.14
		0.70

1

2

3

	S_x	S_y	S_{xy}
	260.59	0.25	8.07
	200.02	0.16	5.66
	172.73	0.16	5.26
	17.16	0.04	0.83
	0.73	0.00	0.00
	435.02	0.49	14.60
	668.59	0.64	20.69
		15.83	0.5
			7.87

4

Pendiente (b)

$$b = 0.031$$

5

Intercepto (a)

$$Y = a + bX$$

$$a = \bar{y} - b\bar{x}$$

$$a = 0.7 - (0.031 * 19.14)$$

$$a = 0.7 - 0.6$$

$$a = 0.1$$

Ejemplo

Solución:

Nro	Cigarrillos	Mortalidad	Sx	Sy	Sxy	Ŷ (Pronóstico)
1	3	0.2	260.59	0.25	8.07	0.19
2	5	0.3	200.02	0.16	5.66	0.26
3	6	0.3	172.73	0.16	5.26	0.29
4	15	0.5	17.16	0.04	0.83	0.57
5	20	0.7	0.73	0.00	0.00	0.73
6	40	1.4	435.02	0.49	14.60	1.35
7	45	1.5	668.59	0.64	20.69	1.51
8						
9						
10						
11						
12						
	19.14	0.70	15.83	0.5	7.87	

6

Pronóstico: \hat{Y}

$$Y_i = a + bX_i$$

$$\hat{Y}_1 = 0.1 + 0.031(3)$$
$$\hat{Y}_1 = 0.19$$

$$\hat{Y}_2 = 0.1 + 0.031(5)$$
$$\hat{Y}_2 = 0.26$$

$$\hat{Y}_3 = 0.1 + 0.031(6)$$
$$\hat{Y}_3 = 0.29$$

...

Ejemplo

Solución:

Nro	Cigarrillos	Mortalidad	Sx	Sy	Sxy	\hat{Y} (Pronóstico)	$(Y - \hat{Y})^2$
1	3	0.2	260.59	0.25	8.07	0.19	0.000
2	5	0.3	200.02	0.16	5.66	0.26	0.002
3	6	0.3	172.73	0.16	5.26	0.29	0.000
4	15	0.5	17.16	0.04	0.83	0.57	0.005
5	20	0.7	0.73	0.00	0.00	0.73	0.001
6	40	1.4	435.02	0.49	14.60	1.35	0.002
7	45	1.5	668.59	0.64	20.69	1.51	0.000
8							
9							
10							
11							
12							
	19.14	0.70		15.83	0.5	7.87	0.01

7 Error cuadrático

$$SSE = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

$$SSE_1 = (0.2 - 0.19)^2 \\ SSE_1 = 0.0001$$

$$SSE_2 = (0.3 - 0.26)^2 \\ SSE_2 = 0.002$$

$$SSE_3 = (0.3 - 0.29)^2 \\ SSE_3 = 0.0001$$

...

Ejemplo

Solución:

Nro	Cigarrillos	Mortalidad	Sx	Sy	Sxy	\hat{Y} (Pronóstico)	$(Y - \hat{Y})^2$
1	3	0.2	260.59	0.25	8.07	0.19	0.000
2	5	0.3	200.02	0.16	5.66	0.26	0.002
3	6	0.3	172.73	0.16	5.26	0.29	0.000
4	15	0.5	17.16	0.04	0.83	0.57	0.005
5	20	0.7	0.73	0.00	0.00	0.73	0.001
6	40	1.4	435.02	0.49	14.60	1.35	0.002
7	45	1.5	668.59	0.64	20.69	1.51	0.000
8							
9							
10							
11							
12							
	19.14	0.70		15.83	0.5	7.87	0.01

8 Residuo

$$SEE = \sqrt{\frac{SSE}{n - 2}}$$

$$SEE = \sqrt{\frac{0.01}{7 - 2}}$$

$$SEE = 0.04$$

Ejemplo

9

MSE y RMSE

Nro	Cigarrillos	Mortalidad	Sx	Sy	Sxy	\hat{Y} (Pronóstico)	$(Y - \hat{Y})^2$
1	3	0.2	260.59	0.25	8.07	0.19	0.000
2	5	0.3	200.02	0.16	5.66	0.26	0.002
3	6	0.3	172.73	0.16	5.26	0.29	0.000
4	15	0.5	17.16	0.04	0.83	0.57	0.005
5	20	0.7	0.73	0.00	0.00	0.73	0.001
6	40	1.4	435.02	0.49	14.60	1.35	0.002
7	45	1.5	668.59	0.64	20.69	1.51	0.000
8							
9							
10							
11							
12							
	19.14	0.70	15.83	0.5	7.87	0.01	

$$MSE = \frac{1}{M} \sum_{i=1}^M (real_i - estimado_i)^2$$

$$MSE = \frac{1}{7} (0.01)$$

$$MSE = 0.0014$$

$$RMSE = \sqrt{0.0014} = 0.0377$$

Los pronósticos del modelo final se alejan en promedio 0.037 unidades del valor real.

Ejemplo

10

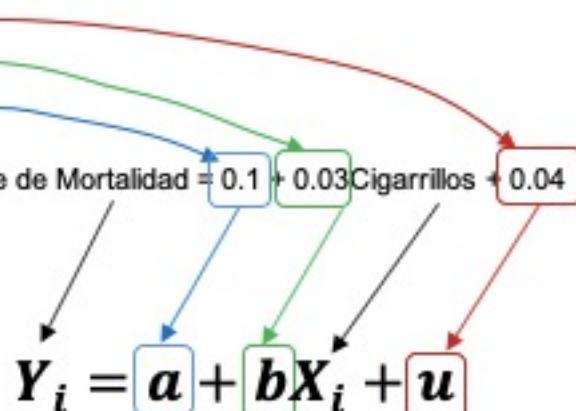
Bondad de ajuste

Residuo	0.04
Pendiente (b)	0.03
Intercepo (a)	0.10
<i>R</i> (coef. Correl)	0.997
<i>R</i> cuadrado	0.99
Total Observaciones	7
<i>R</i> 2 Score	0.99
$\bar{R}^2 = 1 - \frac{n-1}{n-k-1} [1 - R^2]$	
<i>Número de variables independientes</i>	

$$\bar{R}^2 = 1 - \frac{7-1}{7-1-1} [1 - (0.997)^2]$$

$$\bar{R}^2 = 1 - 0.012$$

$$\bar{R}^2 = 0.99$$



Mortalidad	\hat{Y} (Pronóstico)
0.2	0.19
0.3	0.26
0.3	0.29
0.5	0.57
0.7	0.73
1.4	1.35
1.5	1.51

$$\text{corr}(Y, \hat{Y}) = r = \frac{S_{xy}}{S_x S_y} = 0.997$$

Ejemplo

Función de estimación: $Y_i = a + bX_i + u$

$$Y_i = 0.1 + 0.031X_i + 0.04$$

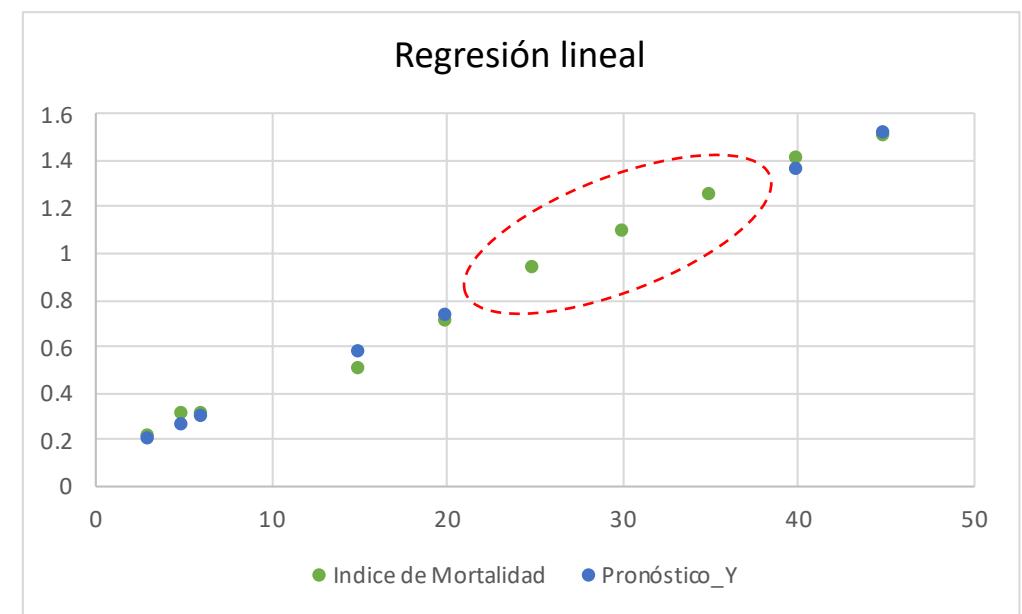
Nuevos casos: \hat{Y}

Nro	Cigarrillos	Mortalidad
1	3	0.2
2	5	0.3
3	6	0.3
4	15	0.5
5	20	0.7
6	40	1.4
7	45	1.5
8	25	1.09
9	30	1.24
10	35	1.71
11		
12		
	19.14	0.70

$$\hat{Y}_8 = 0.1 + 0.031(25) + 0.04 = 0.93$$

$$\hat{Y}_9 = 0.1 + 0.031(30) + 0.04 = 1.09$$

$$\hat{Y}_{10} = 0.1 + 0.031(35) + 0.04 = 1.24$$



Si \bar{R}^2 es 0.99, indica que a mayor **consumo diario de cigarrillos**, hay mayor índice de mortalidad. El pronóstico se logra con un 99% de efectividad (grado de intensidad).

Ejemplo 2

Ejemplo

Sean dos variables: Meses del año y Cantidad (coches vendidos)

Nro	Mes(X)	Cantidad(Y)
1	1	132
2	2	170
3	3	230
4	4	205
5	5	260
6	6	302
7	7	330
8	8	360
9	9	480
10	10	440
11	11	485
12	12	540
13		
14		
15		

6.50 **327.83**

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$$

$$\bar{Y} = \frac{\sum_{i=1}^n Y_i}{n}$$

Variable a predecir
Variable dependiente

Intercepto
Punto de corte
Alfa

$$Y_i = a + bX_i + u$$

Pendiente
Coeficiente

Variable independiente
Variable explicativa
Predictora

Residuo
Error

Ejemplo

Solución:

Nro	Mes(X)	Cantidad(Y)
1	1	132
2	2	170
3	3	230
4	4	205
5	5	260
6	6	302
7	7	330
8	8	360
9	9	480
10	10	440
11	11	485
12	12	540
13		
14		
15		

6.50

327.83

1	2	3
Sx	Sy	Sxy
30.25	38350.69	1077.08
20.25	24911.36	710.25
12.25	9571.36	342.42
6.25	15088.03	307.08
2.25	4601.36	101.75
0.25	667.36	12.92
0.25	4.69	1.08
2.25	1034.69	48.25
6.25	23154.69	380.42
12.25	12581.36	392.58
20.25	24701.36	707.25
30.25	45014.69	1166.92

3.45

129.00

437.33

$$S_x = \sqrt{\sum_{i=1}^n \frac{(X_i - \bar{X})^2}{n}}$$

$$S_y = \sqrt{\sum_{i=1}^n \frac{(Y_i - \bar{Y})^2}{n}}$$

$$S_{xy} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n}$$

Ejemplo

Solución:

Nro	Mes(X)	Cantidad(Y)
1	1	132
2	2	170
3	3	230
4	4	205
5	5	260
6	6	302
7	7	330
8	8	360
9	9	480
10	10	440
11	11	485
12	12	540
13		
14		
15		
	6.50	327.83

1

2

3

4

	S_x	S_y	S_{xy}
	30.25	38350.69	1077.08
	20.25	24911.36	710.25
	12.25	9571.36	342.42
	6.25	15088.03	307.08
	2.25	4601.36	101.75
	0.25	667.36	12.92
	0.25	4.69	1.08
	2.25	1034.69	48.25
	6.25	23154.69	380.42
	12.25	12581.36	392.58
	20.25	24701.36	707.25
	30.25	45014.69	1166.92
	3.45	129.00	437.33

Pendiente (**b**)

$$b = \frac{S_{xy}}{S_x^2}$$

$$b = \frac{437.33}{(3.45)^2} = \frac{437.33}{11.9}$$

$$\color{blue}{b = 36.7}$$

Ejemplo

Solución:

Nro	Mes(X)	Cantidad(Y)
1	1	132
2	2	170
3	3	230
4	4	205
5	5	260
6	6	302
7	7	330
8	8	360
9	9	480
10	10	440
11	11	485
12	12	540
13		
14		
15		
	6.50	327.83

1

2

3

4

5

	Sx	Sy	Sxy
	30.25	38350.69	1077.08
	20.25	24911.36	710.25
	12.25	9571.36	342.42
	6.25	15088.03	307.08
	2.25	4601.36	101.75
	0.25	667.36	12.92
	0.25	4.69	1.08
	2.25	1034.69	48.25
	6.25	23154.69	380.42
	12.25	12581.36	392.58
	20.25	24701.36	707.25
	30.25	45014.69	1166.92
	3.45	129.00	437.33

Pendiente (b)

$$b = 36.7$$

Intercepto (a)

$$Y = a + bX$$

$$a = \bar{y} - b\bar{x}$$

$$a = 327.83 - (36.7 * 6.5)$$

$$a = 327.83 - 238.55$$

$$a = 89.29$$

Ejemplo

Solución:

Nro	Mes_X	Cantidad_Y	Sx	Sy	Sxy	Ŷ (Pronóstico)
1	1	132	30.25	38350.69	1077.08	125.99
2	2	170	20.25	24911.36	710.25	162.69
3	3	230	12.25	9571.36	342.42	199.39
4	4	205	6.25	15088.03	307.08	236.09
5	5	260	2.25	4601.36	101.75	272.78
6	6	302	0.25	667.36	12.92	309.48
7	7	330	0.25	4.69	1.08	346.18
8	8	360	2.25	1034.69	48.25	382.88
9	9	480	6.25	23154.69	380.42	419.58
10	10	440	12.25	12581.36	392.58	456.28
11	11	485	20.25	24701.36	707.25	492.98
12	12	540	30.25	45014.69	1166.92	529.68
13						
14						
15						
	6.50	327.83	3.45	129.00	437.33	

6

Pronóstico: \hat{Y}

$$Y_i = a + bX_i$$

$$\hat{Y}_1 = 89.29 + 36.69(1) \\ \hat{Y}_1 = 125.99$$

$$\hat{Y}_2 = 89.29 + 36.69(2) \\ \hat{Y}_2 = 162.69$$

$$\hat{Y}_3 = 89.29 + 36.69(3) \\ \hat{Y}_3 = 199.39$$

...

Ejemplo

Solución:

Nro	Mes_X	Cantidad_Y	Sx	Sy	Sxy	\hat{Y} (Pronóstico)	$(Y - \hat{Y})^2$
1	1	132	30.25	38350.69	1077.08	125.99	36.15
2	2	170	20.25	24911.36	710.25	162.69	53.49
3	3	230	12.25	9571.36	342.42	199.39	937.23
4	4	205	6.25	15088.03	307.08	236.09	966.28
5	5	260	2.25	4601.36	101.75	272.78	163.44
6	6	302	0.25	667.36	12.92	309.48	56.01
7	7	330	0.25	4.69	1.08	346.18	261.89
8	8	360	2.25	1034.69	48.25	382.88	523.60
9	9	480	6.25	23154.69	380.42	419.58	3650.38
10	10	440	12.25	12581.36	392.58	456.28	265.07
11	11	485	20.25	24701.36	707.25	492.98	63.68
12	12	540	30.25	45014.69	1166.92	529.68	106.51
13							
14							
15							
	6.50	327.83	3.45	129.00	437.33		7083.74

7 Error cuadrático

$$SSE = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

$$SSE_1 = (132 - 125.99)^2 \\ SSE_1 = 36.15$$

$$SSE_2 = (170 - 162.69)^2 \\ SSE_2 = 53.49$$

$$SSE_3 = (230 - 199.39)^2 \\ SSE_3 = 937.23$$

...

Ejemplo

Solución:

Nro	Mes_X	Cantidad_Y	Sx	Sy	Sxy	Ŷ (Pronóstico)	(Y-Ŷ) ²
1	1	132	30.25	38350.69	1077.08	125.99	36.15
2	2	170	20.25	24911.36	710.25	162.69	53.49
3	3	230	12.25	9571.36	342.42	199.39	937.23
4	4	205	6.25	15088.03	307.08	236.09	966.28
5	5	260	2.25	4601.36	101.75	272.78	163.44
6	6	302	0.25	667.36	12.92	309.48	56.01
7	7	330	0.25	4.69	1.08	346.18	261.89
8	8	360	2.25	1034.69	48.25	382.88	523.60
9	9	480	6.25	23154.69	380.42	419.58	3650.38
10	10	440	12.25	12581.36	392.58	456.28	265.07
11	11	485	20.25	24701.36	707.25	492.98	63.68
12	12	540	30.25	45014.69	1166.92	529.68	106.51
13							
14							
15							
	6.50	327.83	3.45	129.00	437.33	7083.74	

8

Error residual

$$SEE = \sqrt{\frac{SSE}{n - 2}}$$

$$SEE = \sqrt{\frac{7083.74}{12 - 2}}$$

$$SEE = 26.62$$

Ejemplo

9

MSE y RMSE

Nro	Mes_X	Cantidad_Y	Sx	Sy	Sxy	Ŷ (Pronóstico)	(Y-Ŷ) ²
1	1	132	30.25	38350.69	1077.08	125.99	36.15
2	2	170	20.25	24911.36	710.25	162.69	53.49
3	3	230	12.25	9571.36	342.42	199.39	937.23
4	4	205	6.25	15088.03	307.08	236.09	966.28
5	5	260	2.25	4601.36	101.75	272.78	163.44
6	6	302	0.25	667.36	12.92	309.48	56.01
7	7	330	0.25	4.69	1.08	346.18	261.89
8	8	360	2.25	1034.69	48.25	382.88	523.60
9	9	480	6.25	23154.69	380.42	419.58	3650.38
10	10	440	12.25	12581.36	392.58	456.28	265.07
11	11	485	20.25	24701.36	707.25	492.98	63.68
12	12	540	30.25	45014.69	1166.92	529.68	106.51
13							
14							
15							
	6.50	327.83	3.45	129.00	437.33	7083.74	

$$MSE = \frac{1}{M} \sum_{i=1}^M (real_i - estimado_i)^2$$

$$MSE = \frac{1}{12} (7083.74)$$

$$MSE = 590.31$$

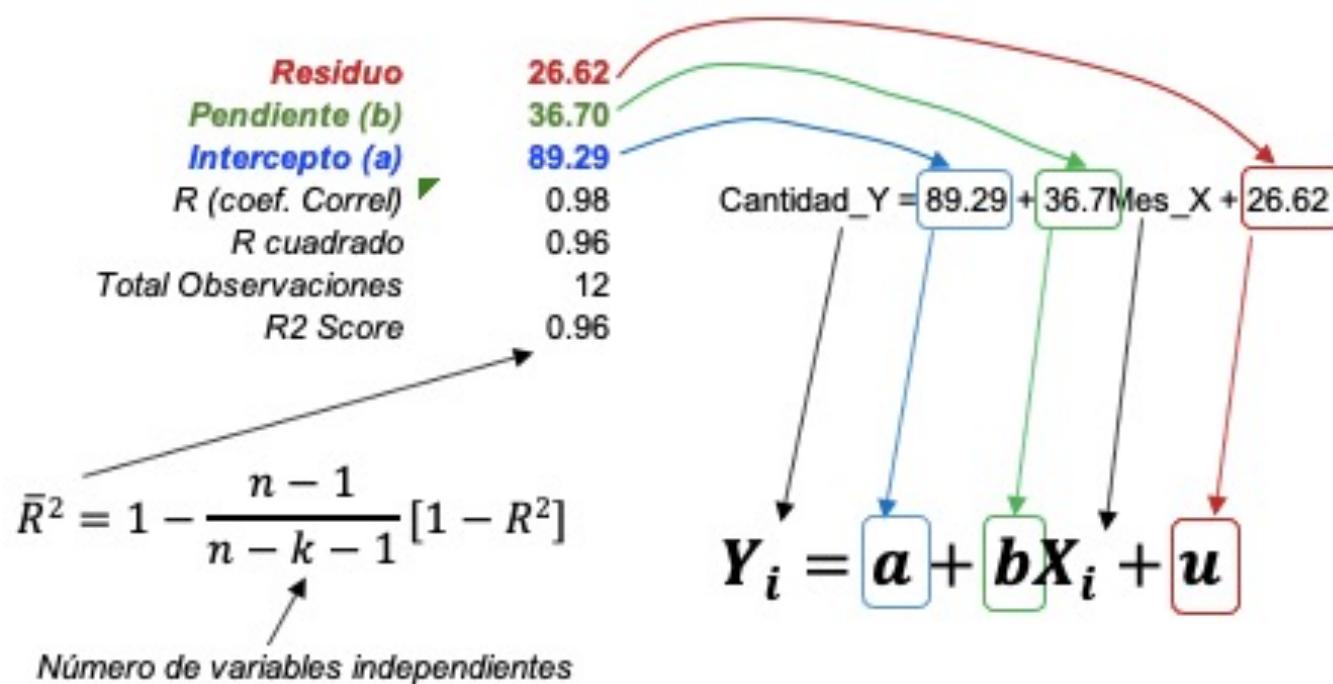
$$RMSE = \sqrt{590.31} = 24.29$$

Los pronósticos del modelo final se alejan en promedio 24.39 unidades del valor real.

Ejemplo

10

Bondad de ajuste



$$\bar{R}^2 = 1 - \frac{n - 1}{n - k - 1} [1 - R^2]$$

$$\bar{R}^2 = 1 - \frac{12 - 1}{12 - 1 - 1} [1 - (0.98)^2]$$

$$\bar{R}^2 = 1 - 0.044$$

$$\bar{R}^2 = 0.96$$

Si \bar{R}^2 es 0.96, indica que el pronóstico de venta (**cantidad**), en un determinado periodo (**mes**), se logrará con un 96% de efectividad (grado de intensidad).

Ejemplo

Función de estimación:

$$Y_i = 89.29 + 36.7X_i + 26.62$$

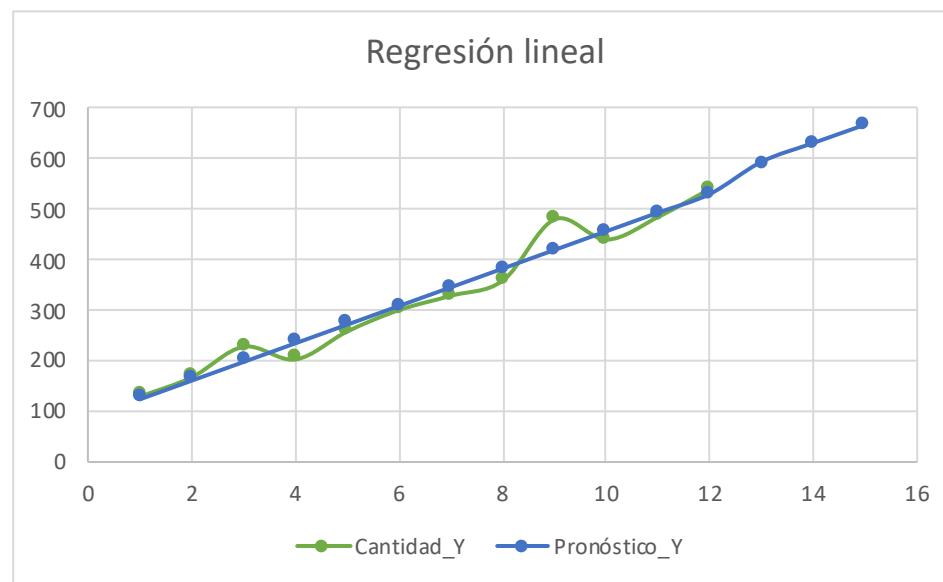
Pronóstico de nuevos casos: \hat{Y}

Nro	Mes_X	Cantidad_Y
1	1	132
2	2	170
3	3	230
4	4	205
5	5	260
6	6	302
7	7	330
8	8	360
9	9	480
10	10	440
11	11	485
12	12	540
13	13	593
14	14	630
15	15	666

$$\hat{Y}_{13} = 89.29 + 36.7(13) + 26.62 = 89.29 + 477.1 + 26.62 = 593.01$$

$$\hat{Y}_{14} = 89.29 + 36.7(14) + 26.62 = 89.29 + 513.8 + 26.62 = 629.71$$

$$\hat{Y}_{15} = 89.29 + 36.7(15) + 26.62 = 89.29 + 550.5 + 26.62 = 666.41$$

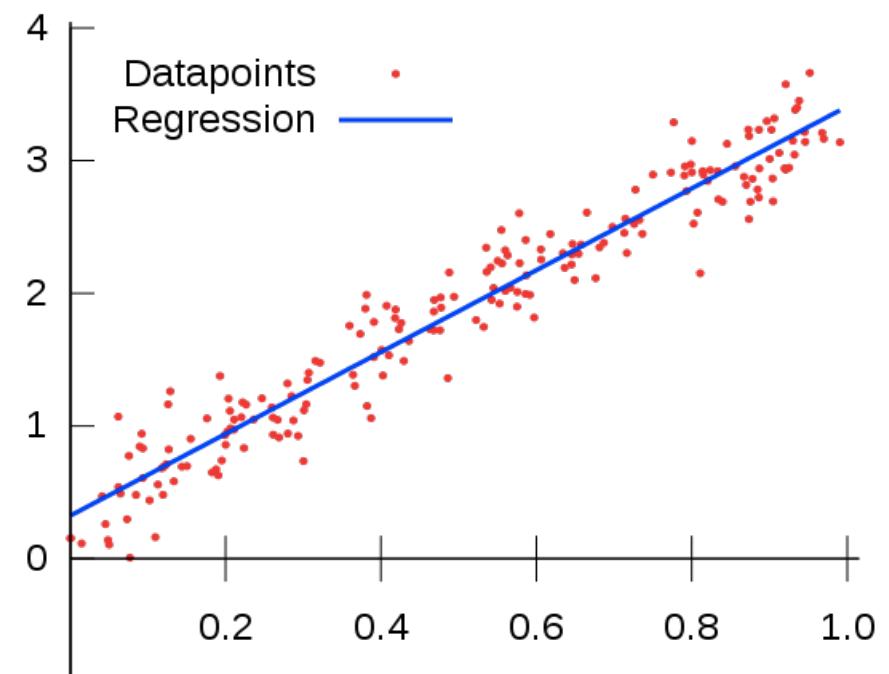


Consideraciones finales

La regresión lineal es un algoritmo útil. Sin embargo, obliga a ajustar los datos en forma de línea, que en ocasiones no encaja.

Cuando el problema incluye una no linealidad, entonces se necesita emplear otros algoritmos.

- Pronóstico de ventas.
- Pronóstico del consumo eléctrico.
- Pronóstico de la demanda de productos y servicios.
- Estimación de precios.
- Pronóstico de la bolsa de valores.
- Entre otros.



Consideraciones finales

Una solución a la no linealidad es usar polinomios, el cual es un problema de regresión.

Entonces, ¿qué tipo de funciones se pueden usar?

Polinomial: $\hat{y} = a + b_1x + b_2x^2 + u$ $\hat{y} = a + b_1x + b_2x^2 + b_3x^3 + u$

Exponencial: $\hat{y} = a + e^{bx}$

Logarítmico: $\hat{y} = a + b \log x$

Sigmoide: $\hat{y} = \frac{1}{1 + e^{-(a+bX)}}$

Gaussiana: $\hat{y} = \frac{(x - \mu_j)}{2\sigma_j^2}$



Universidad Nacional Autónoma de México
Facultad de Ingeniería

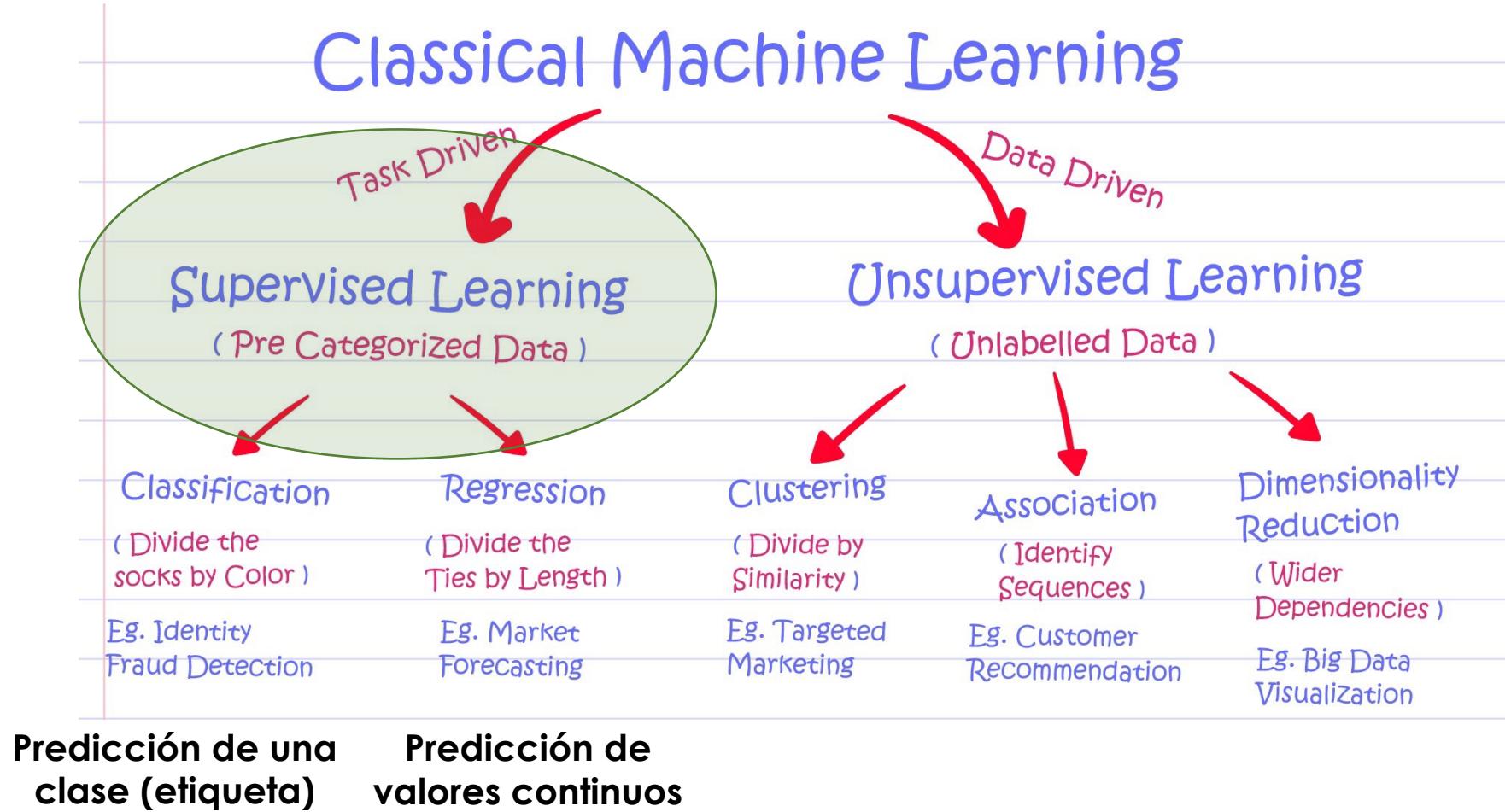
Clasificación Regresión Logística

Guillermo Molero-Castillo
guillermo.molero@ingenieria.unam.edu

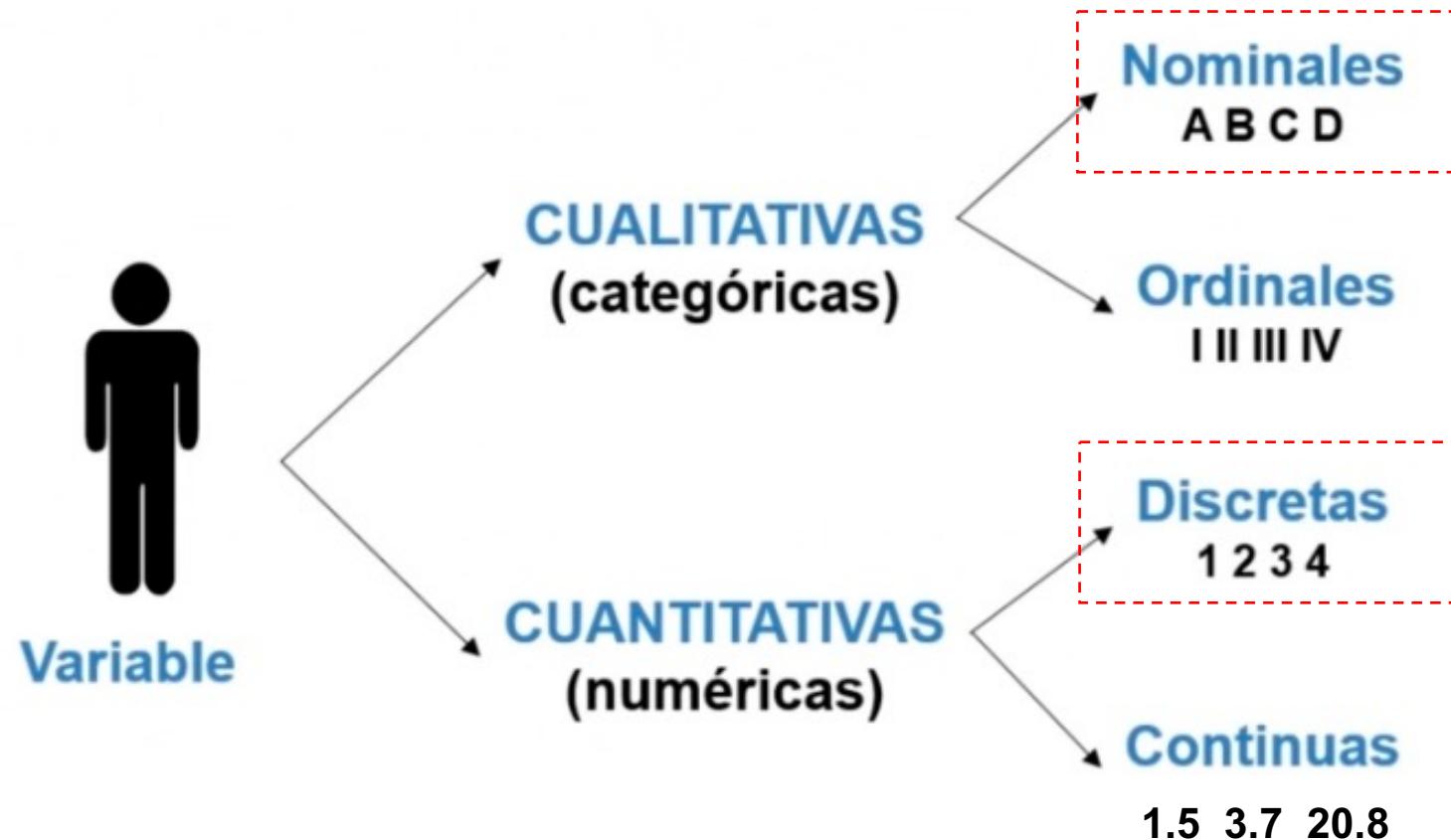
Noviembre, 2021

Contexto



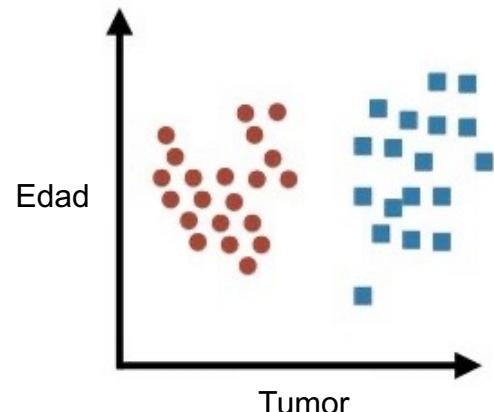


Tipos de variables

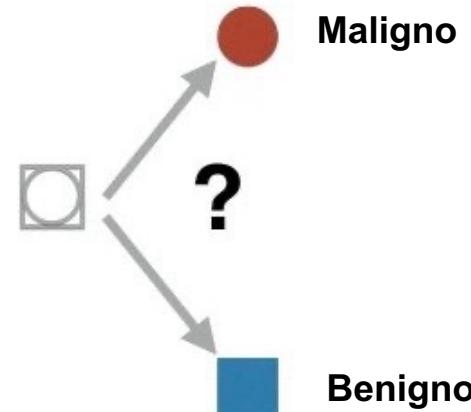


Clasificación

- Predice etiquetas de una o más clases de tipo discretas (0, 1, 2) o nominales (A, B, C; o positivo, negativo; y otros).
- Para esta clasificación se construye un modelo a través de un conjunto de entrenamiento (*training*).
- Se evalúa el modelo con un conjunto de prueba, que es independiente del entrenamiento. De lo contrario, se produce un sobre-ajuste (ajuste excesivo).



1) Aprender de los
datos de entrenamiento



2) Mapear nuevos
datos (nunca vistos)

Contexto

Una solución a la no linealidad es usar otra función.

Polinomial: $\hat{y} = a + b_1x + b_2x^2 + u$

$$\hat{y} = a + b_1x + b_2x^2 + b_3x^3 + u$$

Exponencial: $\hat{y} = a + e^{bx}$

Logarítmico: $\hat{y} = a + b \log x$

Sigmoide:
$$\hat{y} = \frac{1}{1 + e^{-(a+bX)}}$$

Gaussiana:
$$\hat{y} = \frac{(x - \mu_j)}{2\sigma_j^2}$$

Regresión Logística

Regresión Logística

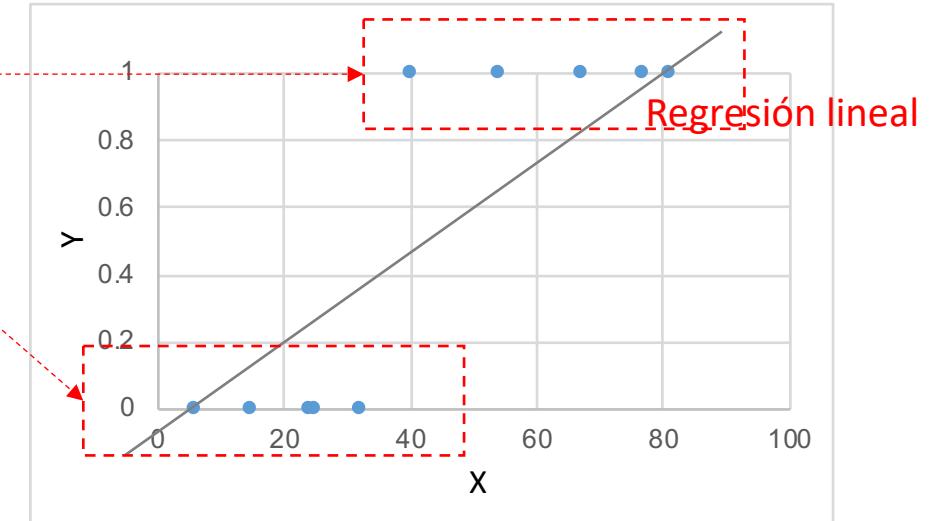
La regresión logística es otro tipo de algoritmo de aprendizaje supervisado cuyo objetivo es predecir valores binarios (0 o 1). Este algoritmo consiste en una transformación a la regresión lineal.

La transformación se debe a que una regresión lineal **no funciona para predecir una variable binaria**.

Nro	X	Y
1	22	0
2	54	1
3	67	1
4	6	0
5	77	1
6	32	0
7	81	1
8	4	0
9	5	0
10	40	1

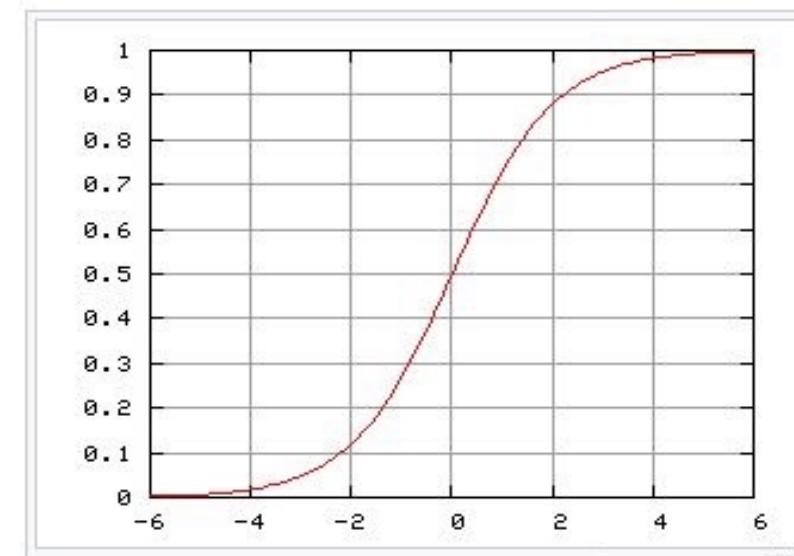
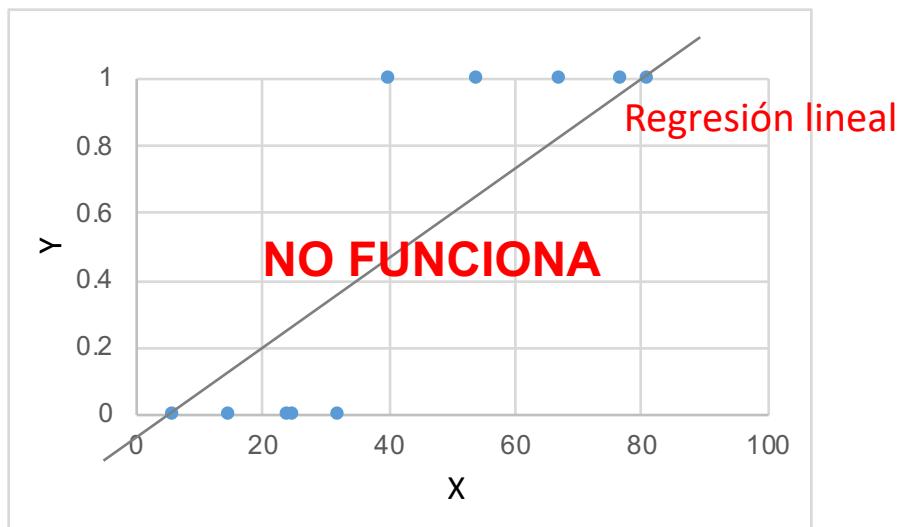
Clasificación: $Y = \{0, 1\}$

$$Y = a + b_i X_i + u$$



Regresión Logística

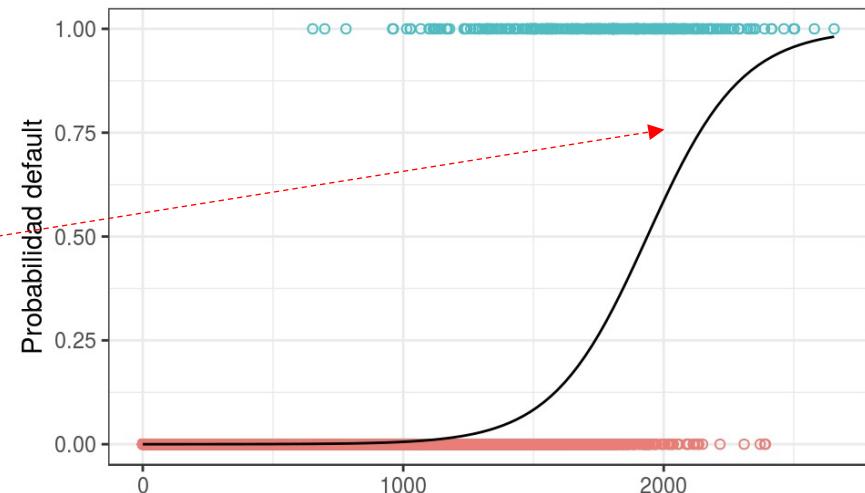
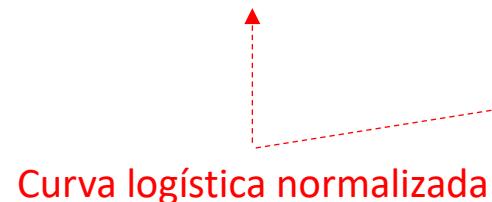
- Se utiliza la misma estructura que la regresión lineal, pero se **transforma** la variable respuesta (0 o 1) en una probabilidad.
- Para esta transformación se utiliza la función logística (conocida también como sigmoide).



Regresión Logística

Función logística

$$\text{Función logística} = \frac{1}{1+e^{-x}} = \frac{1}{1+e^{-(a+bX)}}$$



e es conocido como el **número de Euler**, por Leonhard Euler.

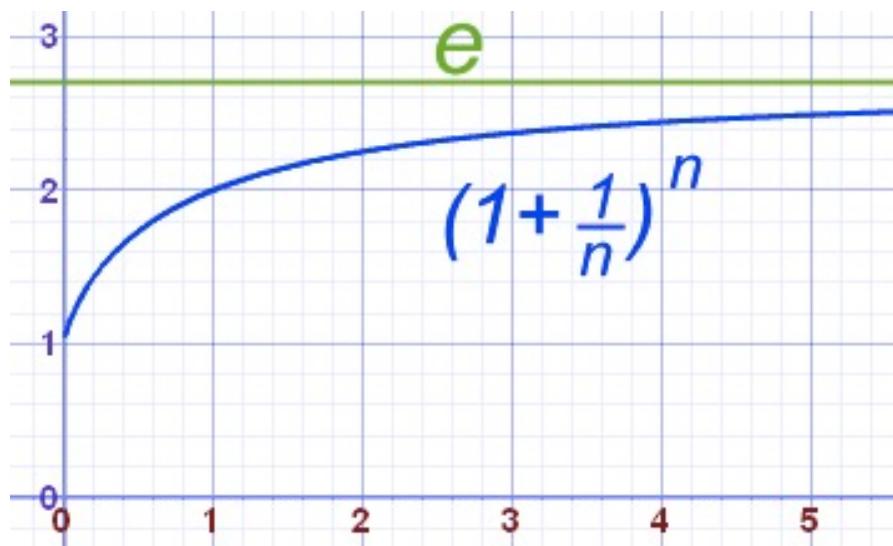
e es una constante matemática, que es la base del logaritmo natural (inventado por John Napier).

e es aproximadamente 2.718281828

Regresión Logística

Número de Euler

- Euler definió una función exponencial a través de una función inversa: $(1 + 1/n)^n$
- El propósito fue tener una función con diversas aplicaciones.
- Usos actuales: el área cubierta por una hipérbola (e^x), el interés compuesto continuo ($C_0 e^{-rt}$) y otros.



n	$(1 + 1/n)^n$
1	2.00000
2	2.25000
5	2.48832
10	2.59374
100	2.70481
1000	2.71692
10000	2.71815
100000	2.71827

Procedimiento

Regresión Logística

Paso 1

Se establece transformar Y en una probabilidad \hat{Y} a partir de una función.

Nro	X	Y
1	25	0
2	54	1
3	67	1
4	6	0
5	77	1
6	32	0
7	81	1
8	24	0
9	15	0
10	40	1



$$Y = a + b_i X_i + u$$

A esta transformación también se conoce como *razón de probabilidad de ser verdadero (Odds Ratio)*.

$$\text{Probabilidad} = \hat{Y} = \frac{1}{1 + e^{-(a+bX)}}$$



Regresión Logística

Paso 2

Se calcula la regresión lineal para predecir la probabilidad: $a + b_i X_i$

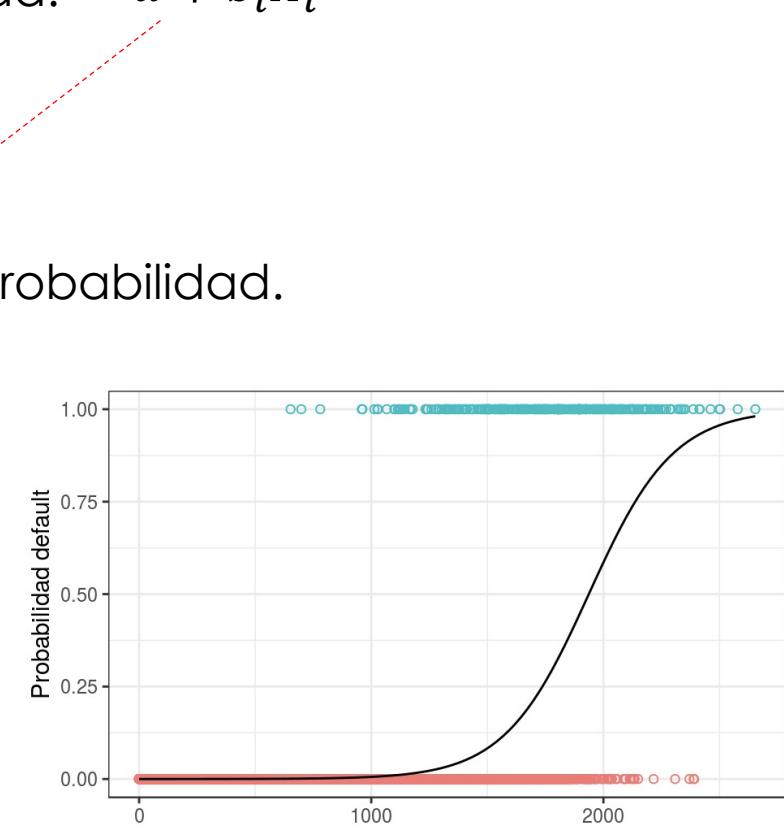
Paso 3

Se transforma el resultado de la regresión lineal en una probabilidad.

$$\frac{1}{1+e^{-x}} = \frac{1}{1+e^{-(a+b_i X_i)}}$$

Donde $e = 2.718281828$

- Si la probabilidad es mayor a **0.5** se asigna **1**.
- Si es menor o igual a **0.5** se asigna **0**.



Regresión Logística

En resumen

Datos		
Nro	X	Y
1	25	0
2	54	1
3	67	1
4	6	0
5	77	1
6	32	0
7	81	1
8	24	0
9	15	0
10	40	1

$$\rightarrow Y = a + b_i X_i + u$$

Paso 1

- Se establece transformar Y en una probabilidad \hat{Y}

$$\hat{Y} = \frac{1}{1 + e^{-(a+b_i X_i)}}$$

Paso 2

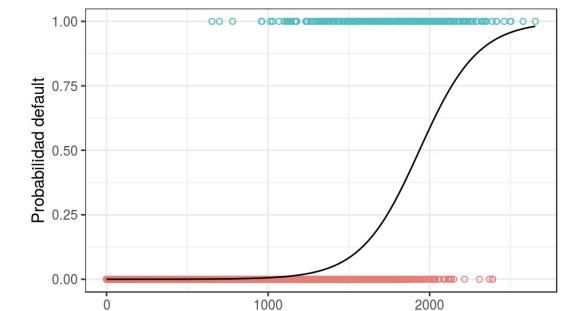
Se calcula la regresión lineal.

Paso 3

$$\frac{1}{1 + e^{-(a+b_i X_i)}}$$

Se transforma el resultado en una probabilidad final.

Regresión logística



Ejemplo

Ejemplo

Sean dos variables:

Datos		
Nro	X	Y
1	22	0
2	54	1
3	67	1
4	6	0
5	77	1
6	32	0
7	81	1
8	4	0
9	5	0
10	40	1
11		
12		
13		
14		
15		

38.80 0.50

$$\text{Probabilidad} = \hat{Y} = \frac{1}{1 + e^{-(\alpha + bX)}}$$

Intercepto Pendiente Variable independiente
Punto de corte Coeficiente Variable predictora

Ejemplo

Solución: **Paso 1**

<i>Datos</i>		
Nro	X	Y
1	22	0
2	54	1
3	67	1
4	6	0
5	77	1
6	32	0
7	81	1
8	4	0
9	5	0
10	40	1
11		
12		
13		
14		
15		

38.80 0.50

1	2	3
Sx	Sy	Sxy
282.24	0.25	8.40
231.04	0.25	7.60
795.24	0.25	14.10
1075.84	0.25	16.40
1459.24	0.25	19.10
46.24	0.25	3.40
1780.84	0.25	21.10
1211.04	0.25	17.40
1142.44	0.25	16.90
1.44	0.25	0.60

28.33 0.50 12.50

$$S_x = \sqrt{\sum_{i=1}^n \frac{(X_i - \bar{X})^2}{n}}$$

$$S_y = \sqrt{\sum_{i=1}^n \frac{(Y_i - \bar{Y})^2}{n}}$$

$$S_{xy} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n}$$

Ejemplo

Solución: **Paso 1**

<i>Datos</i>		
Nro	X	Y
1	22	0
2	54	1
3	67	1
4	6	0
5	77	1
6	32	0
7	81	1
8	4	0
9	5	0
10	40	1
11		
12		
13		
14		
15		
		38.80 0.50

1	2	3
Sx	Sy	Sxy
282.24	0.25	8.40
231.04	0.25	7.60
795.24	0.25	14.10
1075.84	0.25	16.40
1459.24	0.25	19.10
46.24	0.25	3.40
1780.84	0.25	21.10
1211.04	0.25	17.40
1142.44	0.25	16.90
1.44	0.25	0.60
		28.33 0.50 12.50

4

Pendiente (b)

$$b = \frac{S_{xy}}{S_x^2}$$

$$b = \frac{12.5}{(28.33)^2} = \frac{12.5}{802.59}$$

$$\color{blue}{b = 0.016}$$

Ejemplo

Solución: **Paso 1**

Datos		
Nro	X	Y
1	22	0
2	54	1
3	67	1
4	6	0
5	77	1
6	32	0
7	81	1
8	4	0
9	5	0
10	40	1
11		
12		
13		
14		
15		
		38.80 0.50

1	2	3
Sx	Sy	Sxy
282.24	0.25	8.40
231.04	0.25	7.60
795.24	0.25	14.10
1075.84	0.25	16.40
1459.24	0.25	19.10
46.24	0.25	3.40
1780.84	0.25	21.10
1211.04	0.25	17.40
1142.44	0.25	16.90
1.44	0.25	0.60
		28.33 0.50 12.50

4 Pendiente (*b*)

$$b = 0.016$$

5 Intercepto (*a*)

$$Y = a + bX$$

$$a = \bar{y} - b\bar{x}$$

$$a = 0.5 - (0.016 * 38.8)$$

$$a = 0.5 - 0.62$$

$$a = -0.12$$

Ejemplo

Solución: **Paso 2**

<i>Datos</i>		
Nro	X	Y
1	22	0
2	54	1
3	67	1
4	6	0
5	77	1
6	32	0
7	81	1
8	4	0
9	5	0
10	40	1
11		
12		
13		
14		
15		
		38.80 0.50

Sx	Sy	Sxy	Ŷ (Pronóstico)
282.24	0.25	8.40	0.24
231.04	0.25	7.60	0.74
795.24	0.25	14.10	0.94
1075.84	0.25	16.40	-0.01
1459.24	0.25	19.10	1.09
46.24	0.25	3.40	0.39
1780.84	0.25	21.10	1.16
1211.04	0.25	17.40	-0.04
1142.44	0.25	16.90	-0.03
1.44	0.25	0.60	0.52
		28.33 0.50	12.50

6

Pronóstico: \hat{Y}

$$Y_i = a + bX_i$$

$$\hat{Y}_1 = -0.12 + 0.016(22)$$

$$\hat{Y}_1 = \mathbf{0.24}$$

$$\hat{Y}_2 = -0.12 + 0.016(54)$$

$$\hat{Y}_2 = -0.12 + 0.864$$

$$\hat{Y}_2 = \mathbf{0.74}$$

$$\hat{Y}_3 = -0.12 + 0.016(67)$$

$$\hat{Y}_3 = -0.12 + 1.072$$

$$\hat{Y}_3 = \mathbf{0.94}$$

...

Ejemplo

Solución: **Paso 3**

Datos		
Nro	X	Y
1	22	0
2	54	1
3	67	1
4	6	0
5	77	1
6	32	0
7	81	1
8	4	0
9	5	0
10	40	1
11		
12		
13		
14		
15		
		38.80 0.50

Sx	Sy	Sxy	Ŷ (Pronóstico)	Prob
282.24	0.25	8.40	0.24	0.56
231.04	0.25	7.60	0.74	0.68
795.24	0.25	14.10	0.94	0.72
1075.84	0.25	16.40	-0.01	0.50
1459.24	0.25	19.10	1.09	0.75
46.24	0.25	3.40	0.39	0.60
1780.84	0.25	21.10	1.16	0.76
1211.04	0.25	17.40	-0.04	0.49
1142.44	0.25	16.90	-0.03	0.49
1.44	0.25	0.60	0.52	0.63
		28.33 0.50 12.50		

7

$$\frac{1}{1 + e^{-(a+b_iX_i)}}$$

$$Prob_1 = 1/(1+e^{-(0.25)})$$

$$Prob_1 = 1/(1+ 2.718281828^{-0.24})$$

$$\textcolor{blue}{Prob}_1 = 0.56$$

$$Prob_2 = 1/(1+ 2.718281828^{-0.74})$$

$$\textcolor{blue}{Prob}_2 = 0.68$$

$$Prob_3 = 1/(1+ 2.718281828^{-0.94})$$

$$\textcolor{blue}{Prob}_3 = 0.72$$

...

Ejemplo

Solución: **Paso 3**

Datos		
Nro	X	Y
1	22	0
2	54	1
3	67	1
4	6	0
5	77	1
6	32	0
7	81	1
8	4	0
9	5	0
10	40	1
11		
12		
13		
14		
15		
		38.80 0.50

Sx	Sy	Sxy	Ŷ (Pronóstico)	Prob	Clase
282.24	0.25	8.40	0.24	0.56	1
231.04	0.25	7.60	0.74	0.68	1
795.24	0.25	14.10	0.94	0.72	1
1075.84	0.25	16.40	-0.01	0.50	0
1459.24	0.25	19.10	1.09	0.75	1
46.24	0.25	3.40	0.39	0.60	1
1780.84	0.25	21.10	1.16	0.76	1
1211.04	0.25	17.40	-0.04	0.49	0
1142.44	0.25	16.90	-0.03	0.49	0
1.44	0.25	0.60	0.52	0.63	1
		28.33 0.50 12.50			

8

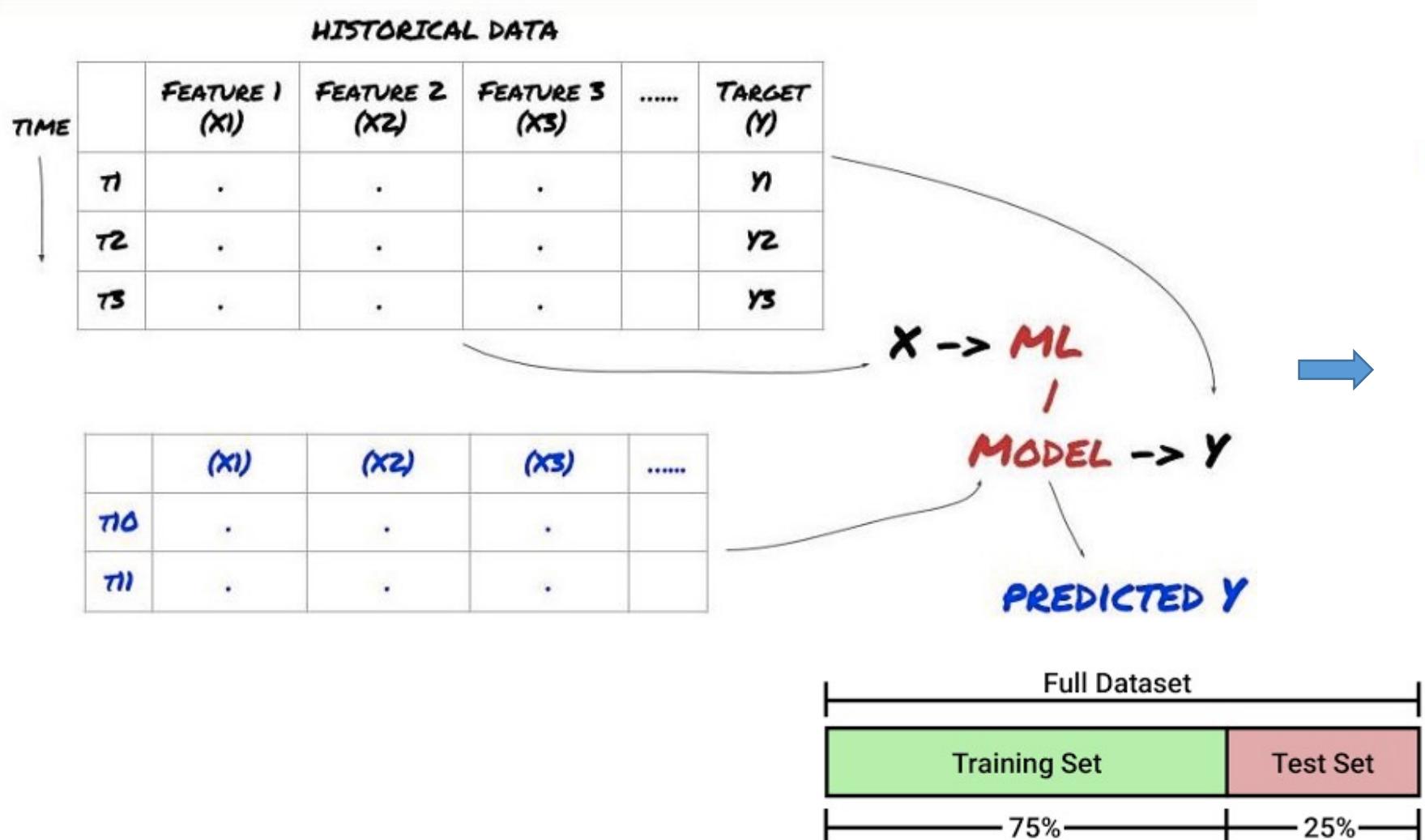
Clase

- Si la probabilidad es superior a 0.5 se asigna **1**.
- Si es menor o igual a 0.5 se asigna **0**.

Validación de la clasificación

1. Matriz de clasificación

Contexto



Matriz de clasificación

- Una matriz de clasificación (matriz de confusión), se utiliza para evaluar una clasificación.
- En la variable clase el conjunto de entrenamiento toma dos valores posibles: 0 o 1; positivo o negativo; falso o verdadero.
- Los valores positivos y negativos que se predicen correctamente se conocen como **verdaderos positivos (VP)** y **verdaderos negativos (VN)**, respectivamente.
- Mientras que los valores clasificados incorrectamente se denominan **falsos positivos (FP)** y **falsos negativos (FN)**.

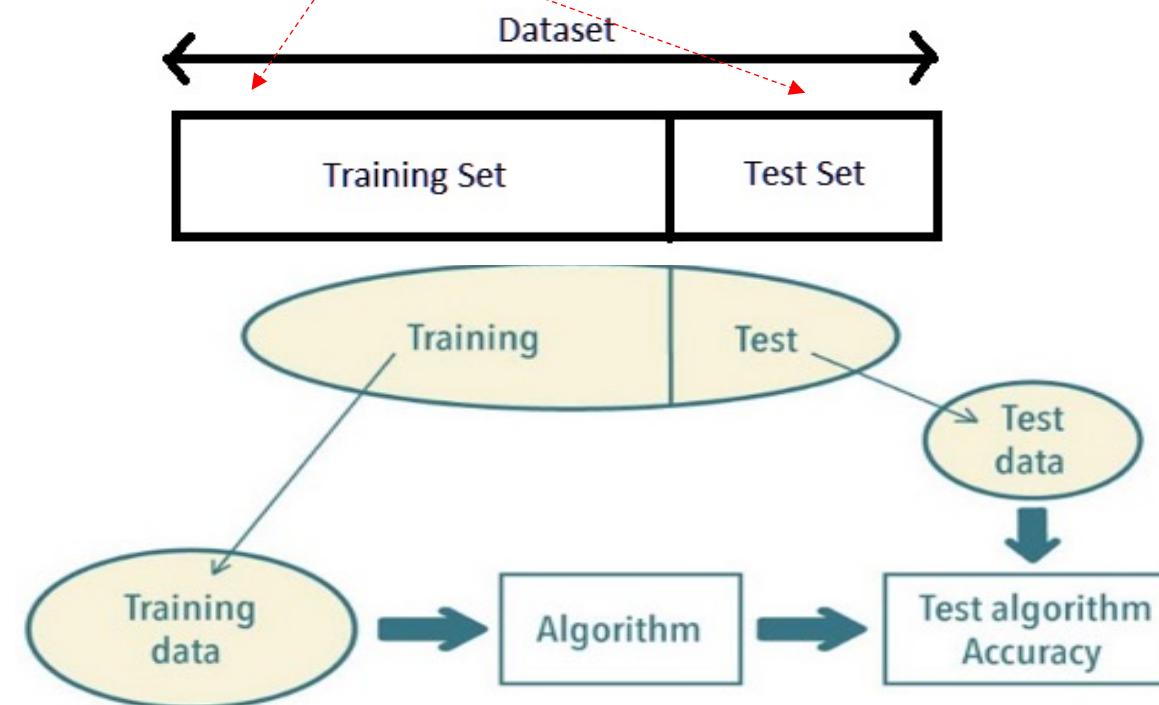
		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Matriz de clasificación

Criterio de división

Para utilizar este método de evaluación del modelo se necesita dividir los datos en:

- a) Datos de entrenamiento (*training*): 80, 75, o 70%
- b) Datos de prueba (*test*): 20, 25, o 30%



Matriz de clasificación

Procedimiento

- 1) Se evalúan todos los elementos y se determina si la **predicción (clase)** coincide con los **valores reales (Y)**.
- 2) Se cuentan todos los elementos y se muestran los totales obtenidos en la matriz.

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Matriz de clasificación

Mediciones

- 1) Exactitud (Accuracy)
- 2) Tasa de error (Misclassification Rate)
- 3) Precisión (Precision)
- 4) Sensibilidad (Recall, Sensitivity, True Positive Rate)
- 5) Especificidad (Especificity, True Negative Rate)

Matriz de clasificación

1) **Exactitud (Accuracy).** Es el porcentaje de datos clasificados correctamente.

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

$$\text{Exactitud} = \frac{VP + VN}{Total} = \frac{VP + VN}{VP + VN + FP + FN}$$

Matriz de clasificación

2) **Precisión (Precision).** Es el porcentaje de clasificación positiva.

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

$$\text{Precisión} = \frac{VP}{\text{Total clasificados positivos}} = \frac{VP}{VP + FP}$$

Matriz de clasificación

3) **Tasa de error (Misclassification Rate).** Porcentaje de datos clasificados incorrectamente.

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

$$\text{Tasa de error} = \frac{FP + FN}{Total} = \frac{FP + FN}{VP + VN + FP + FN}$$

Matriz de clasificación

4) **Sensibilidad (True Positive Rate).** Es el porcentaje de clasificación del total positivos.

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

$$\text{Sensibilidad} = \frac{VP}{\text{Total positivos}} = \frac{VP}{VP + FN}$$

Matriz de clasificación

5) **Especificidad (True Negative Rate).** Es el porcentaje de clasificación del total negativos.

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

$$\text{Especificidad} = \frac{VN}{\text{Total negativos}} = \frac{VN}{VN + FP}$$

Matriz de clasificación

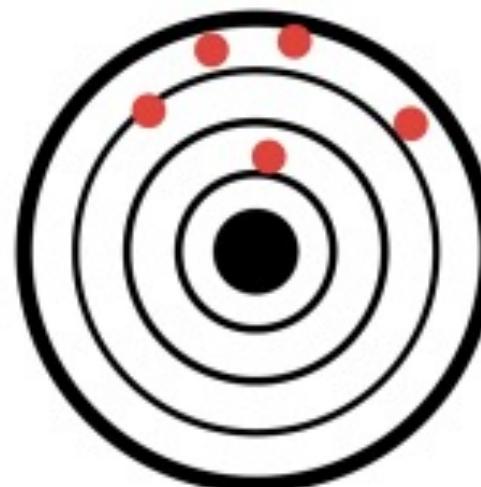
Eficiencia y precisión



Baja eficiencia pero
buena precisión



Buena eficiencia y
buena precisión



Baja eficiencia y
baja precisión

La **exactitud** simboliza el grado de conformidad, mientras que la **precisión** indica el grado de reproducibilidad.

Matriz de clasificación

Ejemplo 1

n = 2000

		Predicción	
		Positivo	Negativo
Observado (Real)	Positivo	VP 1100	FN 100
	Negativo	FP 60	VN 740

$$\text{Exactitud} = \frac{VP+VN}{VP+VN+FP+FN} = \frac{1100+740}{1100+740+60+100} = \frac{1840}{2000} = \mathbf{0.92}$$

$$\text{Precisión} = \frac{VP}{VP+FP} = \frac{1100}{1100+60} = \frac{1100}{1160} = \mathbf{0.95}$$

$$\text{Tasa de error} = \frac{FP+FN}{VP+VN+FP+FN} = \frac{60+100}{1100+740+60+100} = \frac{160}{2000} = \mathbf{0.08}$$

$$\text{Sensibilidad} = \frac{VP}{VP+FN} = \frac{1100}{1100+100} = \frac{1100}{1200} = \mathbf{0.916}$$

$$\text{Especificidad} = \frac{VN}{VN+FP} = \frac{740}{740+60} = \frac{740}{800} = \mathbf{0.925}$$

Matriz de clasificación

Ejemplo 2

$n = 10000$

		Predicción	
		Positivo	Negativo
Observado (Real)	Positivo	VP 3200	FN 340
	Negativo	FP 240	VN 6220

$$\text{Exactitud} = \frac{VP+VN}{VP+VN+FP+FN} = \frac{3200+6220}{3200+6220+340+240} = \frac{9420}{10000} = \mathbf{0.942}$$

$$\text{Precisión} = \frac{VP}{VP+FP} = \frac{3200}{3200+240} = \frac{3200}{3440} = \mathbf{0.93}$$

$$\text{Tasa de error} = \frac{FP+FN}{VP+VN+FP+FN} = \frac{240+340}{3200+6220+340+240} = \frac{580}{10000} = \mathbf{0.058}$$

$$\text{Sensibilidad} = \frac{VP}{VP+FN} = \frac{3200}{3200+340} = \frac{3200}{3540} = \mathbf{0.90}$$

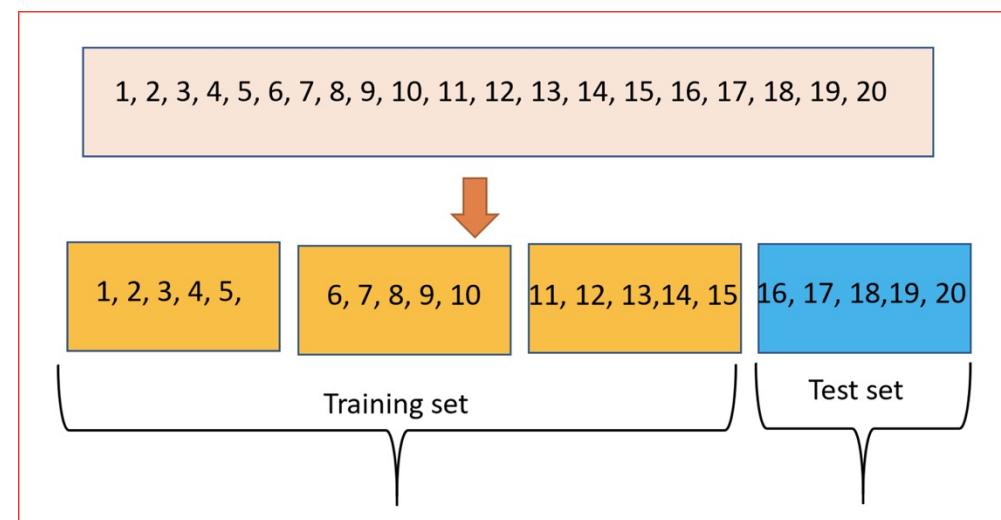
$$\text{Especificidad} = \frac{VN}{VN+FP} = \frac{6220}{6220+240} = \frac{6220}{6460} = \mathbf{0.96}$$

2. Validación cruzada

Validación cruzada

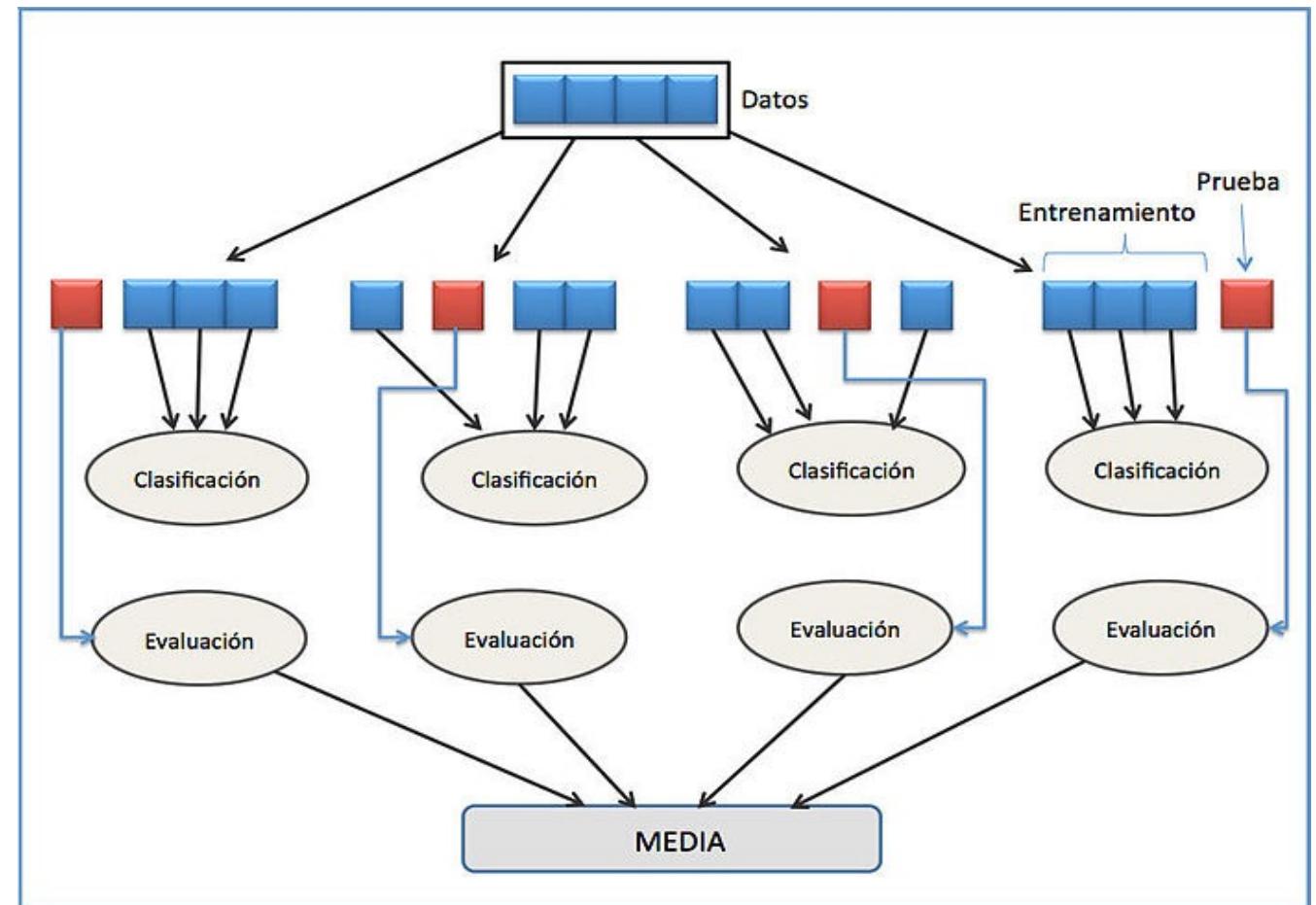
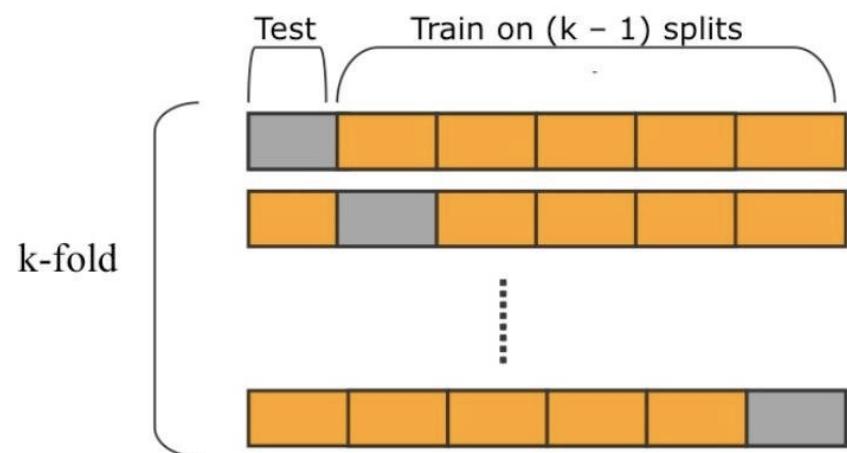
Consiste en dividir los datos en dos conjuntos: **entrenamiento** (*training*) y **prueba** (*test*).

- La clasificación se ajusta a un conjunto de datos de entrenamiento.
- Posteriormente, se calcula los valores de salida con los datos de prueba (**valores que no se han analizado antes**).
- La evaluación depende de la **división** entre los datos de entrenamiento y de prueba.



Validación cruzada

Para esta división se emplea el concepto de '**Cross validation**' de 'K' iteraciones.



Validación cruzada

Ejemplo

Se desea predecir la desafiliación de clientes y se requiere saber la eficiencia de la predicción. Una forma de lograr esto es mediante la **validación cruzada**.

	Plan_Internaci onal	Min_En_Dia	Min_Internaci onales	Reclamos	Llamadas_Int ernacionales	Desafiliado
TEST	no	265.1	10	1	3	no
	no	129.1	12.7	4	6	yes
	no	123	3	2	8	no
Entrenamiento	no	116.9	7	0	10	yes
	no	119.1	2.9	2	12	no
	no	187.7	9.1	0	5	no
	no	128.8	11.2	1	2	no
	no	156.6	12.3	3	5	no
	no	332.9	5.4	4	9	yes

Iteración 1

$$\frac{3}{3} = 100\% \text{ eficiencia en predicción}$$

Validación cruzada

Ejemplo

	Plan_Internaci onal	Min_En_Dia	Min_Internaci onales	Reclamos	Llamadas_Int ernacionales	Desafiliado
Entren...	no	265.1	10	1	3	no
TST	no	129.1	12.7	4	6	yes
Entren...	no	123	3	2	8	no
	no	116.9	7	0	10	yes
	no	119.1	2.9	2	12	no
	no	187.7	9.1	0	5	no
	no	128.8	11.2	1	2	no
	no	156.6	12.3	3	5	no
	no	332.9	5.4	4	9	yes

Iteración 2

$$\frac{1}{3} = 33\% \text{ eficiencia en predicción}$$

Validación cruzada

Ejemplo

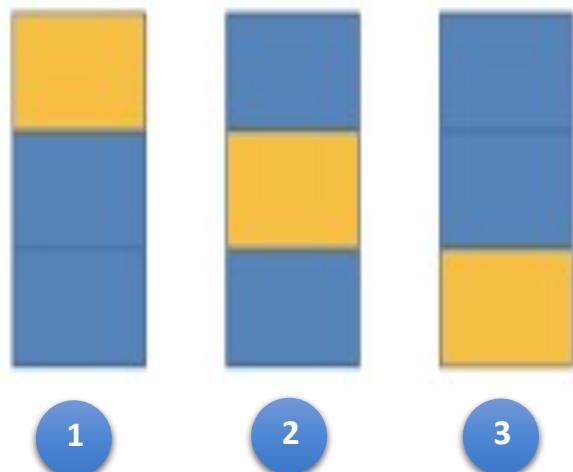
	Plan_Internacional	Min_En_Dia	Min_Internacionales	Reclamos	Llamadas_Internacionales	Desafiliado
Entrenamiento	no	265.1	10	1	3	no
	no	129.1	12.7	4	6	yes
	no	123	3	2	8	no
	no	116.9	7	0	10	yes
	no	119.1	2.9	2	12	no
	no	187.7	9.1	0	5	no
TBT	no	128.8	11.2	1	2	no
	no	156.6	12.3	3	5	no
	no	332.9	5.4	4	9	yes

Iteración 3

$\frac{2}{3} = 66\%$ eficiencia en predicción

Validación cruzada

Ejemplo



Iteración 1: $3/3 = 100\%$

Iteración 2: $1/3 = 33\%$

Iteración 3: $2/3 = 66\%$

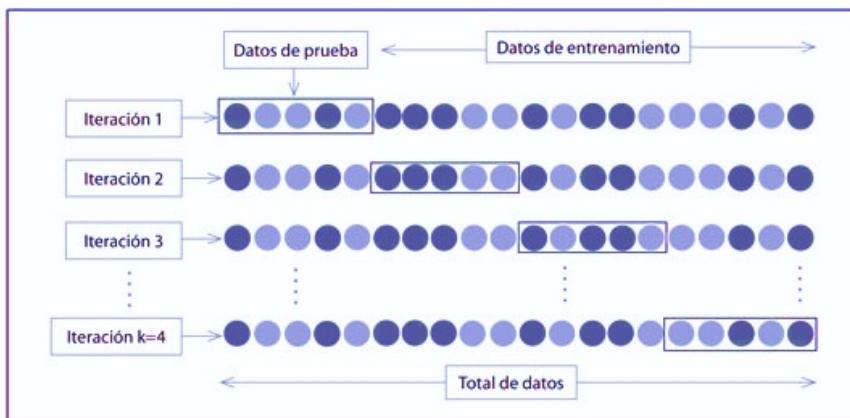
Promedio: 66.3%

Por lo tanto, la eficiencia de la predicción es de 66.3%

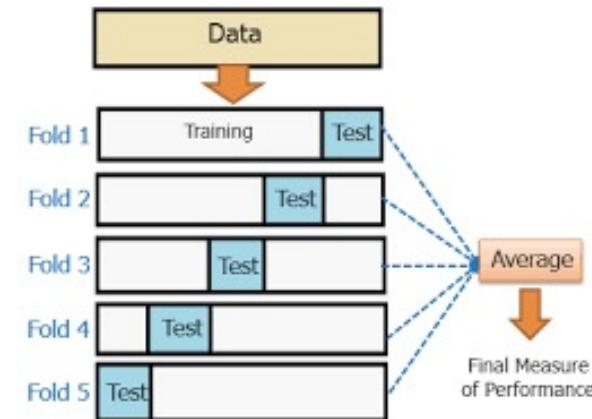
Validación cruzada

Divisiones comunes

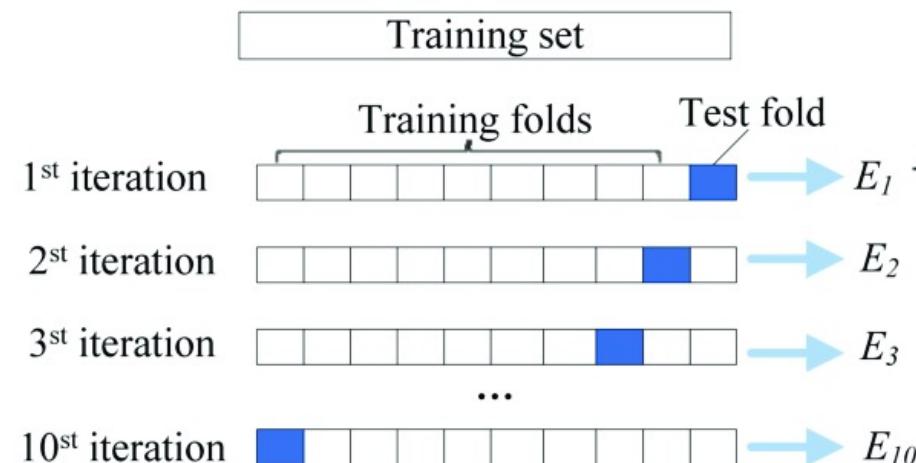
K = 4



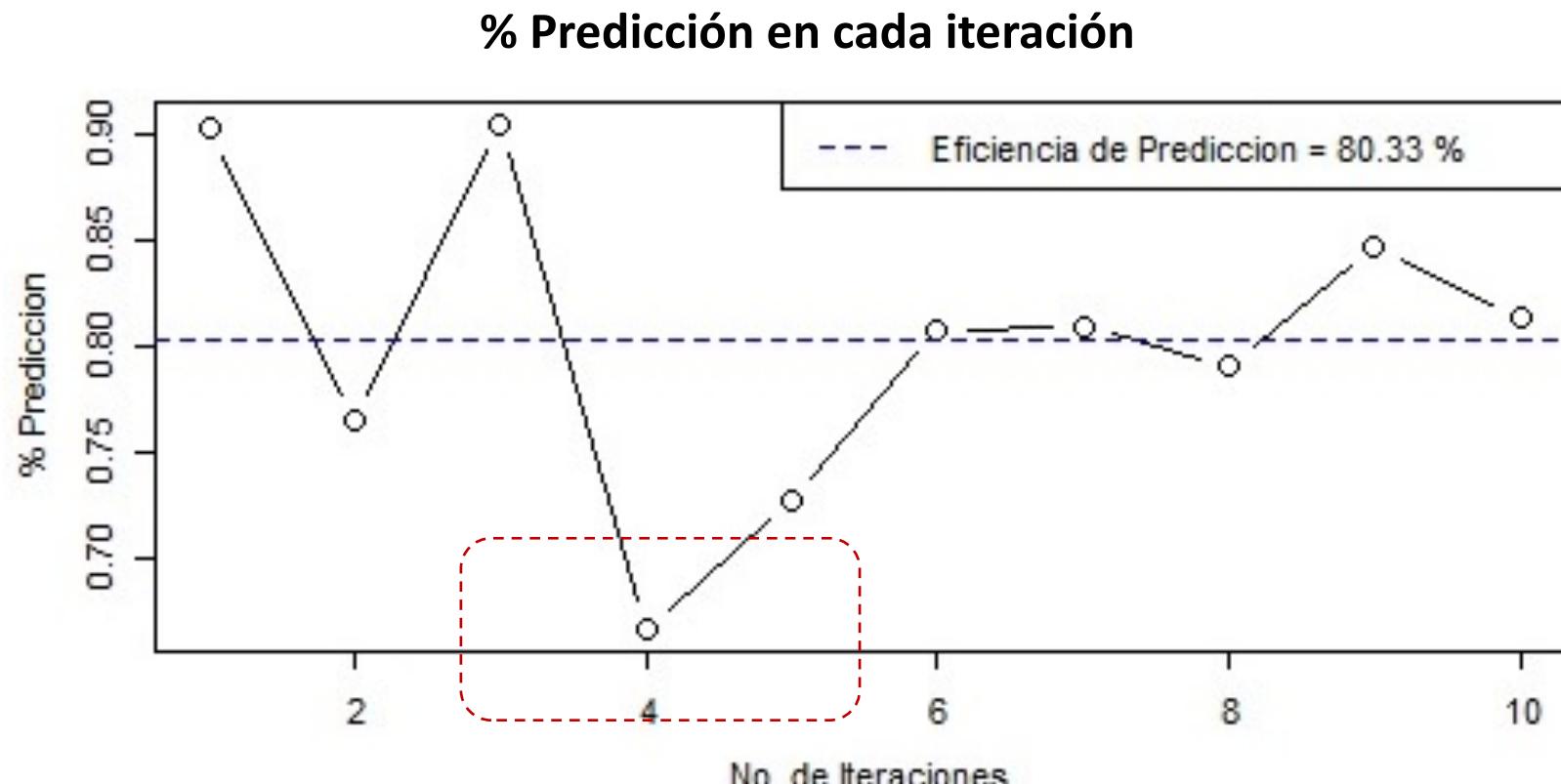
K = 5



K = 10



Validación cruzada

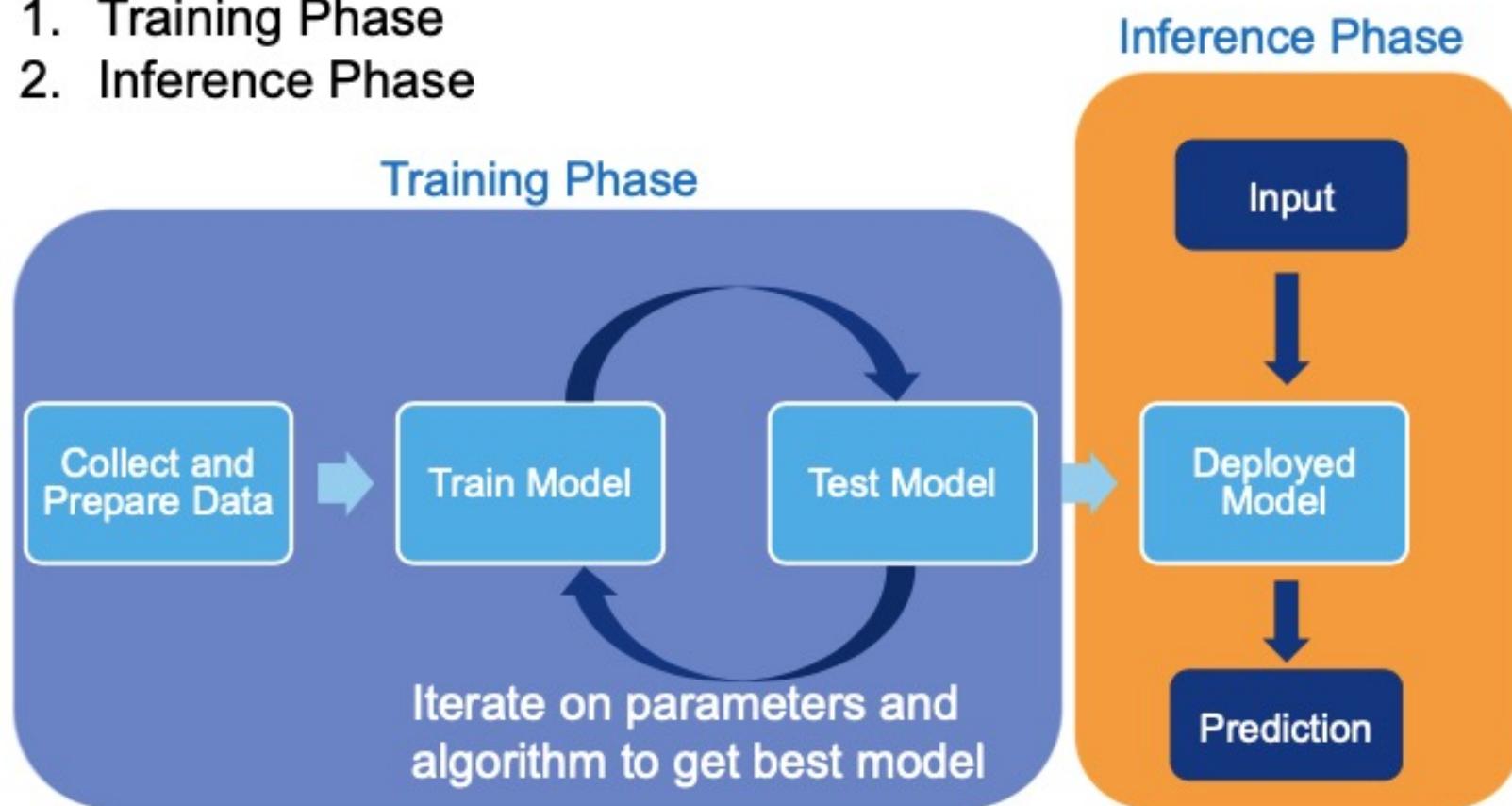


El % de predicción de cada iteración puede graficarse a partir del promedio de la eficiencia general del modelo predictivo.

Lo que se busca

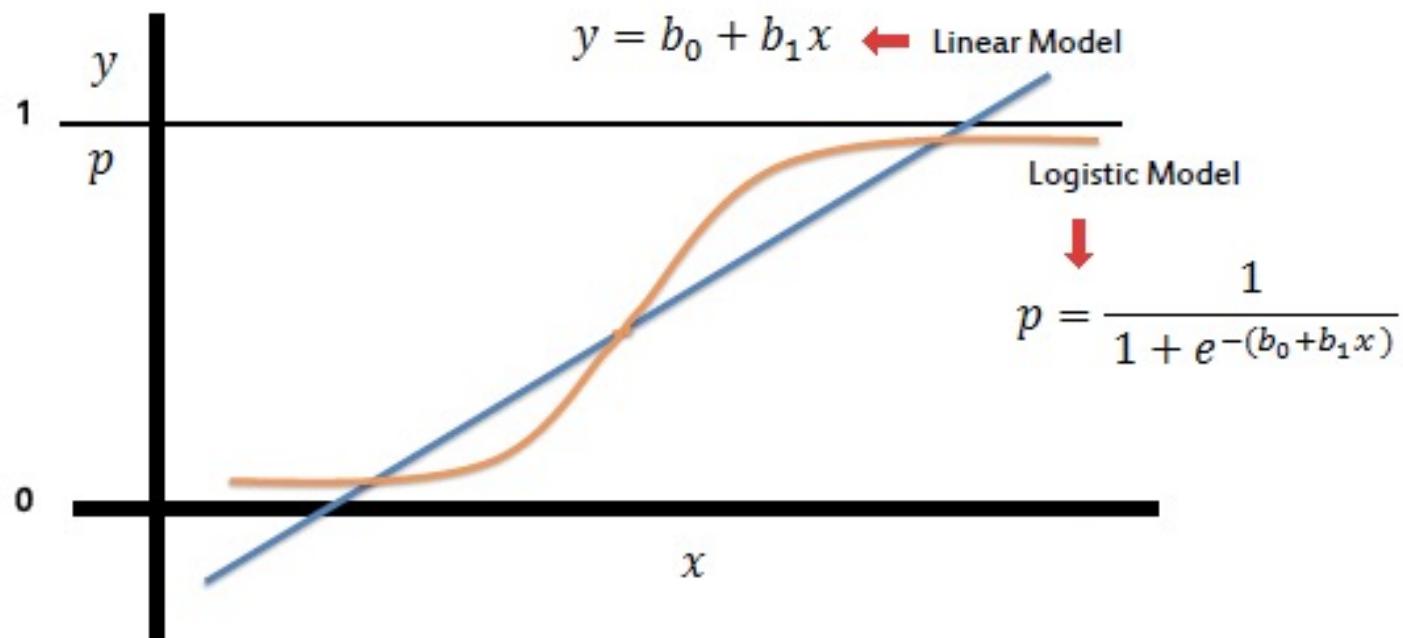
Regresión logística

1. Training Phase
2. Inference Phase

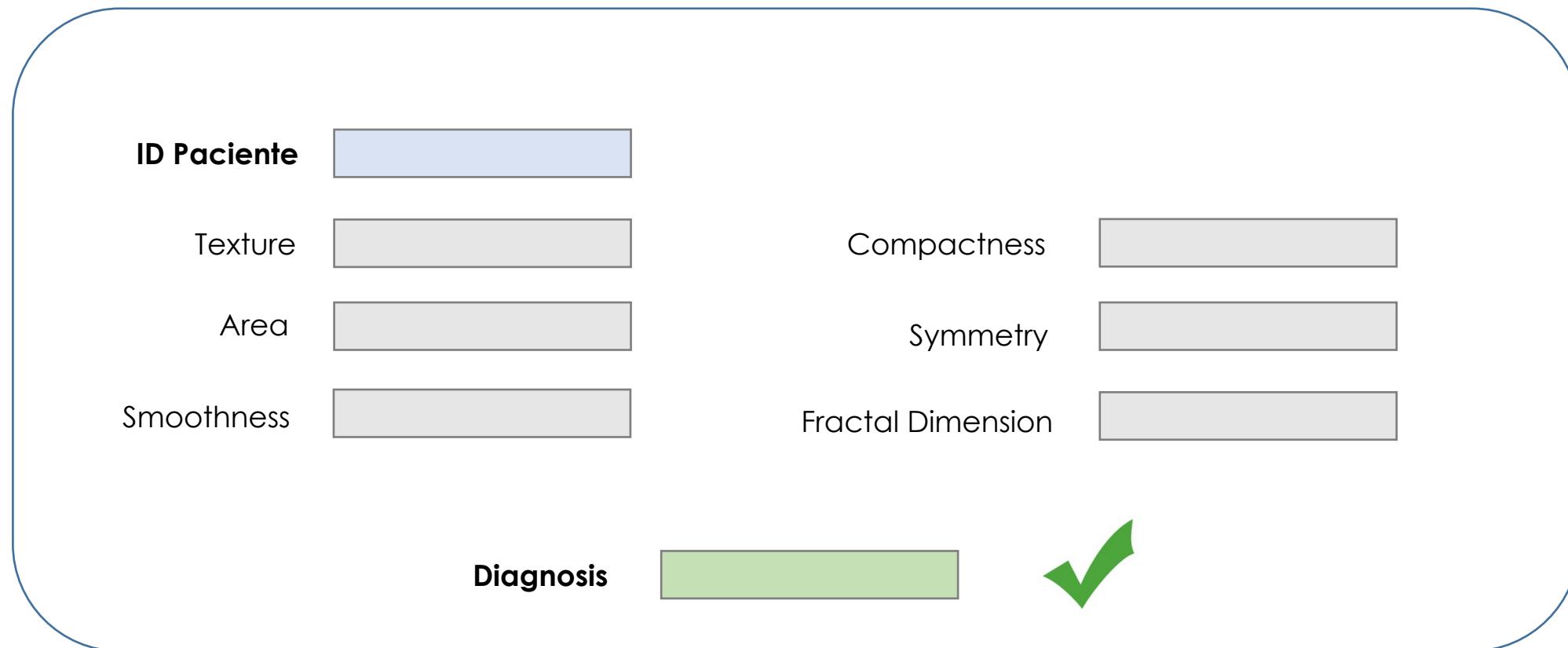


Regresión logística

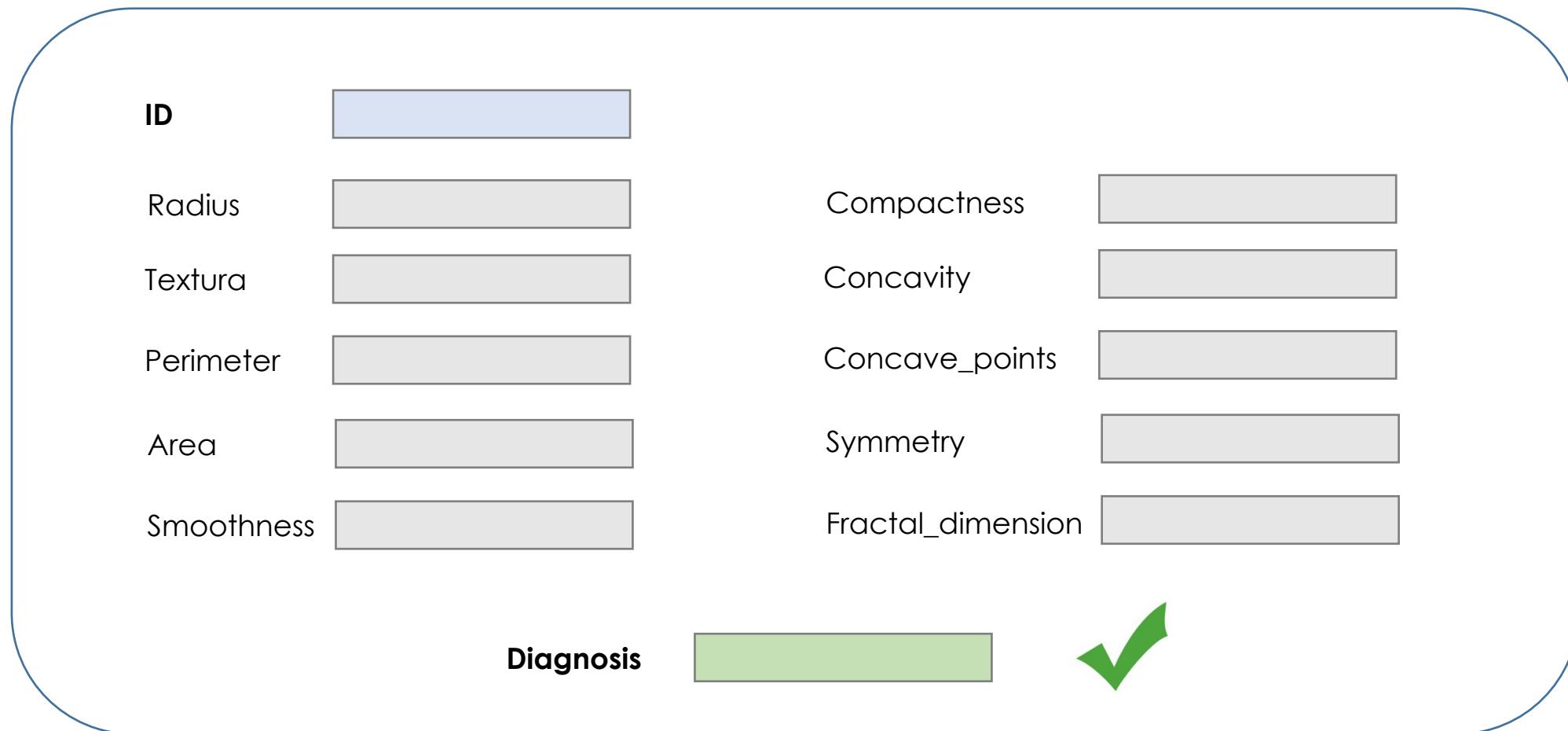
- Diagnóstico (variable dependiente)



Sistemas de inferencia basadas en modelos de predicción



Sistemas de inferencia basadas en modelos de predicción





Universidad Nacional Autónoma de México
Facultad de Ingeniería

Árboles de decisión

Guillermo Molero-Castillo

guillermo.molero@ingenieria.unam.edu

Noviembre, 2021

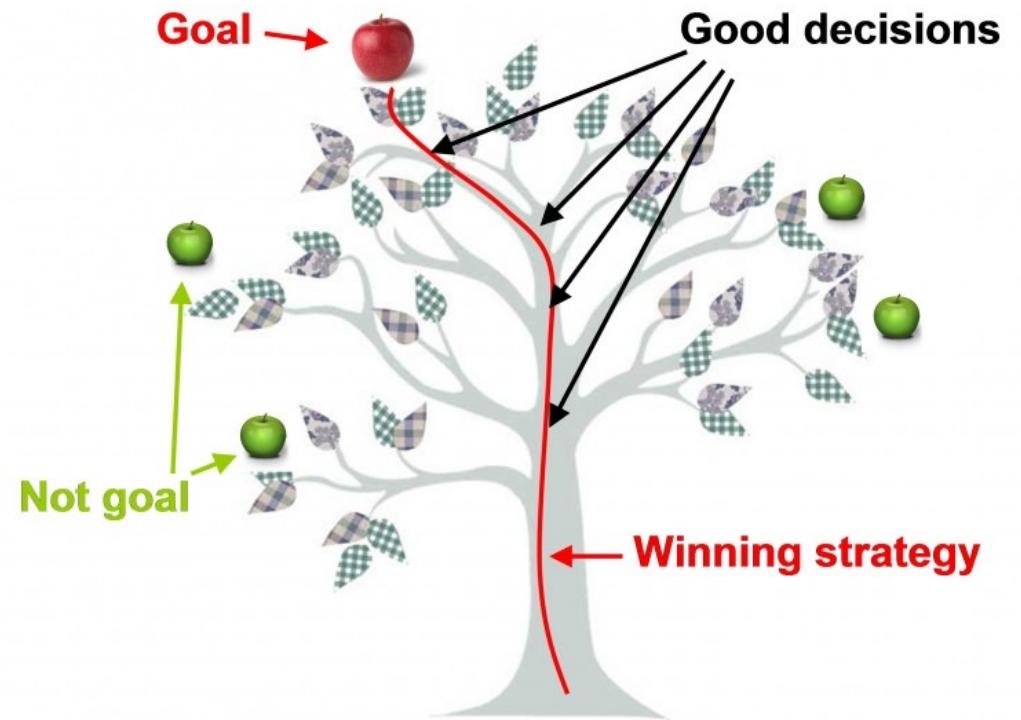
Contexto



Árboles de decisión

Árboles de decisión

- Es uno de los algoritmos más utilizados en el aprendizaje automático supervisado.
- Permiten resolver problemas de regresión (pronóstico) y clasificación.
- Admiten valores **numéricos y nominales**.
- Aportan claridad (despliegan los resultados en profundidad, de mayor a menor detalle).
- Tienen buena precisión en un amplio número de aplicaciones.

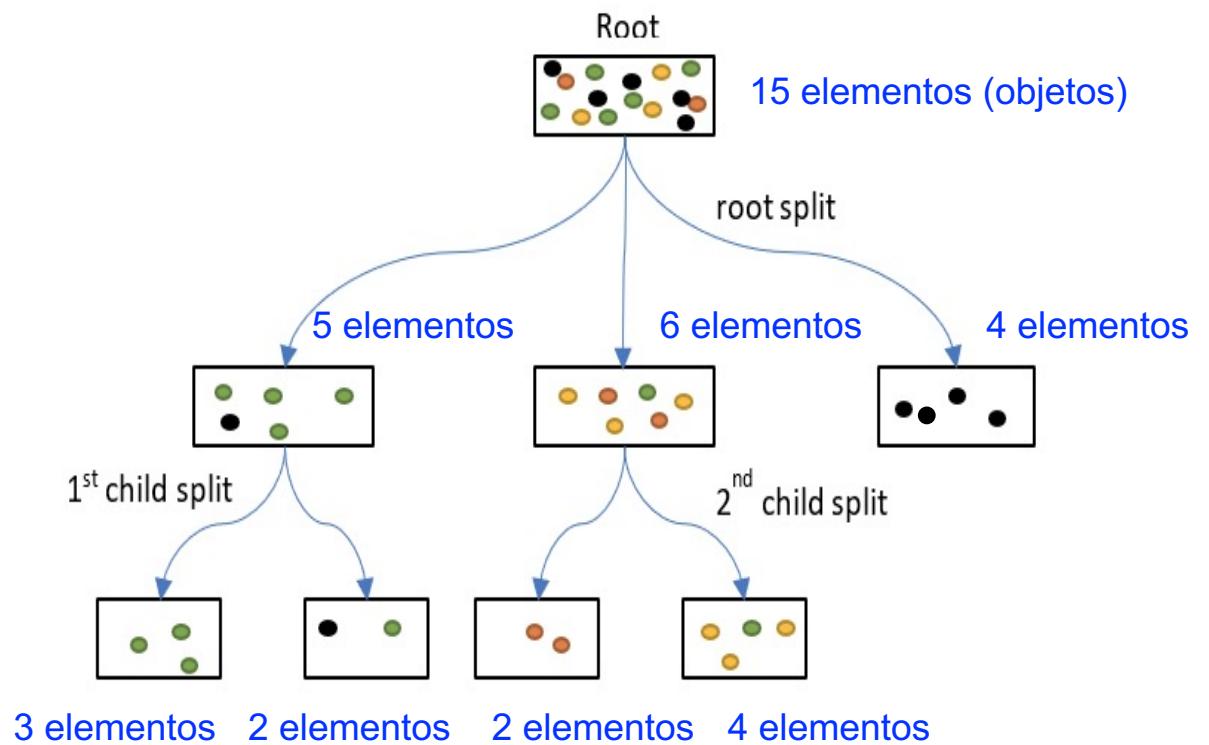
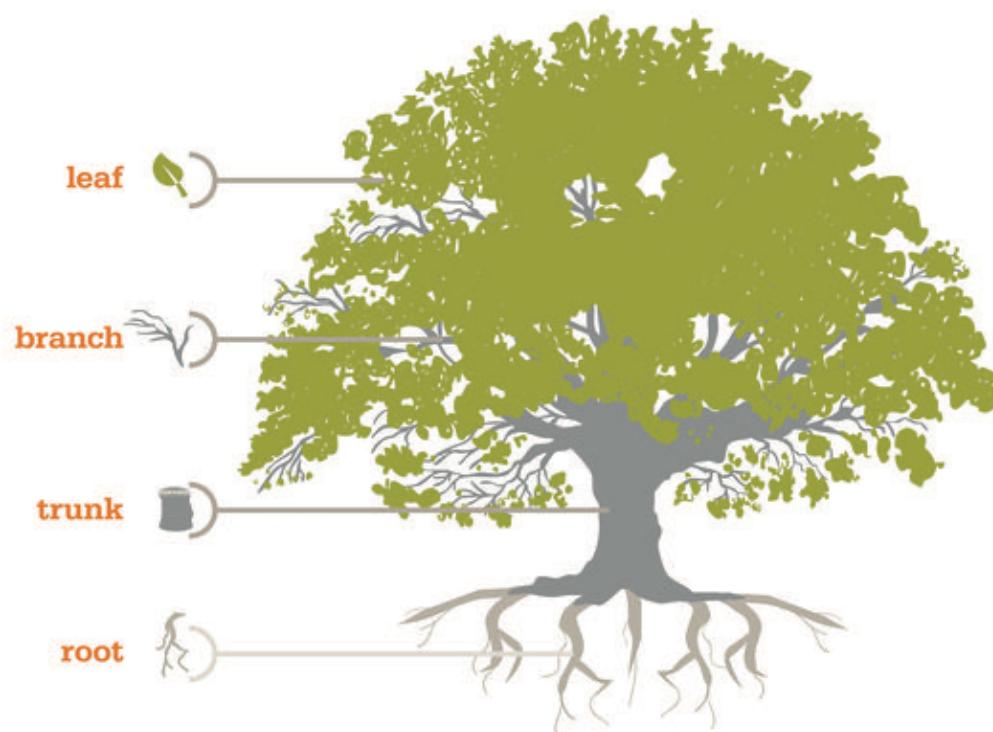


Nota. En la vida real muchas veces somos más cualitativos que cuantitativos. Utilizamos etiquetas del lenguaje natural.

Árboles de decisión

Objetivo. Construir estructura jerárquica eficiente y escalable que divide los datos en función de determinadas condiciones. Para esto se utiliza la estrategia: divide y vencerás.

Para cada variable, el algoritmo forma un nodo, donde la variable más importante se coloca en el nodo raíz.



Árboles de decisión

La construcción de un árbol de decisión puede parecer compleja, pero en realidad hemos estado utilizando árboles de decisiones durante nuestra vida para tomar decisiones. Por ejemplo:

- Si una persona nos pide prestado el coche (automóvil). Se tiene que tomar una decisión.
- Hay varios factores (condiciones) que ayudan a determinar la decisión (prestar o no):
 1. ¿Esta persona es un amigo cercano o solo un conocido? Si solo es un conocido, se rechaza la solicitud. Si es un amigo, se pasa a la siguiente condición.
 2. ¿Es la primera vez que pide prestado el coche? Si es así, se le presta el coche; de lo contrario, se pasa a la siguiente condición.
 3. ¿Se dañó el coche la última vez que lo devolvieron? Si es así, se rechaza la solicitud; si no, se presta el coche.



Árboles de decisión

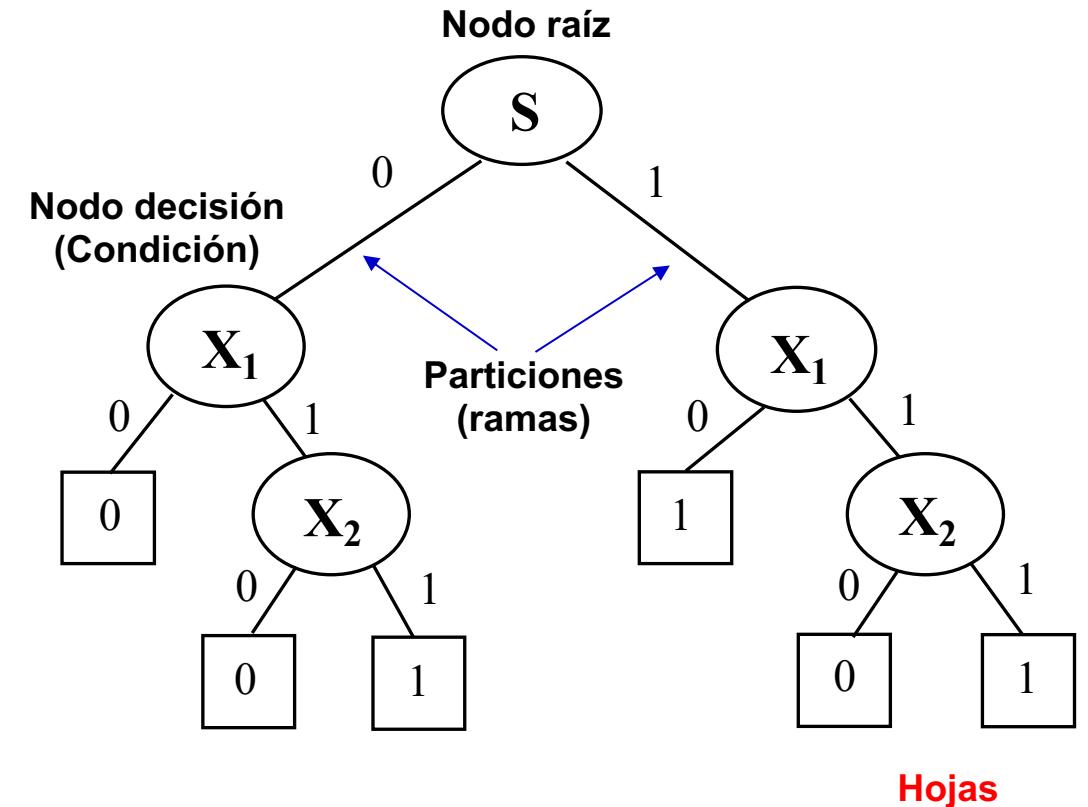
Estructura

Nodo principal. Representa toda la población (todos los datos) que posteriormente se dividirá. También cuenta como un nodo de decisión.

Nodo de decisión. Se encarga de dividir los datos, dependiendo de una decisión, por ejemplo, 'Peso', mayores de 50kg y menores o iguales a 50kg. Se toman dos caminos.

Nodo hoja. Es donde recae la decisión final. Por ejemplo, si una persona tiene diabetes o no.

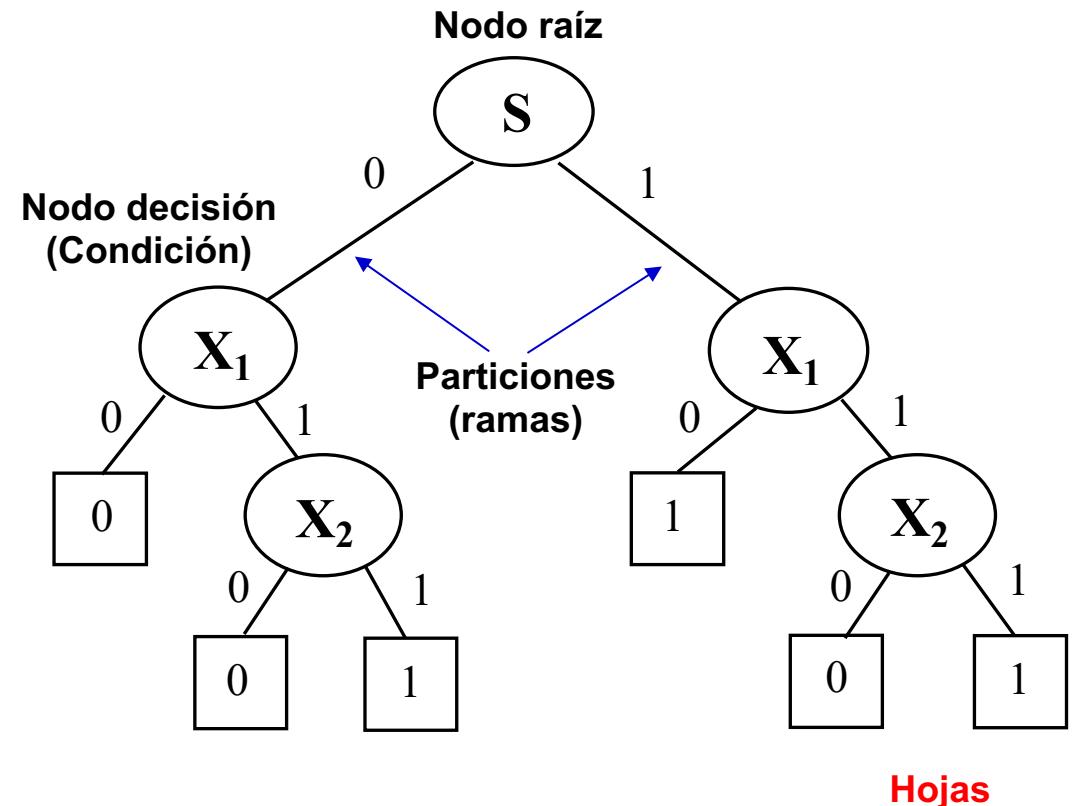
Profundidad. La profundidad indica los niveles que tiene el árbol de decisión.



Árboles de decisión

Estructura

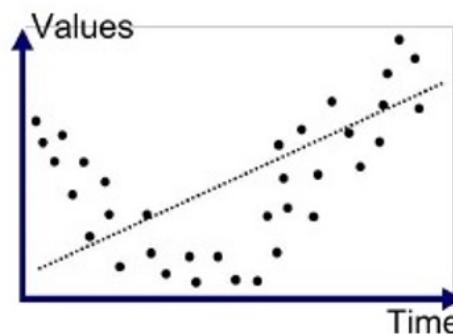
- Para la evaluación, se empieza en el nodo raíz y se avanza en el árbol siguiendo el nodo correspondiente que cumple la condición (**decisión**).
- Este proceso continúa hasta que se alcanza un **nodo hoja**, que contiene la predicción o el resultado del árbol de decisión.



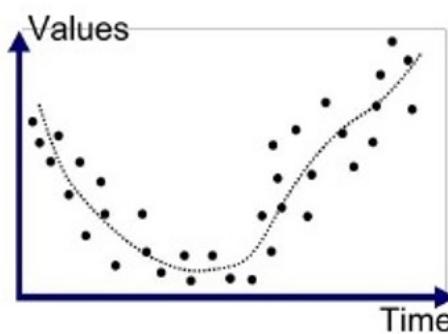
Árboles de decisión

Sobreajuste (Overfitting)

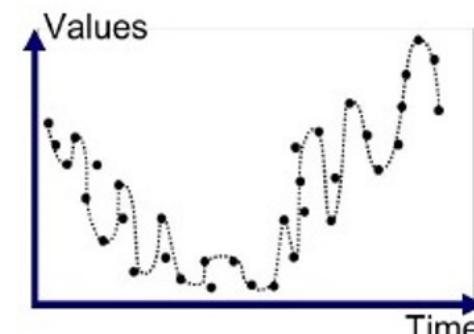
- Un problema común con los árboles de decisión es el **sobreajuste**. Esto es sinónimo de **sobreaprendizaje**, que ocurre cuando se genera un árbol de decisión que cubre todos los posibles casos sobre los datos.
- Por ejemplo, si los datos son sobre personas con diabetes, un árbol con overfitting generara nodos que cubran a cada persona que aparezca en los datos.
- A esto se considera que el árbol aprendió demasiado sobre las personas que aparecen en los datos.



Underfitted



Good Fit/Robust

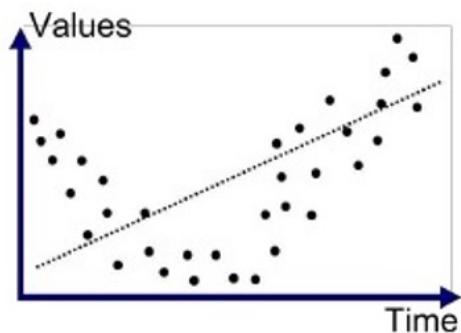


Overfitted

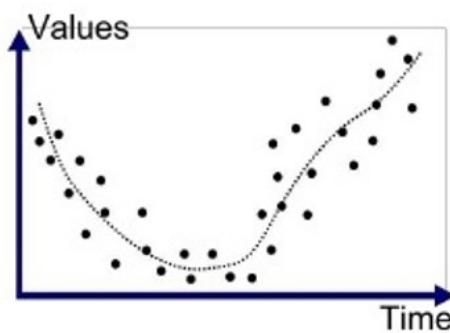
Árboles de decisión

Sobreajuste (Overfitting)

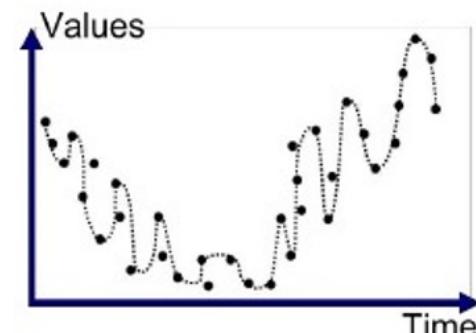
- El **sobreajuste** es uno de los desafíos más importantes en el proceso de modelación de árboles de decisión. Si no se definen límites, el árbol tendrá un 100% de precisión en el conjunto de datos de entrenamiento. En el peor de los casos tendrá una hoja por cada observación (elemento).
- Hay dos formas de evitar el sobreajuste: a) Definir restricciones sobre el tamaño del árbol, y b) Podar el árbol.



Underfitted



Good Fit/Robust



Overfitted

Árboles de decisión

Ventajas

- Se pueden utilizar para predecir valores continuos (regresión) y valores discretos (clasificación).
- Se pueden utilizar para clasificar datos separables de forma no lineal.
- Las reglas extraídas permiten hacer predicciones.
- Pueden ser utilizados con un amplio número de variables y gran cantidad de datos.
- Pueden trabajar con datos nulos (son robustos al ruido).
- Las hojas **no significativas** se podan.

Desventajas

- Si el **criterio de división** es deficiente, entonces el árbol no podrá ser generalizado.
- El principal problema es el **sobreajuste** que se puede tener.
- Este sobreajuste se refleja en tener un árbol demasiado profundo.
- Para evitar este problema se plantean **algoritmos de poda** para eliminar ramas profundas y no significativas.

Árboles de decisión

Parámetros del árbol de decisión

- **max_depth**. Indica la máxima profundidad a la cual puede llegar el árbol. Esto ayuda a combatir el **overfitting**, pero también puede provocar **underfitting**.
- **min_samples_leaf**. Indica la cantidad mínima de datos que debe tener un nodo hoja.
- **min_samples_split**. Indica la cantidad mínima de datos para que un nodo de decisión se pueda dividir. Si la cantidad no es suficiente este nodo se convierte en un nodo hoja.
- **criterion**. Indica la función que se utilizará para dividir los datos. Puede ser (ganancia de información) gini y entropy (Clasificación). Cuando el árbol es de regresión se usan funciones como el error cuadrado medio (MSE).

Árboles de decisión

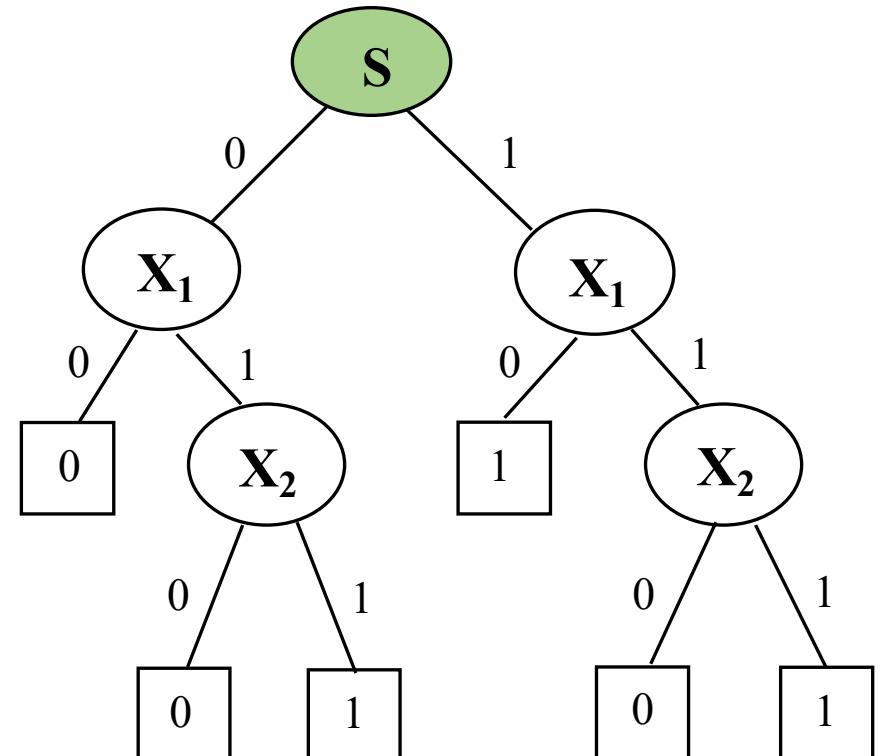
Algoritmo general

Entrada: Conjunto de elementos (E)

Salida: Un árbol de decisión T

Algoritmo divide y vencerás

1. Se crea un nodo raíz $S ::=$ todos los elementos (E).
2. Si todos los elementos de S son de la misma clase, el subárbol se cierra. Solución encontrada.
3. Sino, se elige una condición de partición para dividir E , siguiendo un criterio de partición (**split criterion**) E_1, E_2, \dots, E_n
4. El árbol queda subdivido en subárboles (los que cumplen la condición y los que no)
5. Se repite el paso 3 para cada uno de los subárboles.

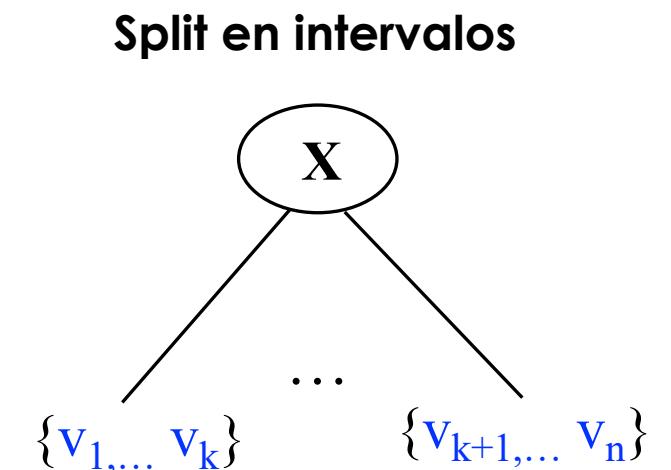
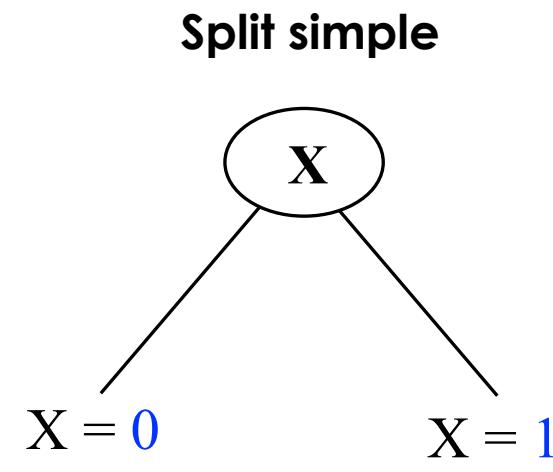


Criterio de decisión

Árboles de decisión

Split criterion. Busca hacer una división lógica de la variable.

- Se debe encontrar el mejor **split** en cada expansión del árbol.
- El número de **splits** depende del tipo de división que se considere.

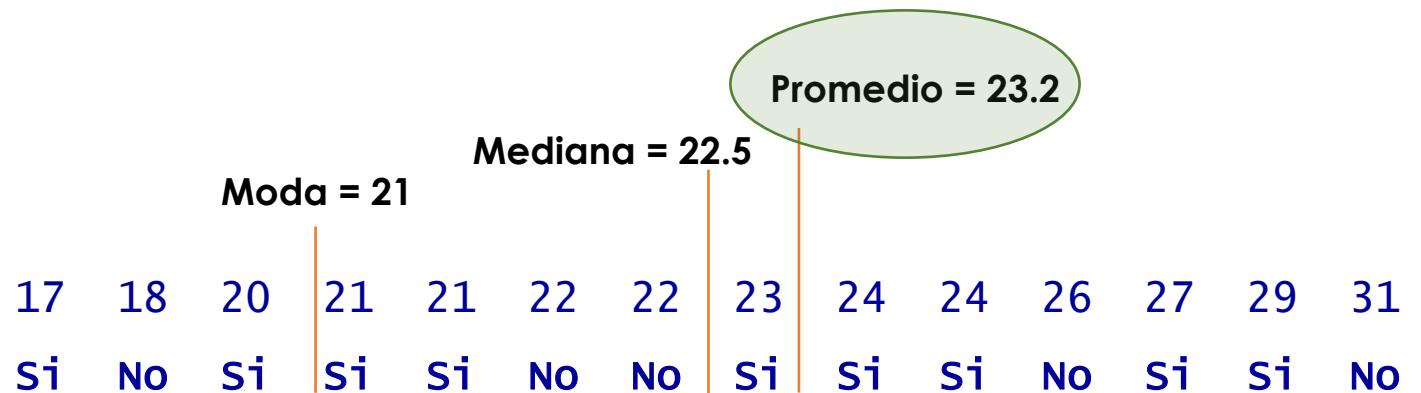
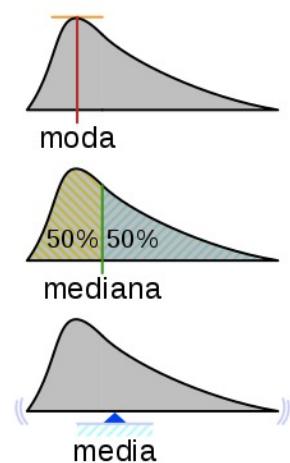


- Decisión binaria: Falso o Verdadero; 0 o 1; Positivo o Negativo; y otros.
- Categorías: Alto, Medio o Bajo; Bueno, Regular, Malo; y otras.
- Rangos y subrangos: [0, 15], [16, 30], [31, 45], ...

Árboles de decisión

La división de los datos (**Split**), numéricos o nominales, debe ser bajo un criterio específico. Por ejemplo, para una variable numérica ‘Temperatura’:

N.	Temperatura (°C)	Actividad
1	17	Si
2	18	No
3	20	Si
4	21	Si
5	21	Si
6	22	No
7	22	No
8	23	Si
9	24	Si
10	24	Si
11	26	No
12	27	Si
13	29	Si
14	31	No



- Temperatura < 23.2 (Si: 5, No: 3)
- Temperatura ≥ 23.2 (Si: 4, No: 2)
- $\text{Info}[(5,3), (4,2)] = 8/14 \text{ Info}[5,3] + 6/14 \text{ Info}[4,2]$
 $= 0.57 + 0.43$
 $= 1$

Ganancia de información
(división de los elementos)

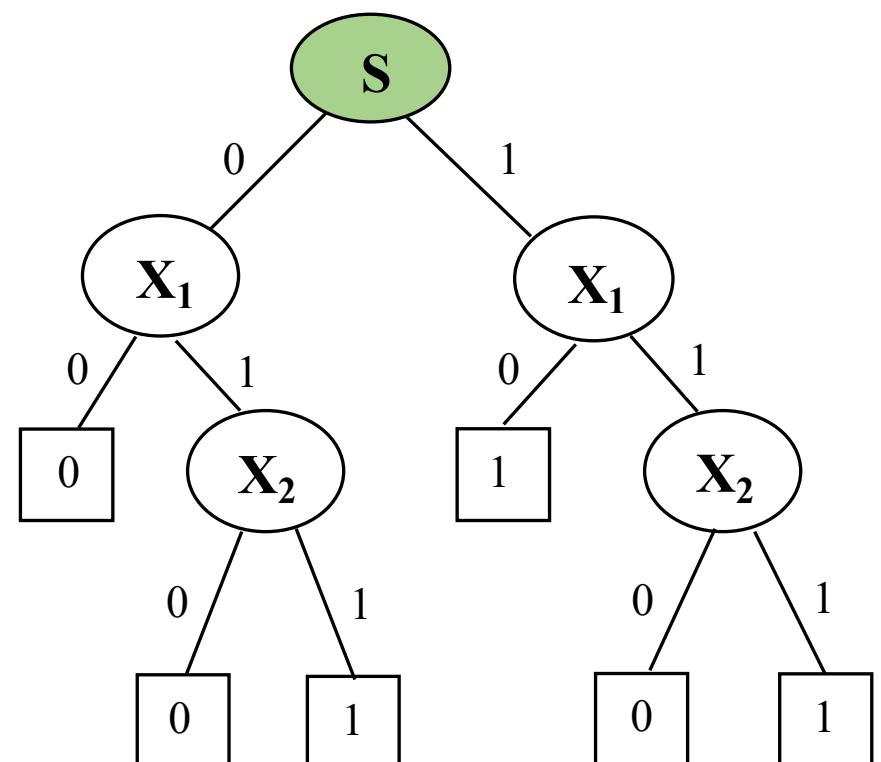
Creación del árbol Ganancia de Información

Ganancia de Información

En el **nodo raíz del árbol**, y en los otros niveles, se debe establecer la variable que mejor clasifica o pronostica los datos. **Entonces ¿Cuál es esa variable?**

La respuesta está en la **ganancia de información**.

- Para definir la **ganancia de información**, primero se debe analizar la **entropía**.
- La **entropía** es una medida de incertidumbre o de desorden en los datos. La cual se utiliza para decidir qué variable debe seleccionarse como nodo de división.



Ganancia de Información

Funcionamiento

- Se utiliza el concepto de **entropía**, la cual es una medida de incertidumbre (información).

$$\text{Entropía}(S) = I(S) = Inf(S) = \sum_{i=1}^n -p_i \log_2 p_i$$

S : es una colección de elementos (objetos).

p_i : es la probabilidad de posibles valores.

$+p$ y $-p$: son la proporción de elementos positivos y negativos en S .

$$GanInf(S, A) = Entropía(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} Entropía(S_v)$$

S : es una colección de elementos.

A : son las variables (atributos).

S_v : es un subconjunto de elementos.

$V(A)$: es el conjunto de valores que A (un atributo) puede tomar.

Ganancia de Información

Funcionamiento

1. Se calcula la entropía para todas las clases y atributos.
2. Se selecciona el mejor atributo basado en la **ganancia de información** de cada variable.
3. Se itera hasta que todos los elementos sean clasificados.

Ejemplo:

Alumno	Asistencia	Participación	Aprovechamiento	Calificación
1	No asiste	Media	Excelente	Exento
2	Asiste	Alta	Bueno	Exento
3	No asiste	Media	Bueno	Final
4	No asiste	Baja	Bueno	Final
5	Asiste	Alta	Regular	Final
6	Asiste	Baja	Deficiente	Extraordinario
7	No asiste	Media	Regular	Extraordinario

Ganancia de Información

1. Se calcula la entropía para todas las clases y atributos

$$\text{Clase (Calificación)} \begin{cases} \text{Exento} & = 2/7 \\ \text{Final} & = 3/7 \\ \text{Extraordinario} & = 2/7 \end{cases}$$

$$\textbf{Entropía}(S) = I(S) = \sum_{i=1}^n -p_i \log_2 p_i$$

$$I(C) = I(2/7, 3/7, 2/7) = -2/7 \log_2 2/7 - 3/7 \log_2 3/7 - 2/7 \log_2 2/7$$

$$I(C) = -0.286 \frac{\ln(2/7)}{\ln(2)} - 0.429 \frac{\ln(3/7)}{\ln(2)} - 0.286 \frac{\ln(2/7)}{\ln(2)} = -0.286 \frac{-1.253}{0.693} - 0.429 \frac{-0.847}{0.693} - 0.286 \frac{-1.253}{0.693}$$

$$I(C) = -0.286(-1.808) - 0.429(-1.222) - 0.286(-1.808) = 0.517 + 0.524 + 0.517 = 1.558$$

I(C) = 1.56 Información de la clase

Alumno	Asistencia	Participación	Aprovechamiento	Calificación
1	No asiste	Media	Excelente	Exento
2	Asiste	Alta	Bueno	Exento
3	No asiste	Media	Bueno	Final
4	No asiste	Baja	Bueno	Final
5	Asiste	Alta	Regular	Final
6	Asiste	Baja	Deficiente	Extraordinario
7	No asiste	Media	Regular	Extraordinario

Ganancia de Información

1. Se calcula la entropía para todas las clases y atributos

Asistencia = {Asiste, No asiste}

Asistencia = Asiste (3)

$$\text{Asiste} \begin{cases} \text{Exento} &= 1/3 \\ \text{Final} &= 1/3 \\ \text{Extraordinario} &= 1/3 \end{cases}$$

$$I(\text{Asistencia} = \text{Asiste}) = I(1/3, 1/3, 1/3) = -1/3\log_2 1/3 - 1/3\log_2 1/3 - 1/3\log_2 1/3 = \mathbf{1.58}$$

Asistencia = No asiste (4)

$$\text{No asiste} \begin{cases} \text{Exento} &= 1/4 \\ \text{Final} &= 2/4 \\ \text{Extraordinario} &= 1/4 \end{cases}$$

$$I(\text{Asistencia} = \text{No asiste}) = I(1/4, 2/4, 1/4) = -1/4\log_2 1/4 - 2/4\log_2 2/4 - 1/4\log_2 1/4 = \mathbf{1.5}$$

Alumno	Asistencia	Participación	Aprovechamiento	Calificación
1	No asiste	Media	Excelente	Exento
2	Asiste	Alta	Bueno	Exento
3	No asiste	Media	Bueno	Final
4	No asiste	Baja	Bueno	Final
5	Asiste	Alta	Regular	Final
6	Asiste	Baja	Deficiente	Extraordinario
7	No asiste	Media	Regular	Extraordinario

Ganancia de Información

1. Se calcula la entropía para todas las clases y atributos

Participación = {Alta, Media, Baja}

Participación = Alta (2)

$$\text{Alta} \begin{cases} \text{Exento} &= 1/2 \\ \text{Final} &= 1/2 \\ \text{Extraordinario} &= 0/2 \end{cases}$$

$$I(\text{Participación} = \text{Alta}) = I(1/2, 1/2, 0/2) \\ = -1/2\log_2 1/2 - 1/2\log_2 1/2 - 0/2\log_2 0/2 = \mathbf{1.0}$$

Participación = Media (3)

$$\text{Media} \begin{cases} \text{Exento} &= 1/3 \\ \text{Final} &= 1/3 \\ \text{Extraordinario} &= 1/3 \end{cases}$$

$$I(\text{Participación} = \text{Media}) = I(1/3, 1/3, 1/3) = -1/3\log_2 1/3 - 1/3\log_2 1/3 - 1/3\log_2 1/3 = \mathbf{1.58}$$

Participación = Baja (2)

$$\text{Baja} \begin{cases} \text{Exento} &= 0/2 \\ \text{Final} &= 1/2 \\ \text{Extraordinario} &= 1/2 \end{cases}$$

$$I(\text{Participación} = \text{Baja}) = I(0/2, 1/2, 1/2) = -0/2\log_2 0/2 - 1/2\log_2 1/2 - 1/2\log_2 1/2 = \mathbf{1.0}$$

Alumno	Asistencia	Participación	Aprovechamiento	Calificación
1	No asiste	Media	Excelente	Exento
2	Asiste	Alta	Bueno	Exento
3	No asiste	Media	Bueno	Final
4	No asiste	Baja	Bueno	Final
5	Asiste	Alta	Regular	Final
6	Asiste	Baja	Deficiente	Extraordinario
7	No asiste	Media	Regular	Extraordinario

Ganancia de Información

1. Se calcula la entropía para todas las clases y atributos

Aprovechamiento = {Excelente, Bueno, Regular, Deficiente}

Aprovechamiento = Excelente (1)

$$\text{Excelente} \begin{cases} \text{Exento} &= 1/1 \\ \text{Final} &= 0/1 \\ \text{Extraordinario} &= 0/1 \end{cases}$$

$$I(\text{Aprovechamiento} = \text{Excelente}) = I(1/1, 0/1, 0/1) = -1/1\log_21/1 - 0/1\log_20/1 - 0/1\log_20/1 = \mathbf{0}$$

Aprovechamiento = Bueno (3)

$$\text{Bueno} \begin{cases} \text{Exento} &= 1/3 \\ \text{Final} &= 2/3 \\ \text{Extraordinario} &= 0/3 \end{cases}$$

$$I(\text{Aprovechamiento} = \text{Bueno}) = I(1/3, 2/3, 0/3) = -1/3\log_21/3 - 2/3\log_22/3 - 0/3\log_20/3 = \mathbf{0.92}$$

Aprovechamiento = Regular (2)

$$\text{Regular} \begin{cases} \text{Exento} &= 0/2 \\ \text{Final} &= 1/2 \\ \text{Extraordinario} &= 1/2 \end{cases}$$

$$I(\text{Aprovechamiento} = \text{Regular}) = I(0/2, 1/2, 1/2) = -0/2\log_20/2 - 1/2\log_21/2 - 1/2\log_21/2 = \mathbf{1.0}$$

Aprovechamiento = Deficiente (1)

$$\text{Deficiente} \begin{cases} \text{Exento} &= 0/1 \\ \text{Final} &= 0/1 \\ \text{Extraordinario} &= 1/1 \end{cases}$$

$$I(\text{Aprovechamiento} = \text{Deficiente}) = I(0/1, 0/1, 1/1) = -0/1\log_20/1 - 0/1\log_20/1 - 1/1\log_21/1 = \mathbf{0}$$

Alumno	Asistencia	Participación	Aprovechamiento	Calificación
1	No asiste	Media	Excelente	Exento
2	Asiste	Alta	Bueno	Exento
3	No asiste	Media	Bueno	Final
4	No asiste	Baja	Bueno	Final
5	Asiste	Alta	Regular	Final
6	Asiste	Baja	Deficiente	Extraordinario
7	No asiste	Media	Regular	Extraordinario

Ganancia de Información

2. Se selecciona el mejor atributo basado en la ganancia de información de cada variable.

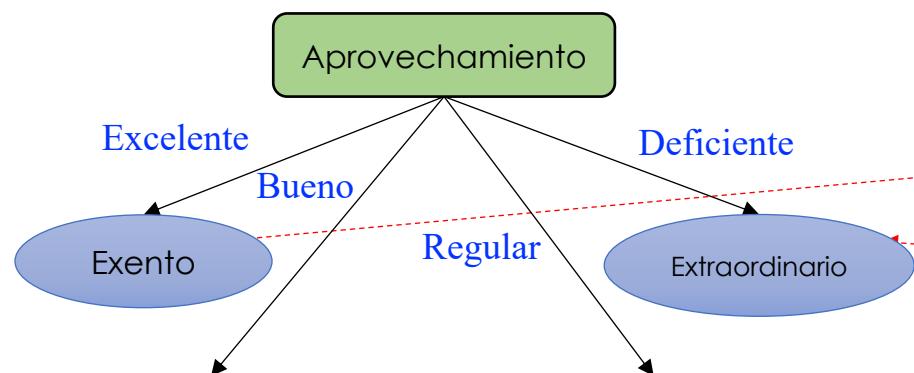
$$GanInf(S, A) = Entropia(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} Entropia(S_v)$$

Variable (Atributo)	Ganancia de información
Asistencia	$1.56 - [3/7(1.58) + 4/7(1.5)] = 0.025$
Participación	$1.56 - [2/7(1.0) + 3/7(1.58) + 2/7(1.0)] = 0.311$
Aprovechamiento	$1.56 - [1/7(0) + 3/7(0.92) + 2/7(1.0) + 1/7(0)] = 0.880$

Ganancia de Información

2. Se selecciona el mejor atributo basado en la ganancia de información de cada variable.

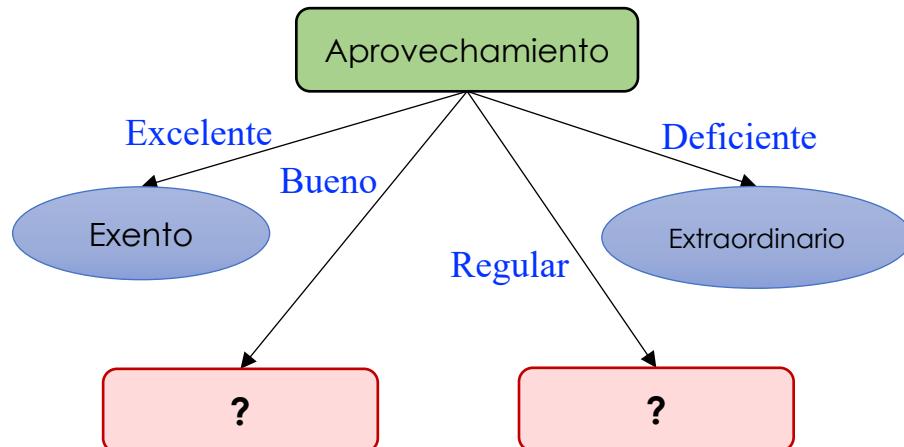
Variable (Atributo)	Ganancia de información
Asistencia	$1.56 - [3/7(1.58) + 4/7(1.5)] = 0.025$
Participación	$1.56 - [2/7(1.0) + 3/7(1.58) + 2/7(1.0)] = 0.311$
Aprovechamiento	$1.56 - [1/7(0) + 3/7(0.92) + 2/7(1.0) + 1/7(0)] = 0.880$



Alumno	Asistencia	Participación	Aprovechamiento	Calificación
1	No asiste	Media	Excelente	Exento
2	Asiste	Alta	Bueno	Exento
3	No asiste	Media	Bueno	Final
4	No asiste	Baja	Bueno	Final
5	Asiste	Alta	Regular	Final
6	Asiste	Baja	Deficiente	Extraordinario
7	No asiste	Media	Regular	Extraordinario

Ganancia de Información

3. Se itera hasta que todos los elementos sean clasificados



Alumno	Asistencia	Participación	Aprovechamiento	Calificación
1	No asiste	Media	Excelente	Exento
2	Asiste	Alta	Bueno	Exento
3	No asiste	Media	Bueno	Final
4	No asiste	Baja	Bueno	Final
5	Asiste	Alta	Regular	Final
6	Asiste	Baja	Deficiente	Extraordinario
7	No asiste	Media	Regular	Extraordinario

Información de la clase: $I(C)$

$$\text{Bueno} \begin{cases} \text{Exento} = 1/3 \\ \text{Final} = 2/3 \end{cases}$$

$$I(C) = I(1/3, 2/3) = -1/3\log_2 1/3 - 2/3\log_2 2/3 = 0.92$$

Ganancia de Información

3. Se itera hasta que todos los elementos sean clasificados

Asistencia = {Asiste, No asiste}

Asistencia = Asiste (1)

$$\text{Asiste} \begin{cases} \text{Exento} = 1/1 \\ \text{Final} = 0/1 \end{cases}$$

$$I(\text{Asistencia} = \text{Asiste}) = I(1/1, 0/1) = -1/1\log_2 1/1 - 0/1\log_2 0/1 = 0$$

Asistencia = No asiste (2)

$$\text{No asiste} \begin{cases} \text{Exento} = 0/2 \\ \text{Final} = 2/2 \end{cases}$$

$$I(\text{Asistencia} = \text{No asiste}) = I(0/2, 2/2) = -0/2\log_2 0/2 - 2/2\log_2 2/2 = 0$$

Alumno	Asistencia	Participación	Aprovechamiento	Calificación
1	No asiste	Media	Excelente	Exento
2	Asiste	Alta	Bueno	Exento
3	No asiste	Media	Bueno	Final
4	No asiste	Baja	Bueno	Final
5	Asiste	Alta	Regular	Final
6	Asiste	Baja	Deficiente	Extraordinario
7	No asiste	Media	Regular	Extraordinario

Ganancia de Información

3. Se itera hasta que todos los elementos sean clasificados

Participación = {Alta, Media, Baja}

Participación = Alta (1)

$$\text{Alta} \begin{cases} \text{Exento} = 1/1 \\ \text{Final} = 0/1 \end{cases}$$

$$I(\text{Participación} = \text{Alta}) = I(1/1, 0/1) = -1/1\log_2 1/1 - 0/1\log_2 0/1 = 0$$

Participación = Media (1)

$$\text{Media} \begin{cases} \text{Exento} = 0/1 \\ \text{Final} = 1/1 \end{cases}$$

$$I(\text{Participación} = \text{Media}) = I(0/1, 1/1) = -0/1\log_2 0/1 - 1/1\log_2 1/1 = 0$$

Participación = Baja (1)

$$\text{Baja} \begin{cases} \text{Exento} = 0/1 \\ \text{Final} = 1/1 \end{cases}$$

$$I(\text{Participación} = \text{Baja}) = I(0/1, 1/1) = -0/1\log_2 0/1 - 1/1\log_2 1/1 = 0$$

Alumno	Asistencia	Participación	Aprovechamiento	Calificación
1	No asiste	Media	Excelente	Exento
2	Asiste	Alta	Bueno	Exento
3	No asiste	Media	Bueno	Final
4	No asiste	Baja	Bueno	Final
5	Asiste	Alta	Regular	Final
6	Asiste	Baja	Deficiente	Extraordinario
7	No asiste	Media	Regular	Extraordinario

Ganancia de Información

3. Se itera hasta que todos los elementos sean clasificados

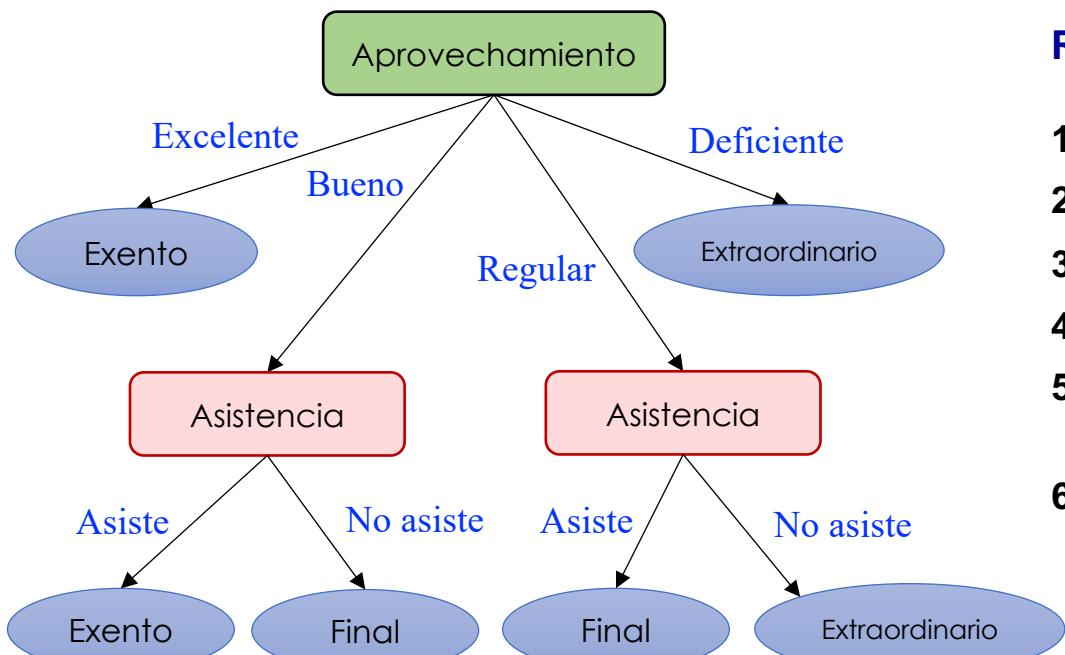
Se selecciona el mejor atributo basado en la ganancia de información de la variable.

$$GanInf(S, A) = Entropia(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} Entropia(S_v)$$

Variable (Atributo)	Ganancia de información
Asistencia	$0.92 - [1/3(0) + 2/3(0)] = 0.92$
Participación	$0.92 - [1/3(0) + 1/3(0) + 1/3(0)] = 0.92$

Ganancia de Información

3. Se itera hasta que todos los elementos sean clasificados



Reglas de decisión:

1. IF (Aprovechamiento = Excelente) THEN Calificación = Exento
2. IF (Aprovechamiento = Bueno) & (Asistencia = Asiste) THEN Calificación = Exento
3. IF (Aprovechamiento = Bueno) & (Asistencia = No asiste) THEN Calificación = Final
4. IF (Aprovechamiento = Regular) & (Asistencia = Asiste) THEN Calificación = Final
5. IF (Aprovechamiento = Regular) & (Asistencia = No asiste) THEN Calificación = Extraordinario
6. IF (Aprovechamiento = Deficiente) THEN Calificación = Extraordinario

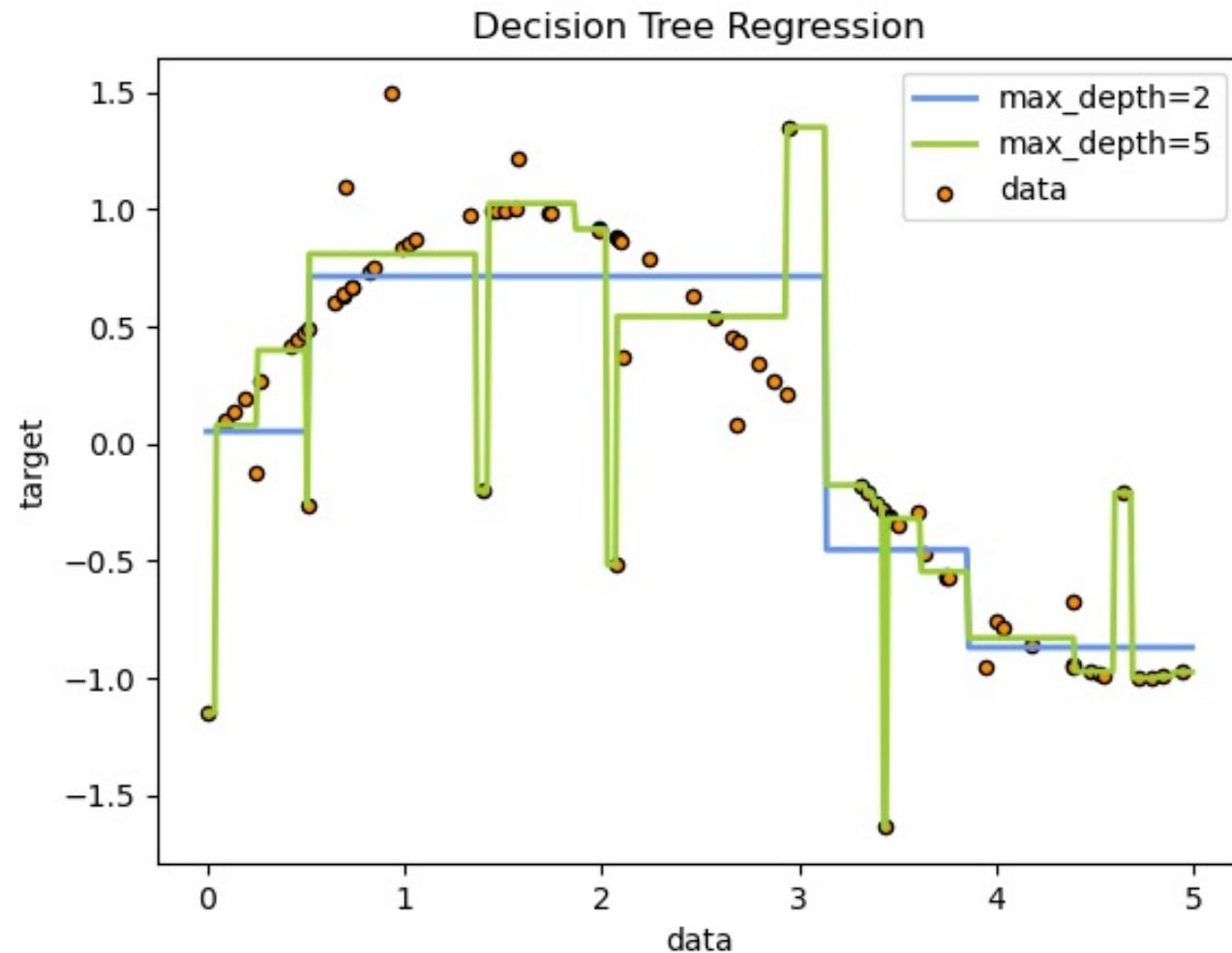
Algoritmos de árboles de decisión

Los tipos más notables de algoritmos de árboles de decisión son:

1. **Iterative Dichotomiser 3 (ID3, Quinlan, 1986)**. Este algoritmo utiliza la **ganancia de información** para decidir qué atributo se utilizará para clasificar los datos. Para cada nivel del árbol, la ganancia de información se calcula para los datos restantes de forma recursiva.
2. **C4.5 (Quinlan, 1993)**. Es el sucesor de ID3. Utiliza la **ganancia de información** o la relación de ganancia para decidir el atributo de clasificación. Es una mejora del algoritmo ID3, ya que puede manejar valores continuos y datos faltantes.
3. **Classification and Regression Tree (CART, Breiman et al., 1984)**. Es un algoritmo de aprendizaje dinámico que puede producir un árbol de regresión y un árbol de clasificación dependiendo de la variable dependiente.
4. **Chi-square automatic interaction detection (CHAID)**. Realiza divisiones de varios niveles al calcular árboles de clasificación.

Árboles de decisión Regresión

Árboles de decisión (Regresión)



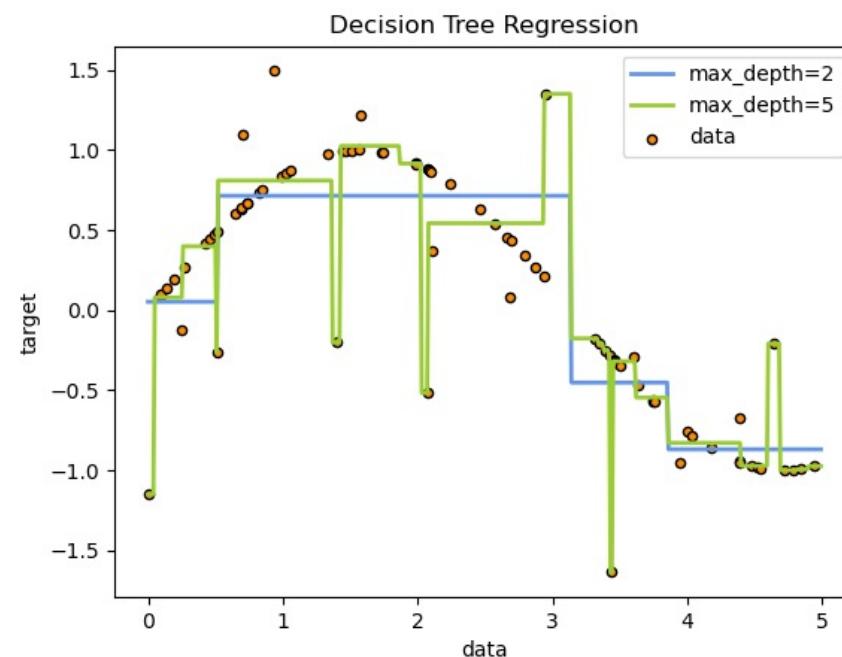
Árboles de decisión (Regresión)

- Los árboles de decisión también se pueden aplicar a problemas de regresión (pronóstico).
- Al igual que en la configuración de clasificación, el método de ajuste tomará como argumentos las matrices X (variables predictoras) e Y (variable a pronosticar).
- En este caso Y tiene valores continuos.
- Los árboles de decisión pueden aprender detalles finos de los datos de entrenamiento, y aprenden también del ruido.
- **En Python**, scikit-learn usa una versión optimizada del algoritmo CART.

Árboles de decisión (Regresión)

Criterio de regresión

- Si el objetivo es un valor continuo, entonces, para el nodo (m), los criterios comunes para determinar las ubicaciones para futuras divisiones son el **error cuadrático medio (MSE)**, y el **error absoluto medio (MAE)**.
- La desviación de **MSE** establece el valor pronosticado de los nodos terminales con respecto el valor medio aprendido (\bar{y}_m).



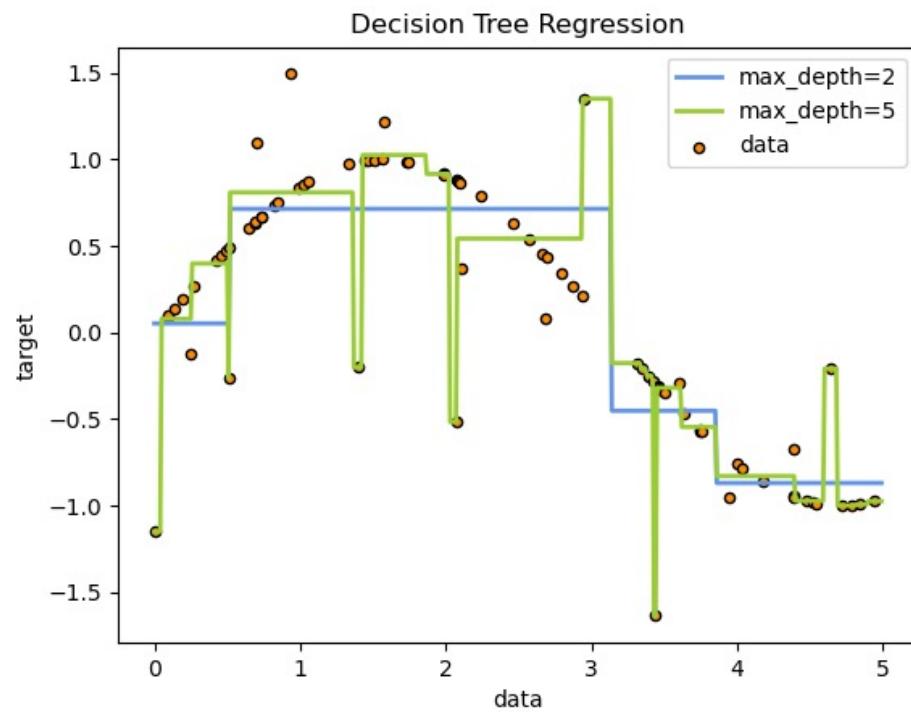
$$\bar{y}_m = \frac{1}{N_m} \sum_{y \in m}^N y$$

$$MSE = \frac{1}{N_m} \sum (y - \bar{y}_m)^2$$

Árboles de decisión (Regresión)

Criterio de regresión

- Mientras que el **MAE** establece el valor pronosticado de los nodos terminales con respecto a la mediana: $\text{median}(y)_m$



$$\text{median}(y)_m = \text{median}(y)$$

$$MAE = \frac{1}{N_m} \sum |y - \text{median}(y)_m|$$

Árboles de decisión (Regresión)

- Posterior de la estimación del **MSE** se obtiene la raíz del error cuadrático medio (**RMSE**), para dar una estimación en términos de la calidad del pronóstico.

$$RMSE = \sqrt{\frac{1}{N_m} \sum (y - \bar{y}_m)^2}$$

- Se puede obtener también el Score (coeficiente de determinación) para medir la efectividad del modelo de regresión.

$$\bar{R}^2 = 1 - \frac{n - 1}{n - k - 1} [1 - R^2]$$

Número de elementos Número de variables independientes Coeficiente de correlación (valor real y valores pronosticados)
 $\text{corr}(Y, \hat{Y})$

Árboles de decisión

Clasificación

Árboles de decisión (Clasificación)

Criterio de clasificación

- Se utiliza el concepto de **entropía**, la cual es una medida de incertidumbre (información).

$$\text{Entropía}(S) = I(S) = Inf(S) = \sum_{i=1}^n -p_i \log_2 p_i$$

S : es una colección de elementos (objetos).

p_i : es la probabilidad de posibles valores.

$+p$ y $-p$: son la proporción de elementos positivos y negativos en S .

$$GanInf(S, A) = Entropía(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} Entropía(S_v)$$

S : es una colección de elementos.

A : son las variables (atributos).

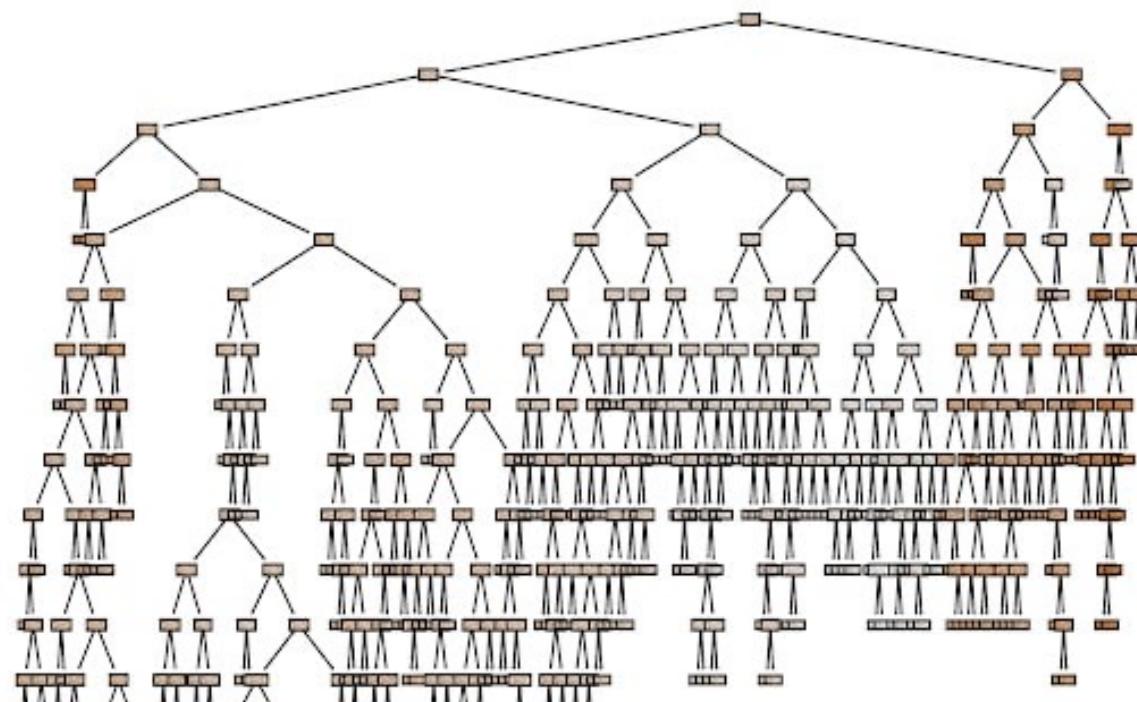
S_v : es un subconjunto de elementos.

$V(A)$: es el conjunto de valores que A (un atributo) puede tomar.

Árboles de decisión (Clasificación)

Criterio de clasificación

1. Se calcula la entropía para todas las clases y atributos.
2. Se selecciona el mejor atributo basado en la **ganancia de información** de cada variable.
3. Se itera hasta que todos los elementos sean clasificados.



Árboles de decisión (Clasificación)

Validación a través de una matriz de clasificación

- 1) Se evalúan todos los elementos y se determina si la **predicción (clase)** coincide con los **valores reales (Y)**.
- 2) Se cuentan todos los elementos y se muestran los totales obtenidos en la matriz.

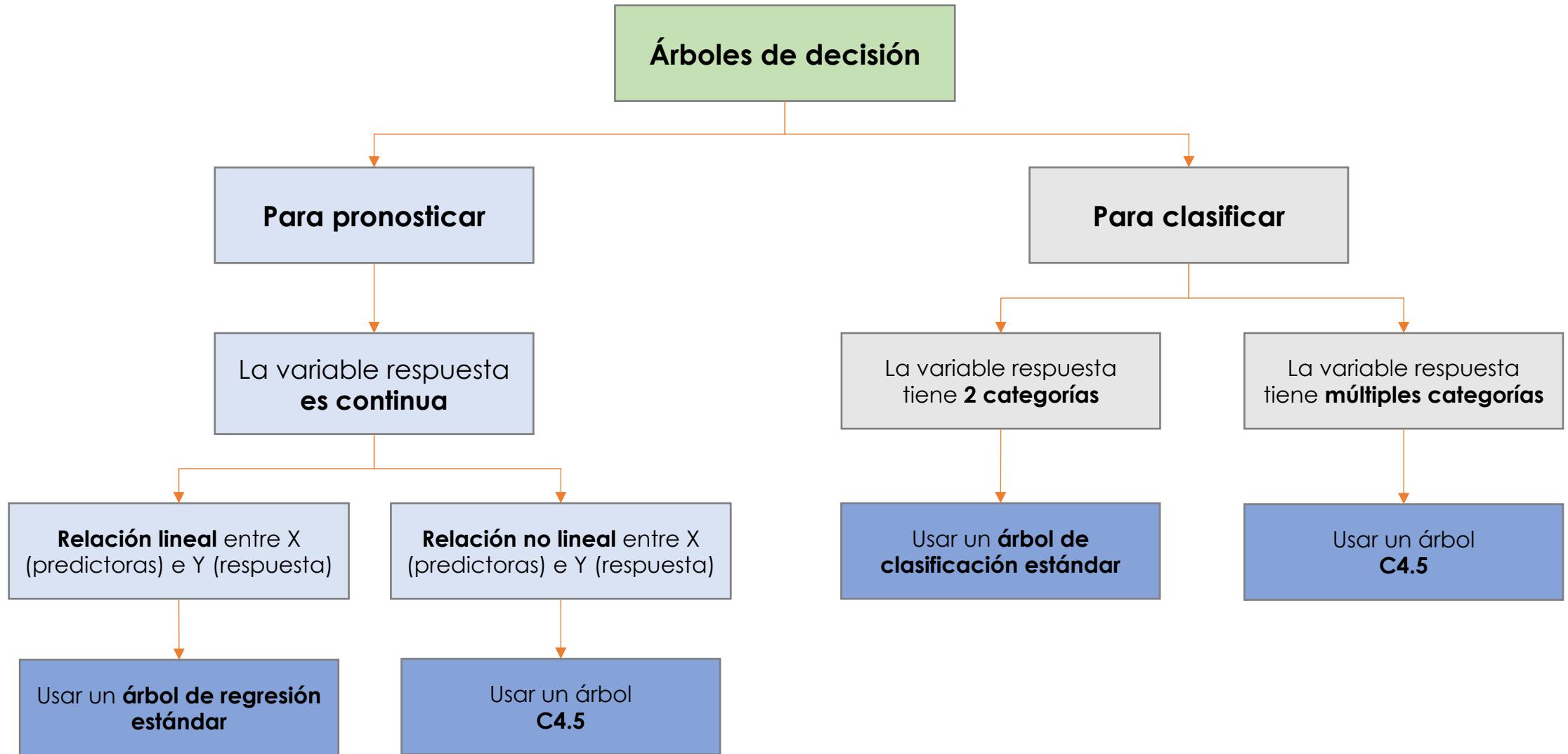
		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Árboles de decisión (Clasificación)

Mediciones

- 1) Exactitud (Accuracy)
- 2) Tasa de error (Misclassification Rate)
- 3) Precisión (Precision)
- 4) Sensibilidad (Recall, Sensitivity, True Positive Rate)
- 5) Especificidad (Especificity, True Negative Rate)

En resumen



Tarea 4

Importar datos en Google Colab o Jupyter a través de alguna plataforma de Git. Por ejemplo, Github.

Objetivo. Mostrar el procedimiento y la correcta ejecución de la importación de datos (archivos CSV) desde un repositorio Git dentro de un cuaderno en Google Colab o Jupyter.

Fecha de entrega: Martes 30 de noviembre de 2021

Hora: Antes de las 11:00 horas

Formato: Libre, subir a la carpeta compartida el reporte de la tarea en un archivo 'pdf'.

Se debe pasar el parámetro a `read_csv()` en pandas para obtener la matriz de datos.

```
url = 'copied_raw_github_link'  
df = pd.read_csv(url)
```

Salida:

	Unnamed: 0	V1	V2
0	1	2.072345	-3.241693
1	2	17.936710	15.784810
2	3	1.083576	7.319176
3	4	11.120670	14.406780
4	5	23.711550	2.557729
...
2995	2996	85.652800	-6.461061
2996	2997	82.770880	-2.373299
2997	2998	64.465320	-10.501360
2998	2999	90.722820	-12.255840
2999	3000	64.879760	-24.877310

3000 rows × 3 columns



Universidad Nacional Autónoma de México
Facultad de Ingeniería

Bosques Aleatorios

Guillermo Molero-Castillo

guillermo.molero@ingenieria.unam.edu

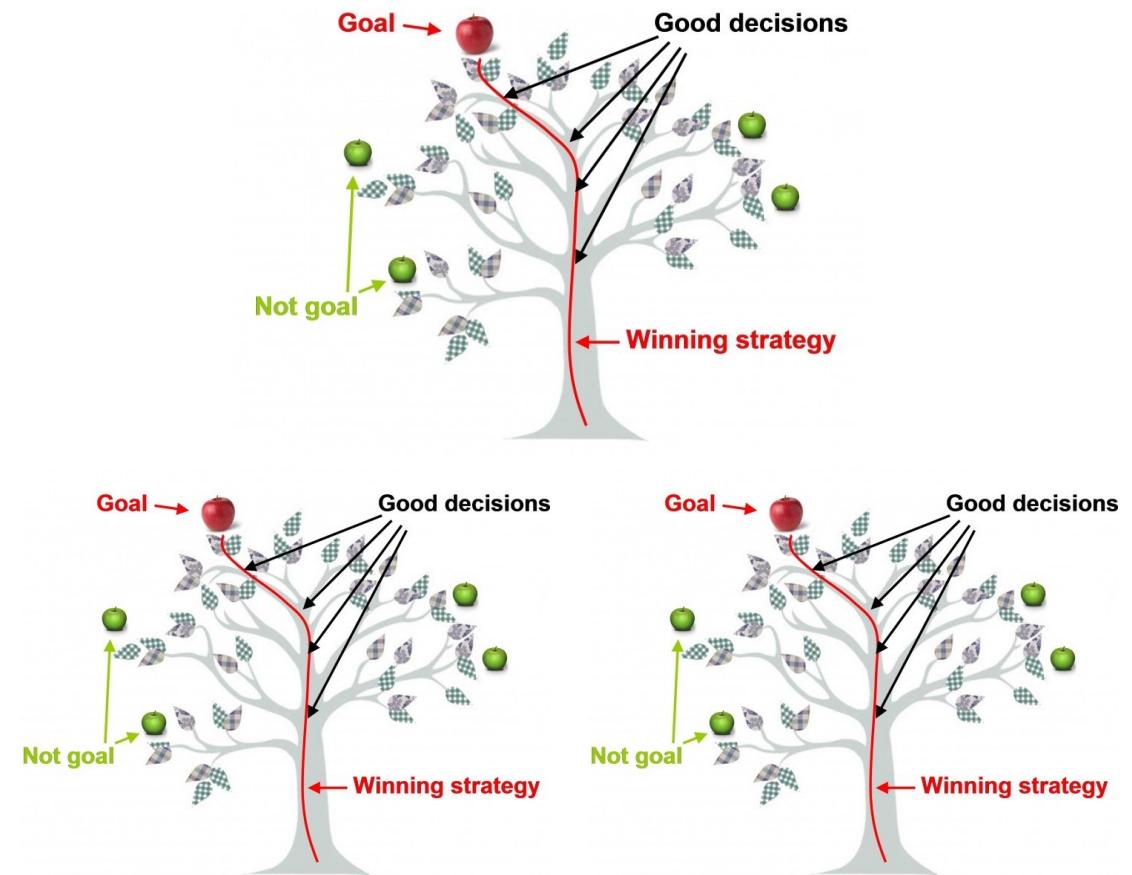
Noviembre, 2021

Contexto



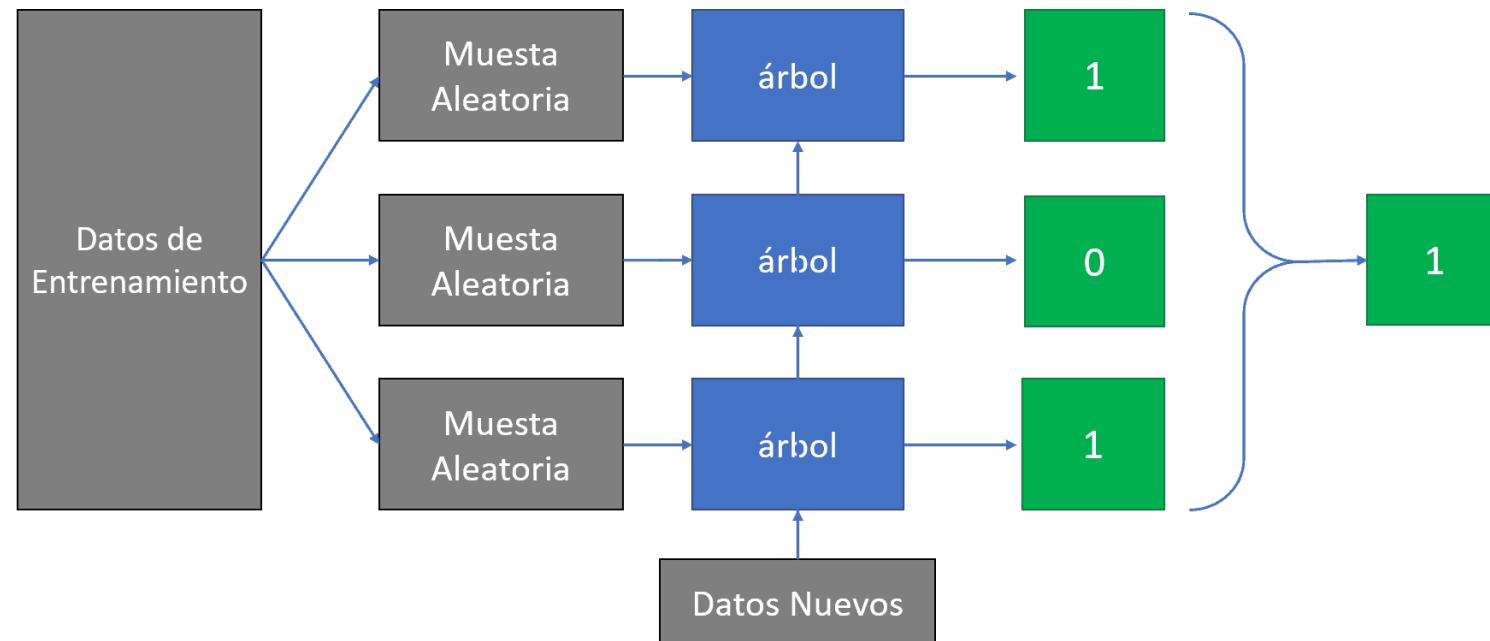
Bosques aleatorios

- En algunas ocasiones los árboles de decisión tienen la tendencia de sobreajuste (overfit). Esto significa que tienden a aprender muy bien de los datos de entrenamiento, pero su generalización pudiera ser no tan buena.
- Una forma de **mejorar la generalización** de los árboles de decisión es combinar varios árboles.
- A esto se conoce como **Bosque Aleatorio (Random Forest)**, el cual es un poderoso algoritmo de aprendizaje automático, ampliamente utilizado en la actualidad.
- Los bosques aleatorios tienen una capacidad de generalización alta.



Bosques aleatorios

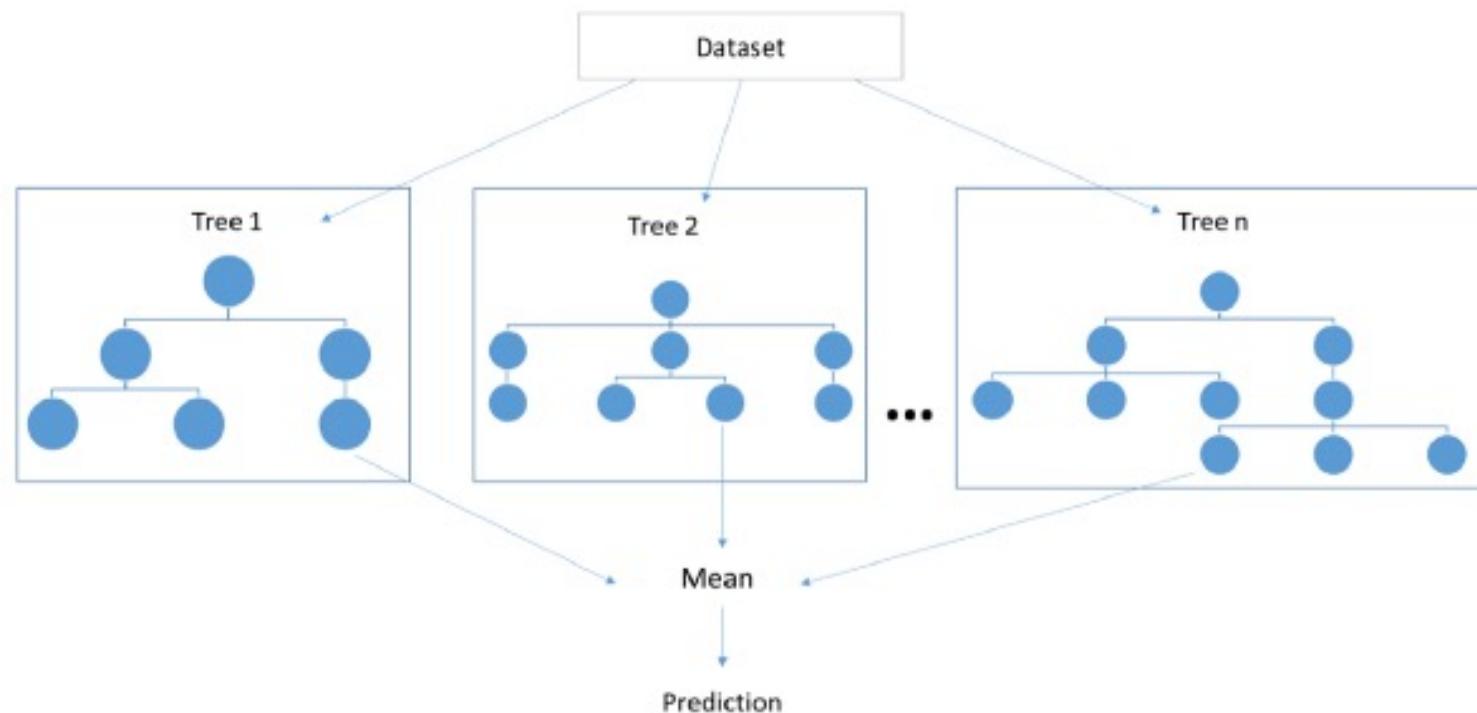
Objetivo. Construir un conjunto (ensamble) de árboles de decisión combinados. Al combinar lo que en realidad está pasando es que distintos árboles ven distintas porciones de los datos.



Ningún árbol ve todos los datos de entrenamiento, sino cada uno se entrena con distintas muestras para un mismo problema.

Bosques aleatorios

- Al combinar los resultados, los errores se compensan con otros y se tiene una predicción (pronóstico o clasificación) que generaliza mejor al problema.
- Por lo que, los bosques aleatorios son una variación moderna, que agrupan varios árboles de decisión para producir un modelo generalizado con el objetivo de reducir la tendencia al sobreajuste.

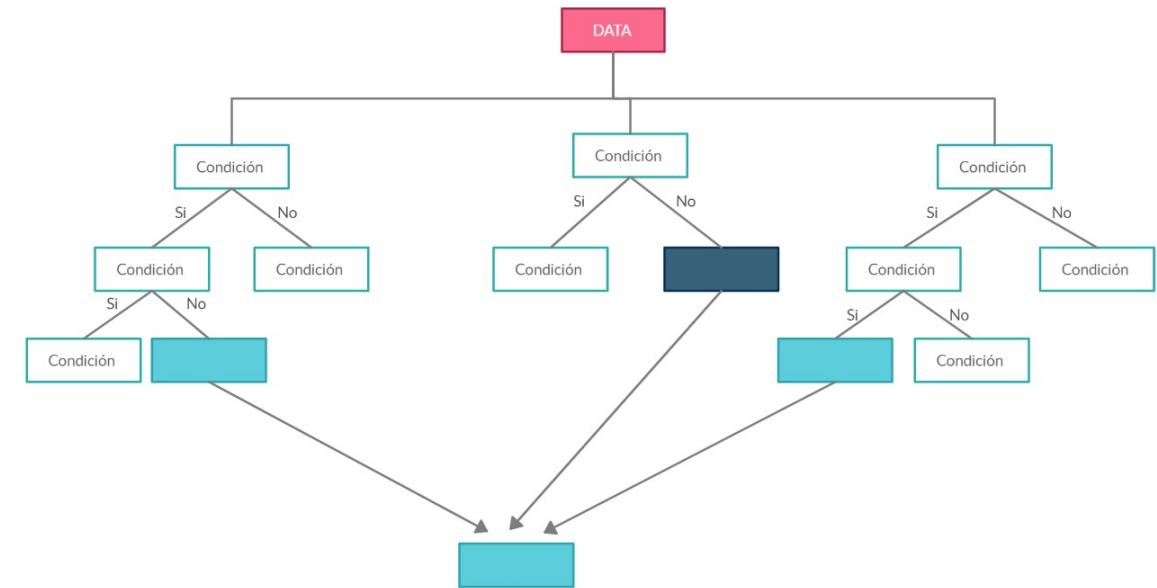


Se dice que cuantos más árboles se tiene, más robusto es el bosque.

Bosques aleatorios

El algoritmo de los bosques aleatorios tiene **también** analogía con el razonamiento humano. Por ejemplo, en términos sencillos:

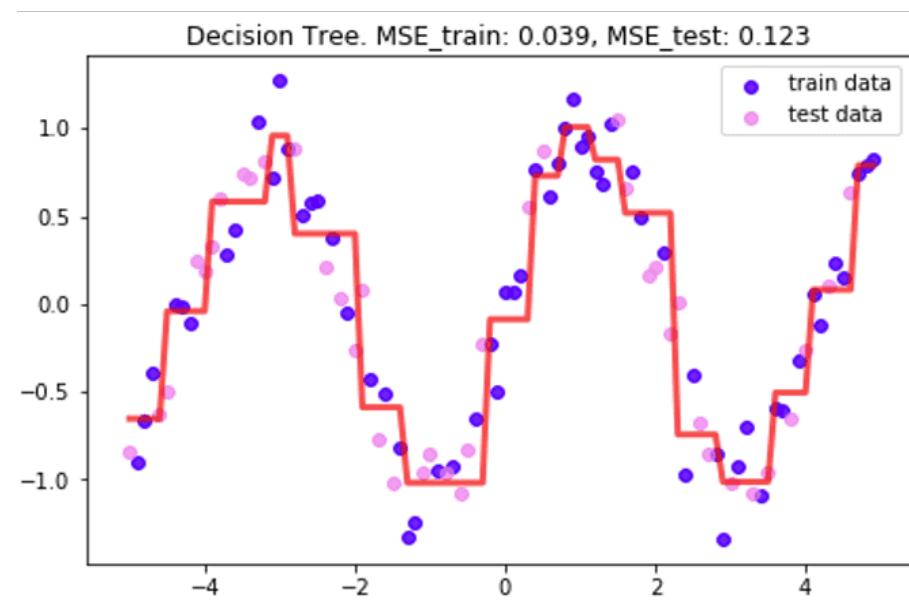
- Si queremos ir de viaje y nos gustaría conocer ciertos lugares, entonces:
 1. Se puede buscar y leer las reseñas en blogs y portales de viajes, o también se puede preguntar a los amigos.
 2. Si se pregunta a los amigos sobre su experiencia de viaje a esos lugares de interés, entonces se reciben recomendaciones.
 3. Con base en esas recomendaciones se hace una lista de los lugares de interés.
 4. Se somete a votación (selección del mejor lugar para el viaje) con base en el mayor número de coincidencias en las recomendaciones.



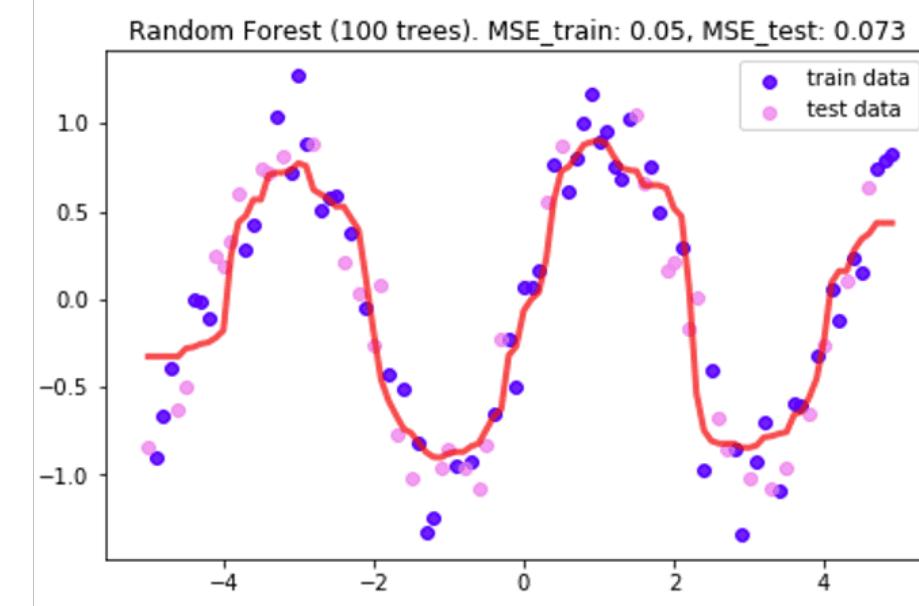
Bosques aleatorios

La diferencia visual entre un árbol de decisión y un bosque aleatorio es el ajuste (generalización – más suave –), por ejemplo, al resolver el mismo problema de pronóstico (regresión).

Modelo aprendido por el árbol de decisión



Modelo aprendido por el bosque aleatorio

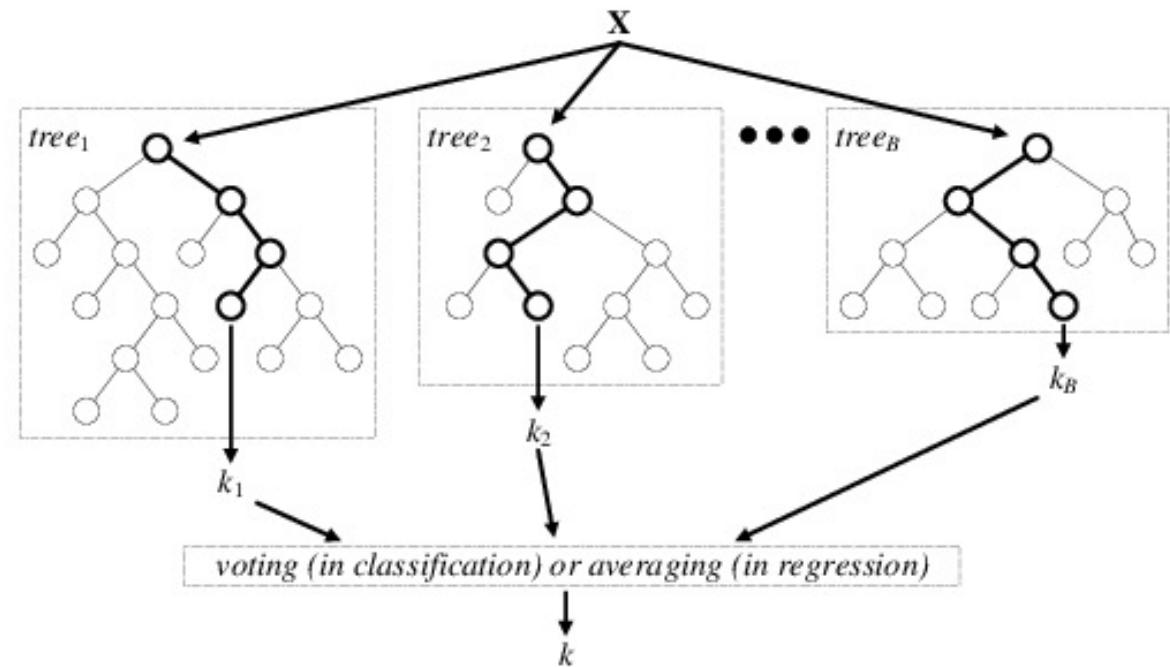


Bosques aleatorios

Procedimiento

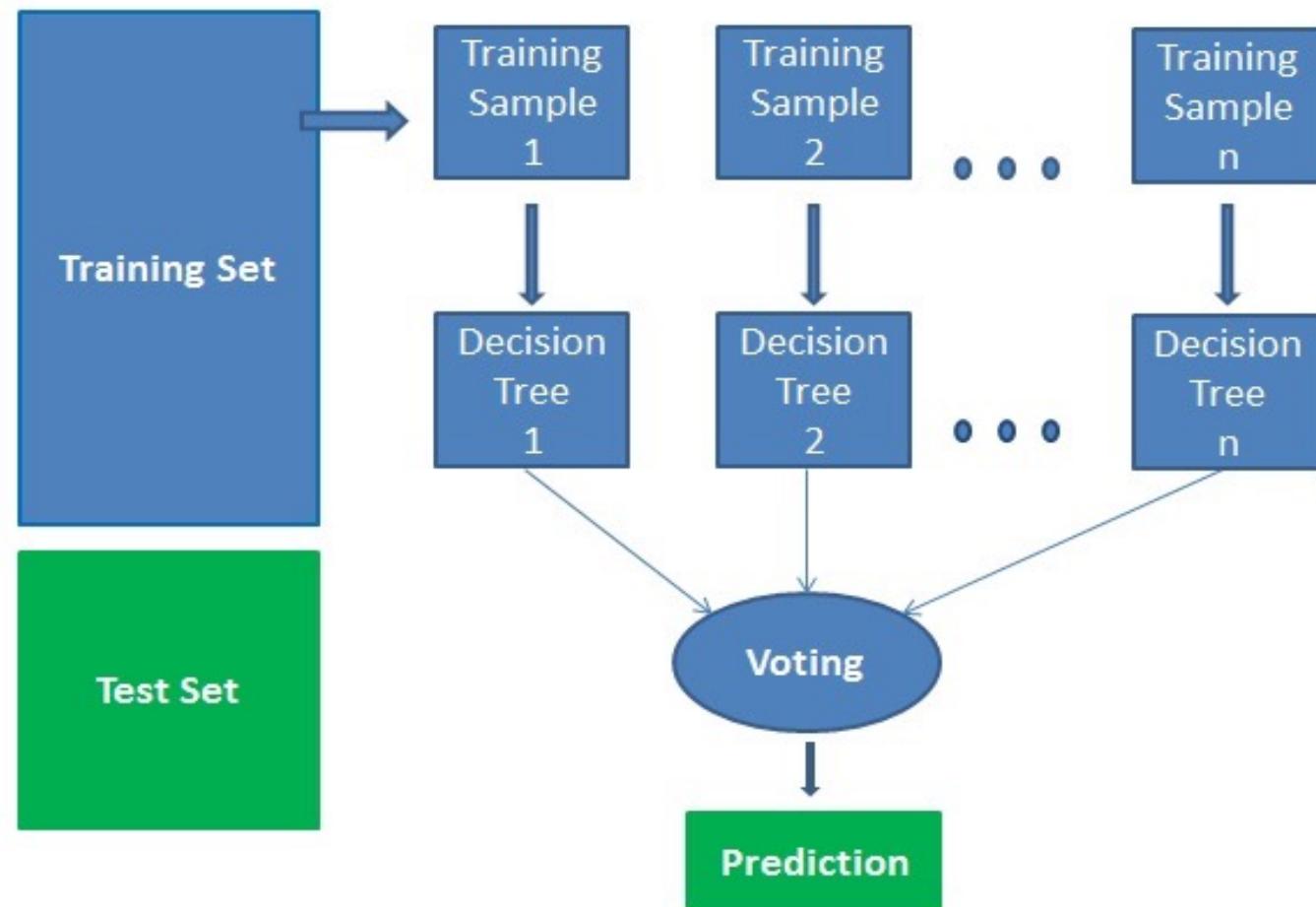
Los bosques aleatorios funcionan en cuatro pasos:

1. Se selecciona muestras aleatorias a partir del conjunto de datos.
2. Se construye un árbol de decisión para cada muestra y se obtiene un resultado.
3. Se realiza una votación (clasificación) o promedio (regresión) con base en los resultados.
4. Se selecciona el resultado con más votos (clasificación) o promedio final (regresión).



Bosques aleatorios

Procedimiento

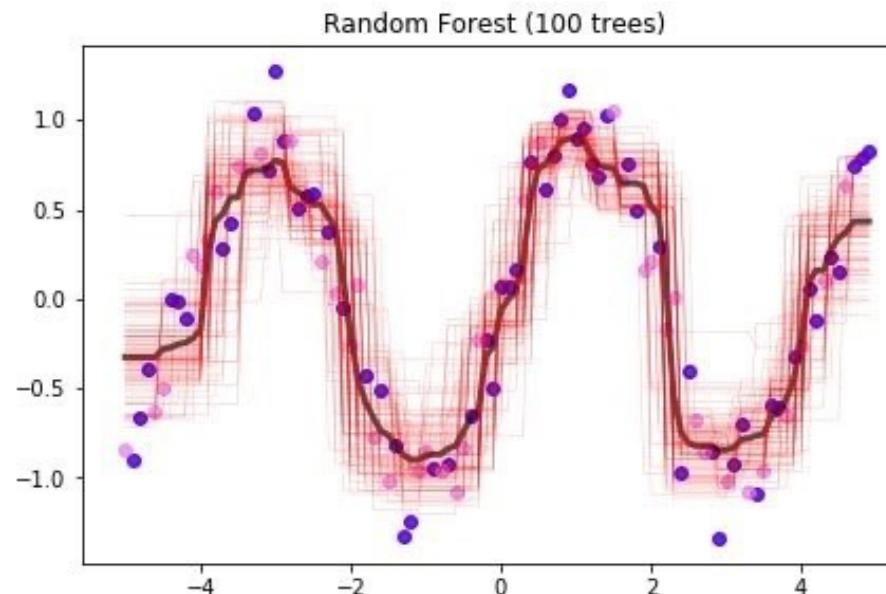


Bosques aleatorios

Combinación de predicciones

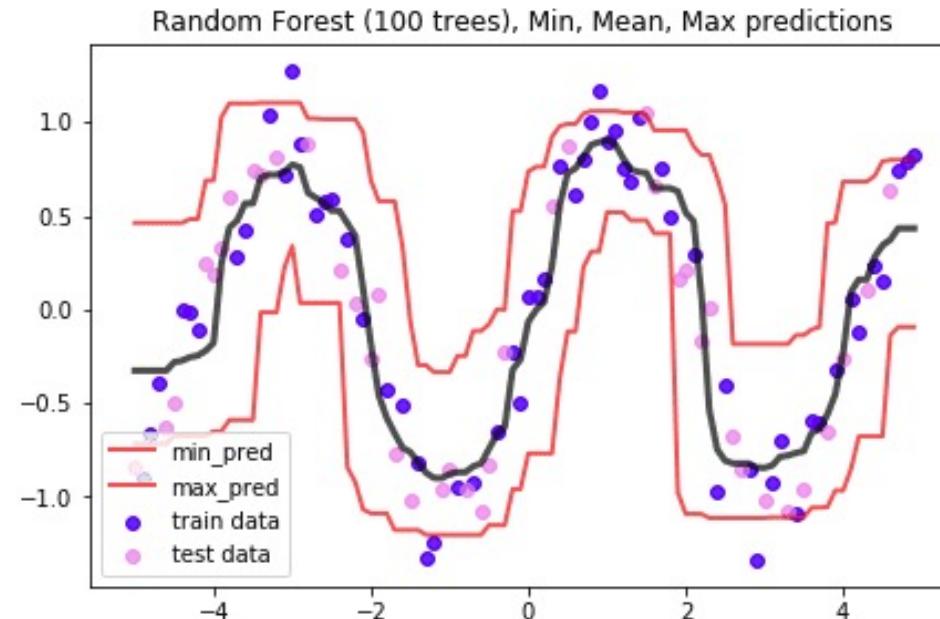
Para problemas de clasificación, se combinan los resultados de los árboles de decisión usando diferentes estrategias. Uno de los más comunes es soft-voting (voto suave), mediante el cual se da más importancia a los resultados de mayor coincidencia (0s o 1s).

Para problemas de regresión, la forma habitual de combinar los resultados de los árboles de decisión es tomando el promedio (media aritmética).



Combinación de predicciones

Otra estrategía es definir la propia forma de combinar predicciones. Por ejemplo, se puede obtener los valores mínimos y máximos de cada predicción para definir el intervalo más probable.



Bosques aleatorios

Random Forest en Python

scikit-learn ofrece dos implementaciones de bosques aleatorios:

- Para regresión: [RandomForestRegressor](#)
- Para clasificación: [RandomForestClassifier](#)

Bosques aleatorios

Hiperparámetros útiles

- **n_estimators.** Indica el número de árboles que va a tener el bosque aleatorio. Normalmente, cuantos más árboles es mejor, pero a partir de cierto punto deja de mejorar y se vuelve más lento. El valor por defecto es 100 árboles.
- **n_jobs.** Es el número de núcleos que se pueden usar para entrenar los árboles. Cada árbol es independiente del resto, así que entrenar un bosque aleatorio es una tarea paralelizable. Por defecto se utiliza 1 core de la CPU. Si se usa **n_jobs = -1**, se indica que se quiere usar tantos cores como tenga el equipo de cómputo.
- **max_features.** Para garantizar que los árboles sean diferentes, éstas se entranan con una muestra aleatoria de datos. Si se quiere que sean más diferentes, se puede hacer que distintos árboles usen distintos atributos. Esto puede ser útil especialmente cuando algunas variables están relacionadas entre sí.

Ajustes en los árboles de decisión

- **max_depth**. Indica la máxima profundidad a la cual puede llegar el árbol. Esto ayuda a combatir el **overfitting**, pero también puede provocar **underfitting**.
- **min_samples_split**. Indica el número mínimo de muestras necesarias antes de dividir este nodo. También se puede expresar en porcentaje. Si la cantidad no es suficiente este nodo se convierte en un nodo hoja.
- **min_samples_leaf**. Indica la cantidad mínima de muestras que debe tener un nodo hoja.
- **criterion**. Indica la función que se utilizará para dividir los datos. Puede ser (ganancia de información) gini y entropy (Clasificación). Cuando el árbol es de regresión se usan funciones como el error cuadrado medio (MSE).
- **max_leaf_nodes**. Indica el número máximo de nodos finales.

Bosques aleatorios

Ventajas

- Funciona bien aún sin los ajustes en los hiperparámetros.
- Funciona bien para problemas de clasificación y regresión (pronóstico).
- Al utilizar múltiples árboles se reduce considerablemente el riesgo de sobreajuste.
- Pueden ser utilizados con un amplio número de variables y gran cantidad de datos.
- Es estable con nuevas muestras.

Desventajas

- Es mayor el costo computacional en comparación con la creación y ejecución de un árbol de decisión.
- Puede requerir mayor tiempo de entrenamiento.
- No funciona bien con pequeños datasets.
- Puede ser difícil de interpretar debido a las decenas de árboles de decisión creados en el bosque.

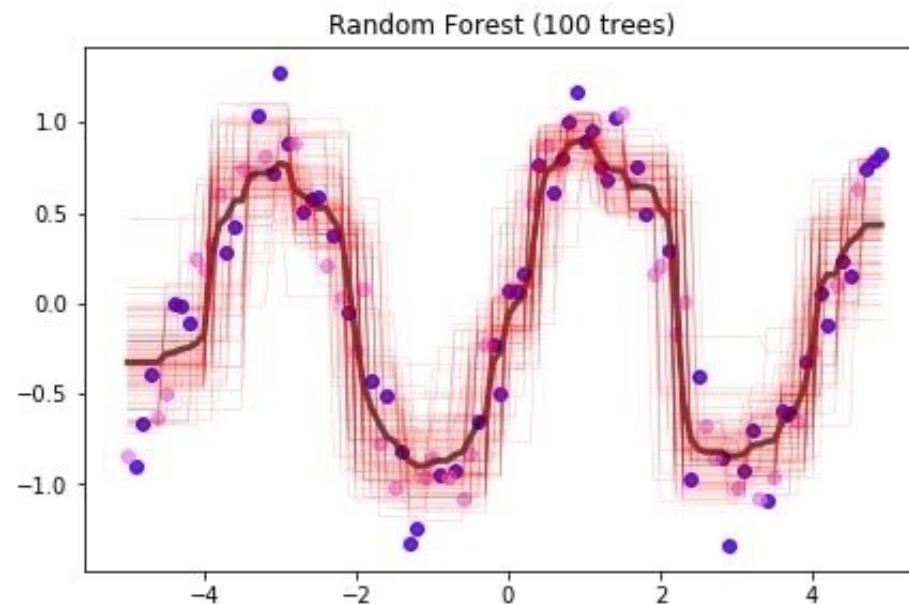
Bosques aleatorios

Regresión

Bosques aleatorios (Regresión)

Criterio de regresión

- Si el objetivo es un valor continuo, entonces, para el nodo (m), los criterios comunes para determinar las ubicaciones para futuras divisiones son el **error cuadrático medio (MSE)**, y el **error absoluto medio (MAE)**.
- La desviación de **MSE** establece el valor pronosticado de los nodos terminales con respecto el valor medio aprendido (\bar{y}_m).



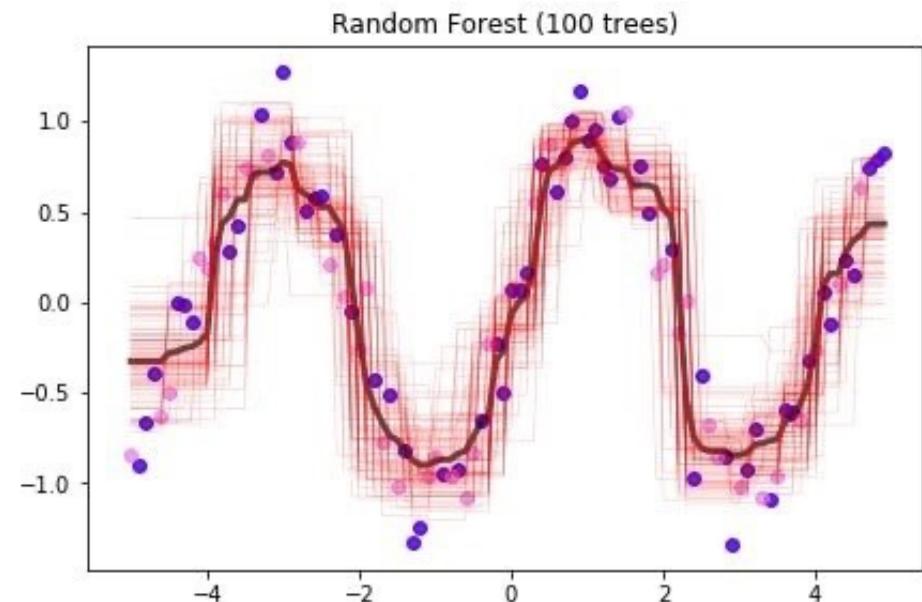
$$\bar{y}_m = \frac{1}{N_m} \sum_{y \in m}^N y$$

$$MSE = \frac{1}{N_m} \sum (y - \bar{y}_m)^2$$

Bosques aleatorios (Regresión)

Criterio de regresión

- Mientras que el **MAE** establece el valor pronosticado de los nodos terminales con respecto a la mediana: $\text{median}(y)_m$



$$\text{median}(y)_m = \text{median}(y)$$

$$MAE = \frac{1}{N_m} \sum |y - \text{median}(y)_m|$$

Bosques aleatorios (Regresión)

- Posterior de la estimación del **MSE** se obtiene la raíz del error cuadrático medio (**RMSE**), para dar una estimación en términos de la calidad del pronóstico.

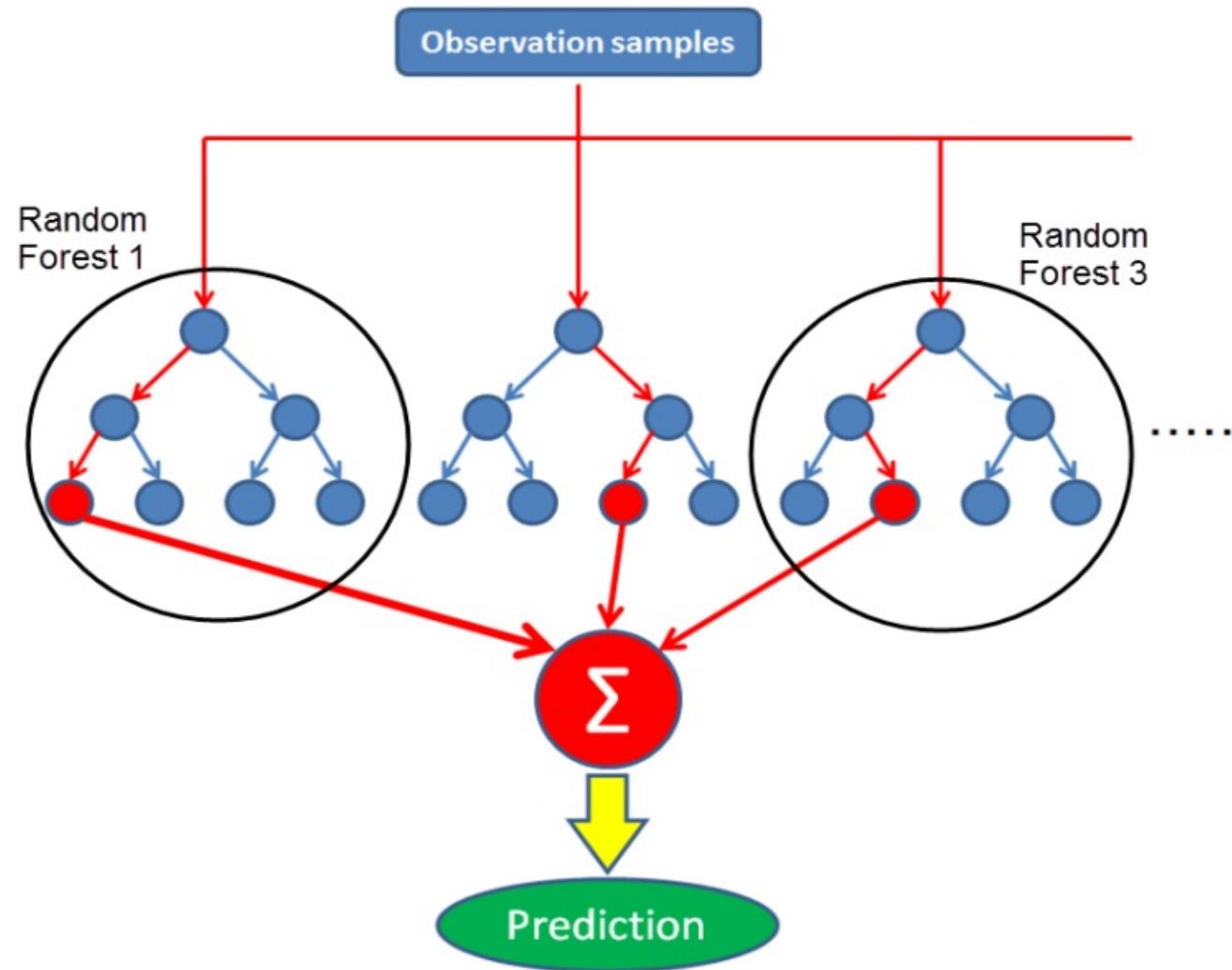
$$RMSE = \sqrt{\frac{1}{N_m} \sum (y - \bar{y}_m)^2}$$

- Se puede obtener también el Score (coeficiente de determinación) para medir la efectividad del modelo de regresión.

$$\bar{R}^2 = 1 - \frac{n - 1}{n - k - 1} [1 - R^2]$$

Número de elementos Número de variables independientes Coeficiente de correlación (valor real y valores pronosticados)
 $\text{corr}(Y, \hat{Y})$

Bosques aleatorios (Regresión)



Bosques aleatorios

Clasificación

Bosques aleatorios (Clasificación)

Criterio de clasificación

- Se utiliza el concepto de **entropía**, la cual es una medida de incertidumbre (información).

$$\text{Entropia}(S) = I(S) = Inf(S) = \sum_{i=1}^n -p_i \log_2 p_i$$

S : es una colección de elementos (objetos).

p_i : es la probabilidad de posibles valores.

$+p$ y $-p$: son la proporción de elementos positivos y negativos en S .

$$GanInf(S, A) = Entropia(S) - \sum_{v \in V(A)} \frac{|Sv|}{|S|} Entropia(Sv)$$

S : es una colección de elementos.

A : son las variables (atributos).

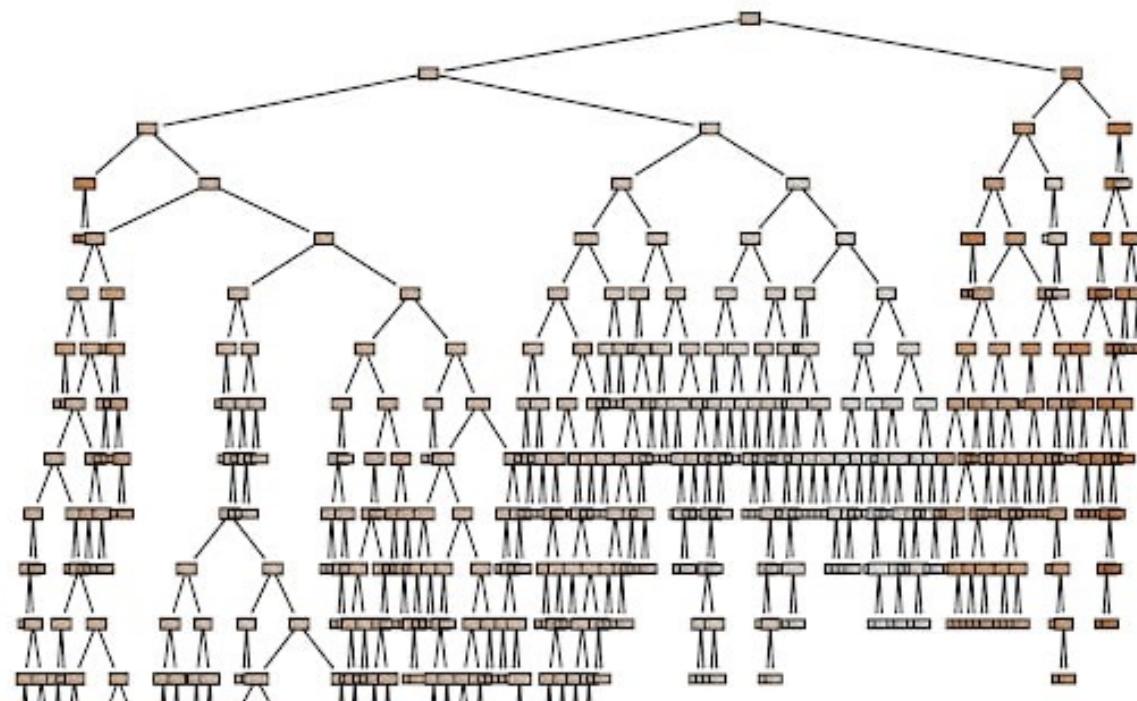
Sv : es un subconjunto de elementos.

$V(A)$: es el conjunto de valores que A (un atributo) puede tomar.

Bosques aleatorios (Clasificación)

Criterio de clasificación

1. Se calcula la entropía para todas las clases y atributos.
2. Se selecciona el mejor atributo basado en la **ganancia de información** de cada variable.
3. Se itera hasta que todos los elementos sean clasificados.



Bosques aleatorios (Clasificación)

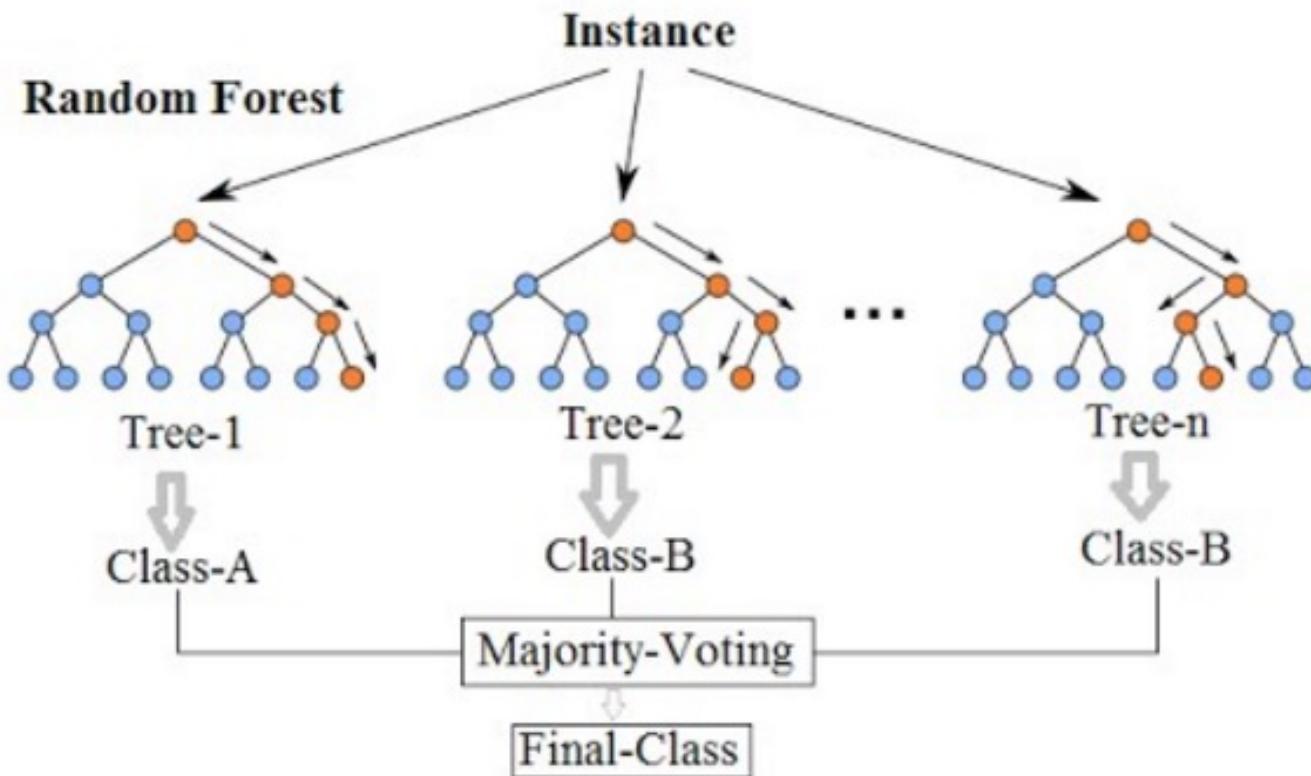
Validación a través de una matriz de clasificación

- 1) Se evalúan todos los elementos y se determina si la **predicción (clase)** coincide con los **valores reales (Y)**.
- 2) Se cuentan todos los elementos y se muestran los totales obtenidos en la matriz.

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

- 1) Exactitud (Accuracy)
- 2) Tasa de error (Misclassification Rate)
- 3) Precisión (Precision)
- 4) Sensibilidad (Recall, Sensitivity, True Positive Rate)
- 5) Especificidad (Especificity, True Negative Rate)

Bosques aleatorios (Clasificación)





Universidad Nacional Autónoma de México
Facultad de Ingeniería

Máquinas de soporte vectorial (SVM, Support Vector Machines)

Noviembre, 2021

Conclusión del semestre 2022-1

Ítem	Fecha	Actividad académica
1	Martes 30 de noviembre de 2021	Práctica 15 (SVM)
2	Jueves 2 de diciembre de 2021	Práctica 16 (RL, AD, BA, SVM)
3	Martes 7 de diciembre de 2021	Examen parcial II
4	Jueves 9 de diciembre de 2021	Entrega del proyecto final
5	Viernes 10 de diciembre de 2021	Último día de entrega de prácticas pendientes y reportes de lectura
6	Martes 14 de diciembre de 2021	Entrega de calificaciones (último corte)
7	Jueves 16 de diciembre de 2021	Examen final (estudiantes previamente notificados)

Contexto



Clasificación

Binaria: Cada elemento puede pertenecer a una de dos clases.

$$f: \rightarrow \{-1,1\}, \{0,1\}$$

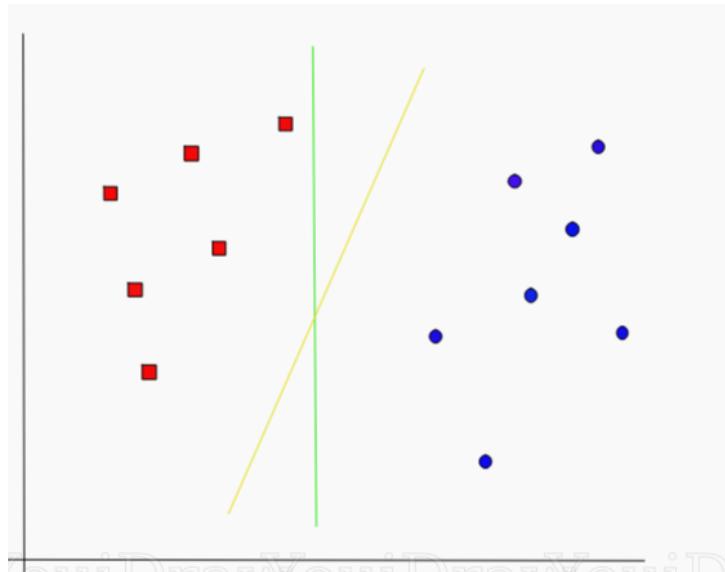
Multiclasé : Cada elemento puede pertenecer a una de las K clases.

$$f: \rightarrow \{1, \dots, K\}$$

Máquinas de soporte vectorial

Contexto

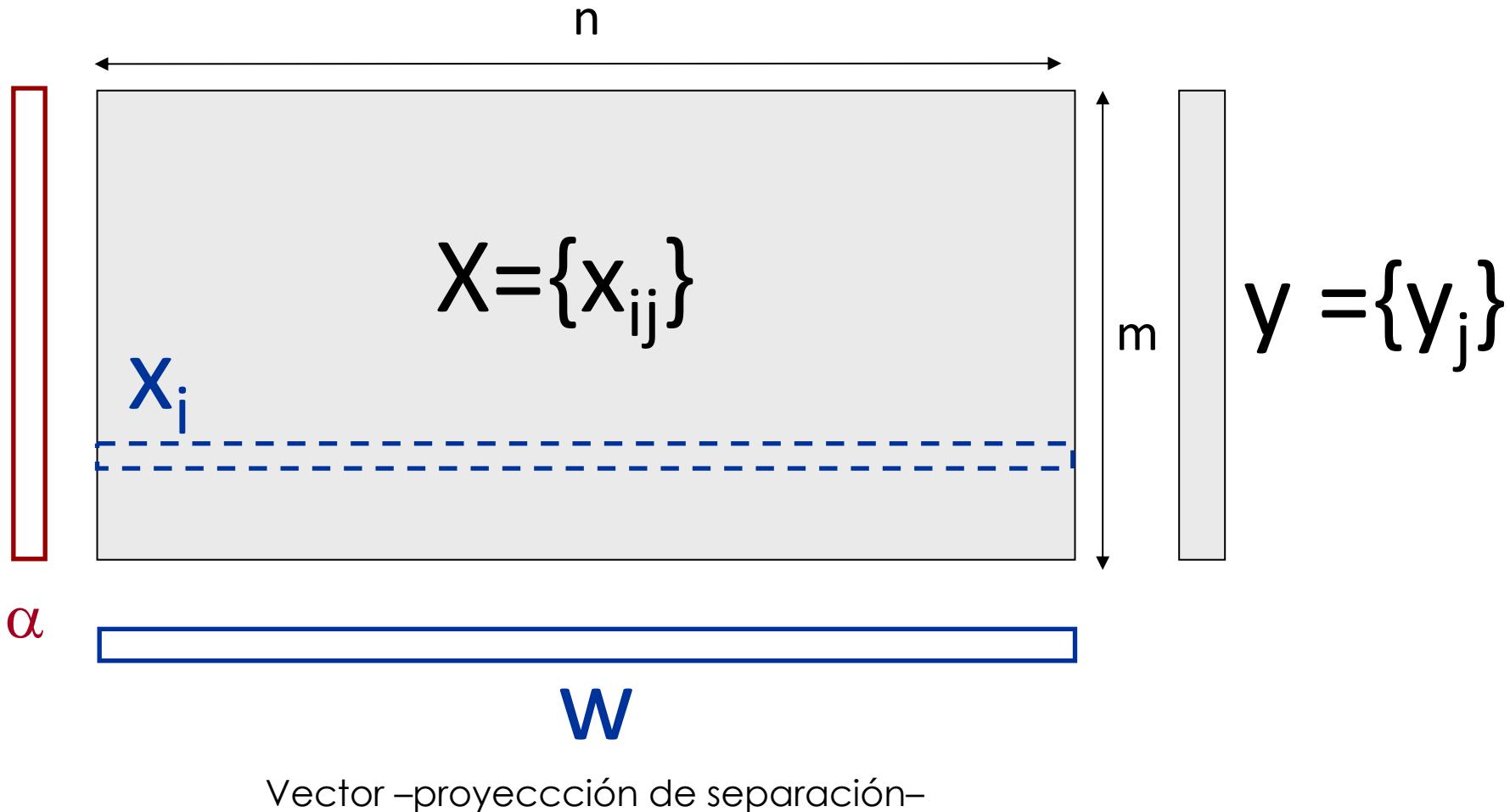
- **SVM** (máquinas de soporte vectorial) es un algoritmo que toma los datos como entrada y genera una línea que separa a estos datos en dos clases.
- A esta línea de separación se conoce como hiperplano.
- **Por ejemplo**, si se tiene un conjunto de datos y se necesita clasificar los **rectángulos** de los **círculos** (los positivos de los negativos), **entonces**, la tarea del algoritmo es encontrar una línea ideal (óptima) que separe los datos en dos clases.



Pueden existir diversas líneas que separan las dos clases, pero se debe encontrar el hiperplano óptimo que maximiza el margen de separación.

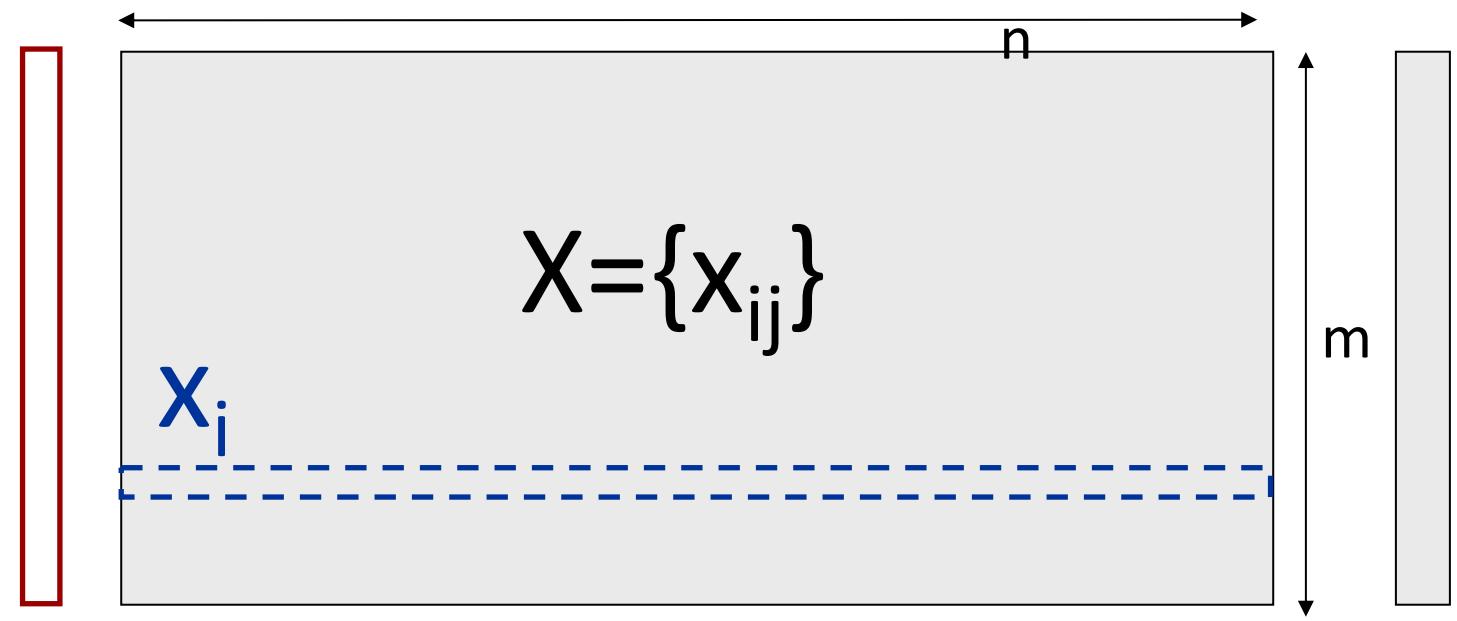
Máquinas de soporte vectorial

Contexto



Máquinas de soporte vectorial

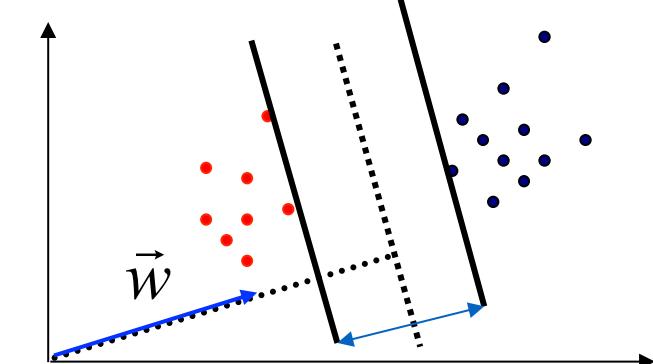
Contexto



α

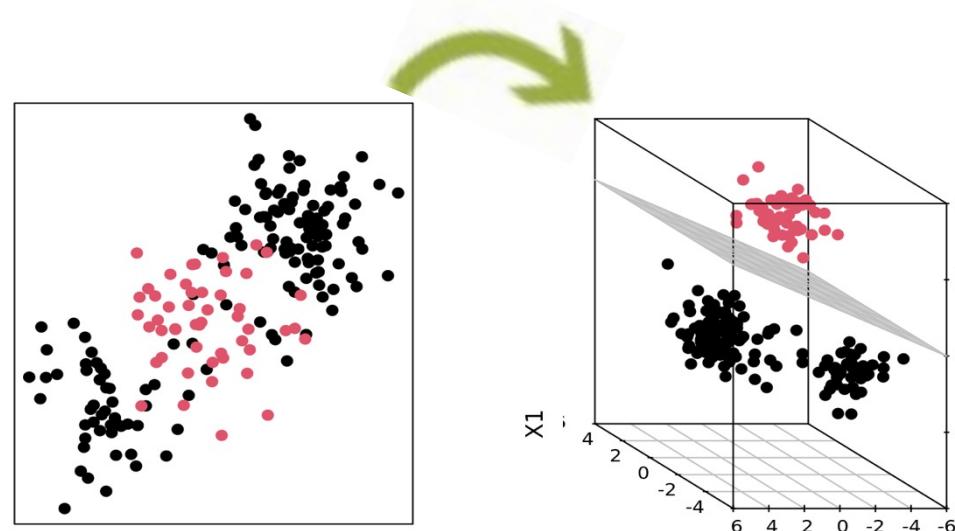
W

Vector –proyección de separación–



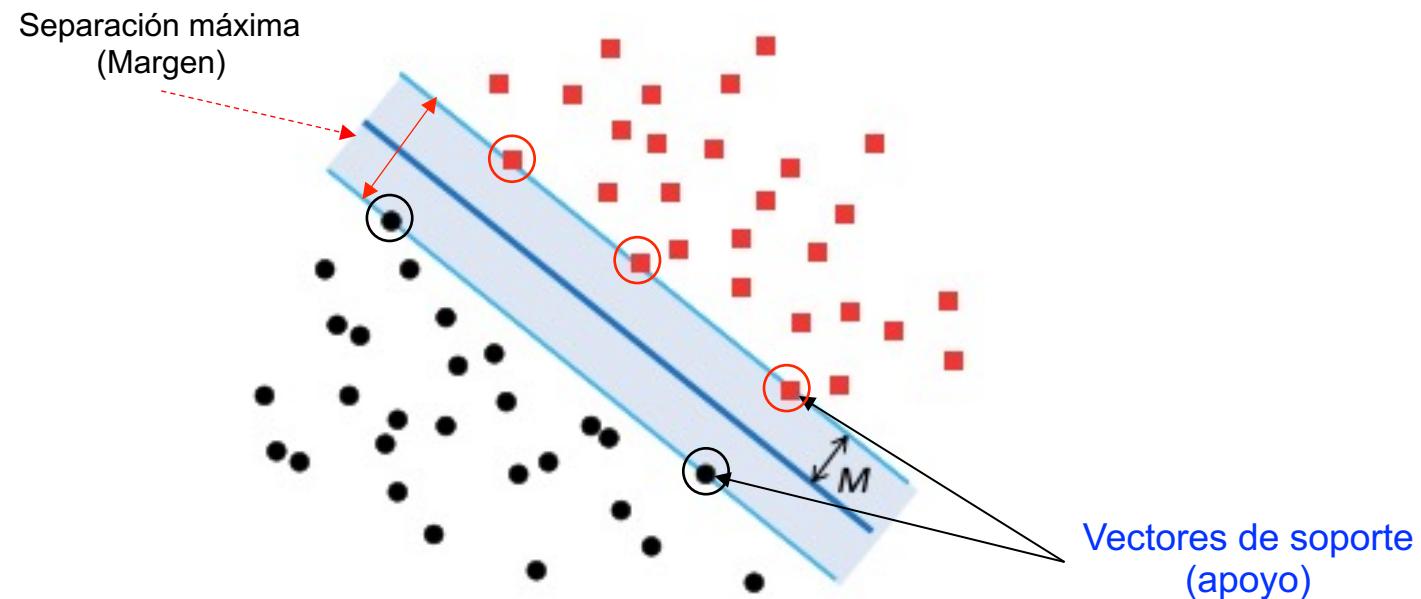
Máquinas de soporte vectorial

- **SVM**, tal como se entiende actualmente, fue presentado en una conferencia –COLT– (COmputational Learnig Theory) por Vapnik, Boser y Guyon en 1992.
- Posteriormente, fue descrito con mayor detalle en 1995 (Cortes y Vapnik) y 1998 (Vapnik) para pasar de la formulación teórica a su aplicación práctica en problemas reales de reconocimiento de patrones (*pattern recognition*).
- En la actualidad, el interés por este algoritmo no ha dejado de crecer en su aplicación a problemas reales. Se utiliza tanto para clasificación como para pronóstico (regresión).



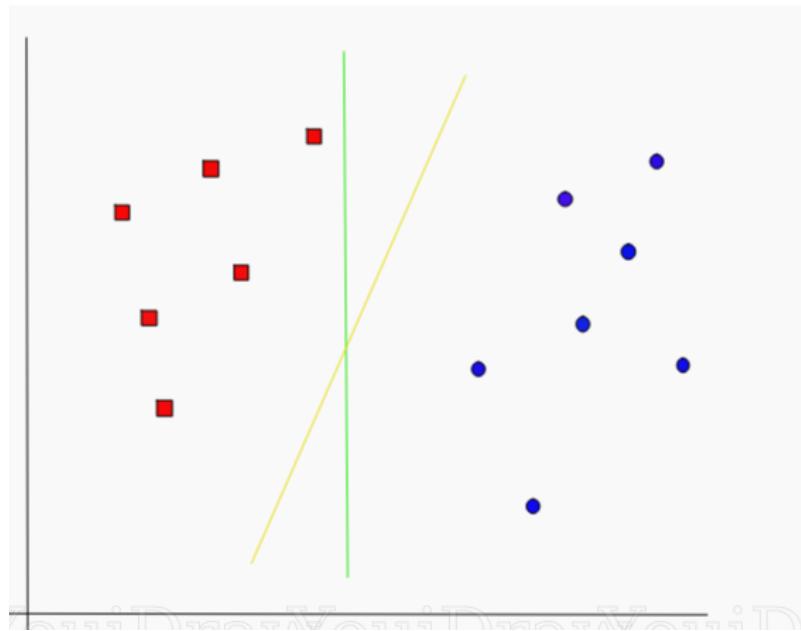
Máquinas de soporte vectorial

- En el presente, **SVM** constituye un referente en el aprendizaje automático, sobre todo para encontrar la manera óptima de resolver problemas de clasificación.
- Intuitivamente, una SVM es un modelo que representa los puntos en el espacio, separando las clases en dos (espacios) mediante un hiperplano.
- Se busca la **separación máxima posible** entre los puntos más cercanos a cada clase (vectores de soporte).



Máquinas de soporte vectorial

- Sin embargo, encontrar esa **máxima separación posible** hace que se puedan generar infinitas líneas (hiperplanos) que traten de separar las dos clases. Entonces, ¿cómo SVM encuentra el hiperplano ideal?



Si son dos candidatos: línea verde y amarillo. ¿Cuál de esas líneas separa mejor los datos?

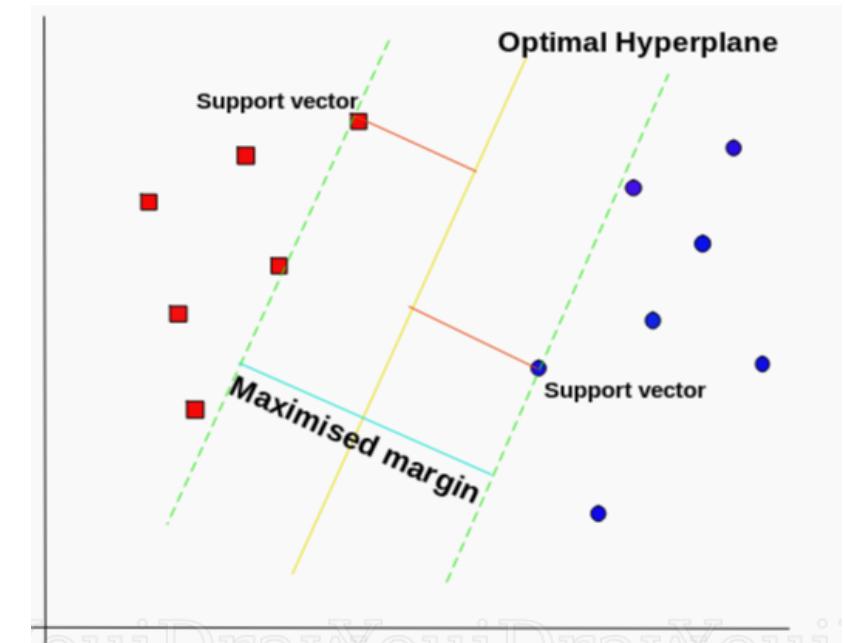
- La línea verde está cerca de la clase roja. Aunque puede clasificar los datos actuales, no sería una línea óptima, dado que se busca un separador generalizado.
- La línea amarilla evidentemente es la más adecuada, debido a que hace un mejor separación (posición neutra).
- De eso trata el algoritmo de SVM, encontrar la mejor línea de separación.

Máquinas de soporte vectorial

Procedimiento

SVM intenta establecer un límite de decisión de tal manera que la separación entre las dos clases sea la más amplia posible.

- 1) Se identifican los puntos más cercanos a la línea de ambas clases (frontera).
- 2) A estos puntos se denominan vectores de soporte.
- 3) Luego se calcula la **distancia** entre los hiperplanos y los vectores de soporte.
- 4) Esta distancia se llama margen (M).
- 5) El objetivo es maximizar el margen.
- 6) El hiperplano cuyo margen es mayor, es considerado como el hiperplano óptimo.



Máquinas de soporte vectorial

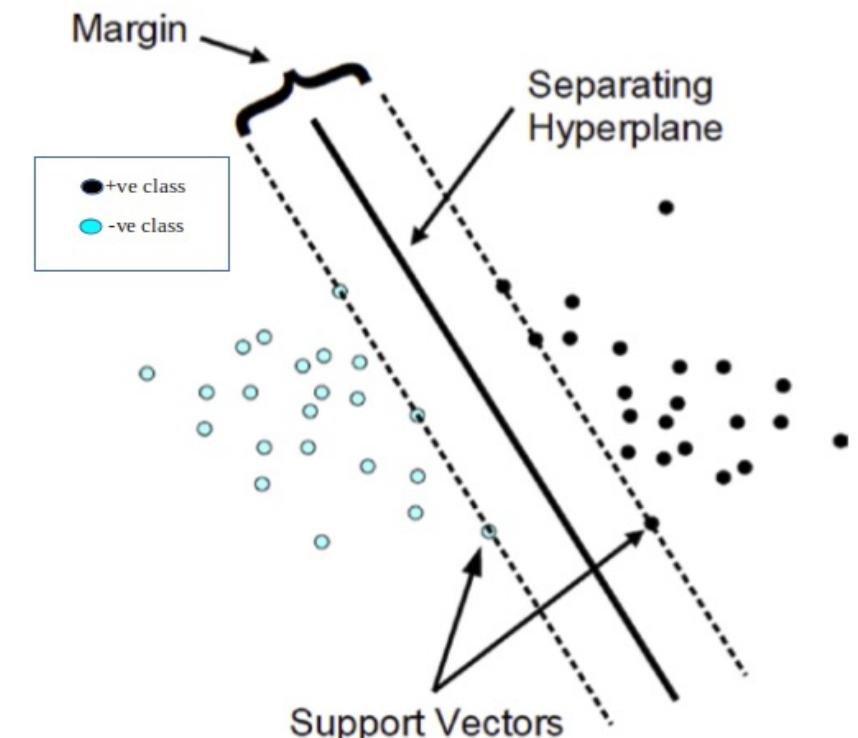
Vectores de soporte

- El **separador de margen máximo** se determina por un subconjunto de puntos de datos (vectores de soporte).
- Computacionalmente es útil una pequeña fracción de puntos (vectores de soporte), dado que éstos se utilizan para decidir en qué lado del separador se clasificará cada caso de prueba.

$$Y = f[wx + b > 0] = \begin{cases} +1 & wx + b > 0 \\ -1 & wx + b \leq 0 \end{cases}$$

$$b_0 + b_1x_1 + b_2x_2 + \dots + b_px_p > 0, \text{ si } y_i=1$$

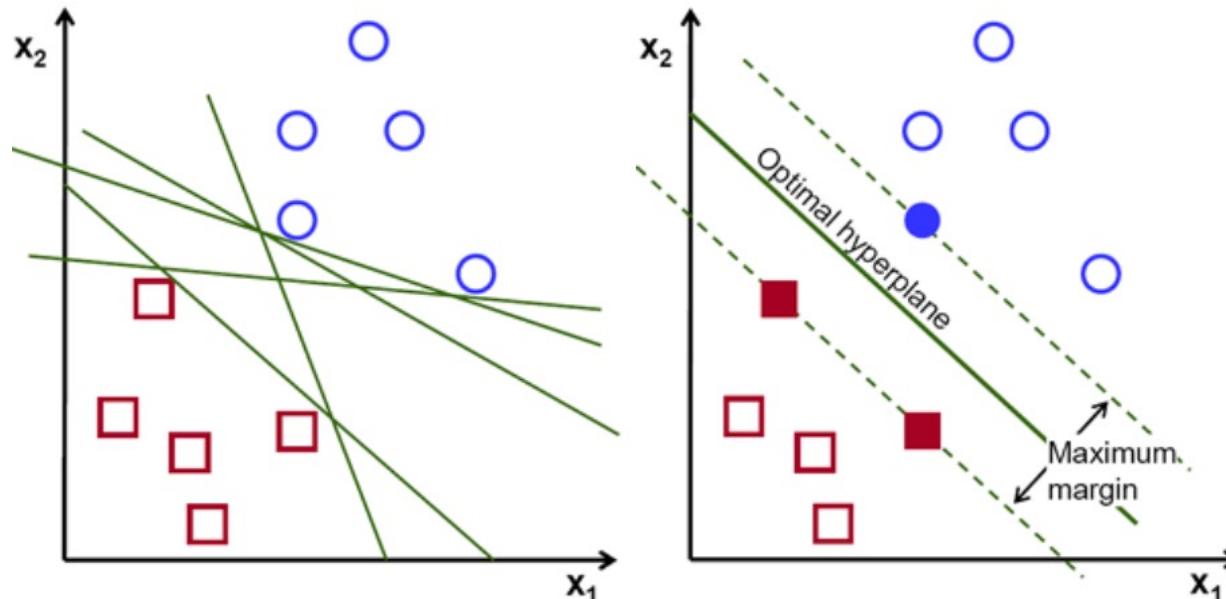
$$b_0 + b_1x_1 + b_2x_2 + \dots + b_px_p < 0, \text{ si } y_i=-1$$



Máquinas de soporte vectorial

Vectores de soporte

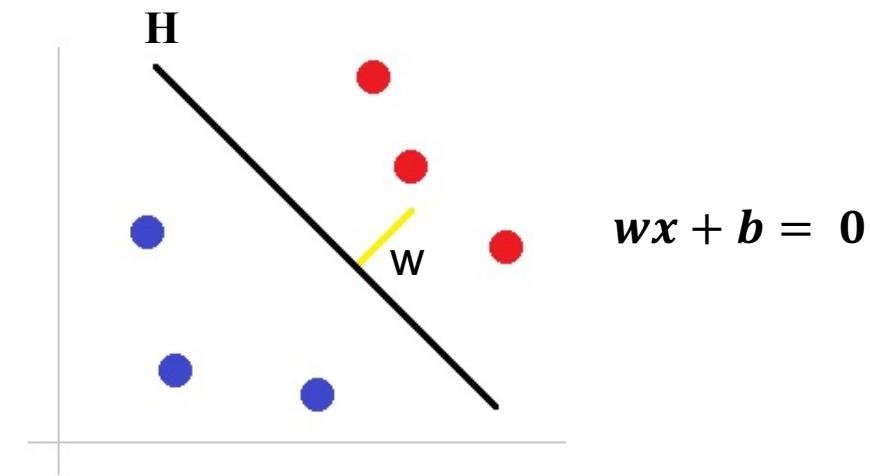
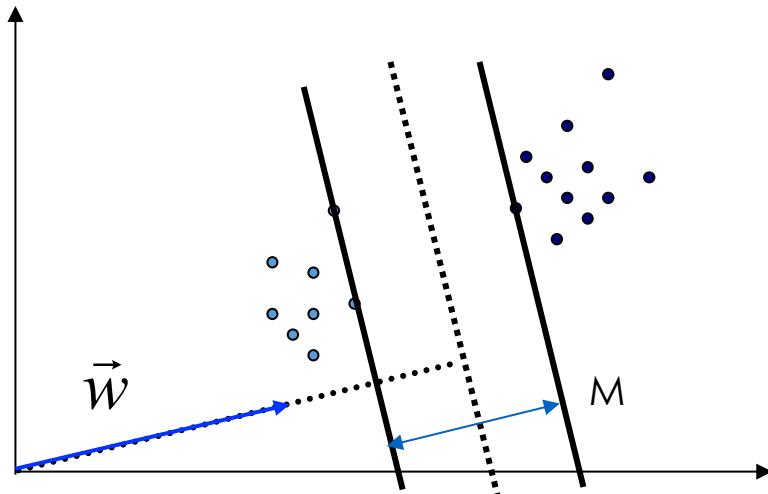
- Cuanto más alejados del hiperplano estén los vectores de soporte (SV), mayor será la probabilidad de clasificar correctamente los nuevos casos en sus respectivas clases.
- Los SV son claves para determinar el hiperplano. Si la posición de los vectores cambia, cambia también el hiperplano (se altera).



Máquinas de soporte vectorial

Hiperplano

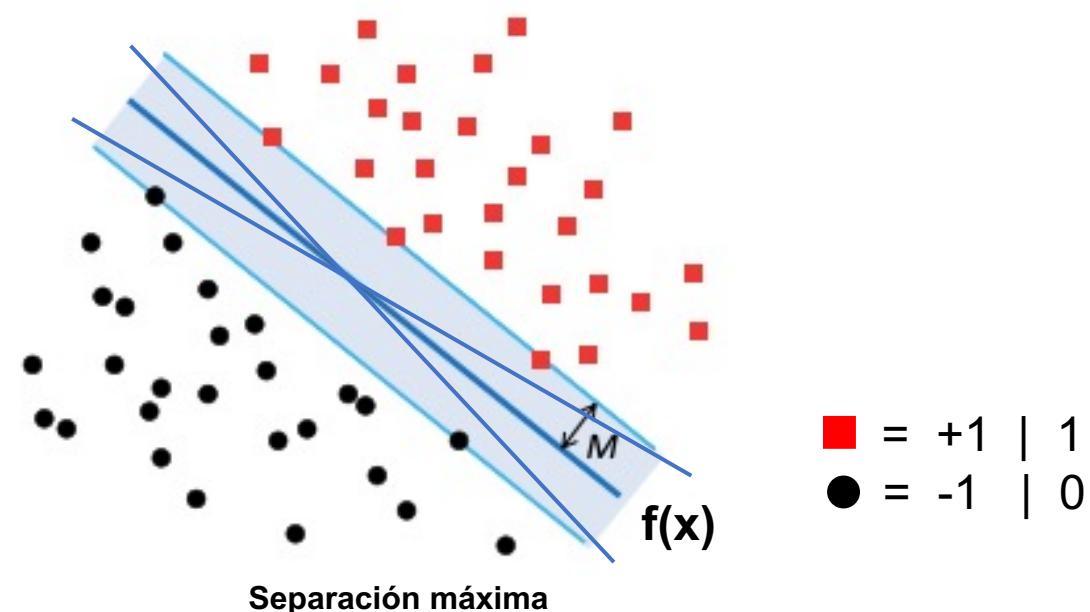
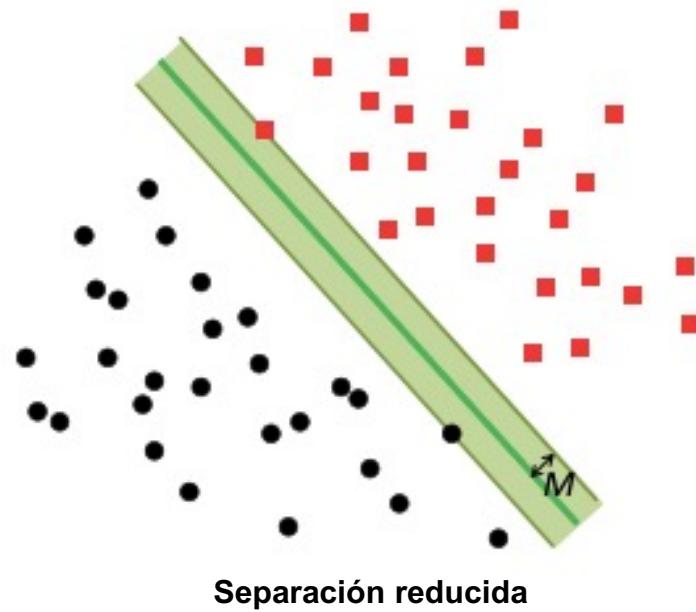
- Un hiperplano es una función que se utiliza en un espacio euclíadiano n-dimensional que divide en dos partes a los vectores de datos.
- **SVM** puede ser utilizado para conjuntos de datos linealmente separables, así como para los no linealmente separables.
- Para medir el hiperplano separador se utiliza el vector (**w**) y la ordenada al origen (**b**) – constante–:



Máquinas de soporte vectorial

Hiperplano

Asumiendo un conjunto de datos $D (x_i, y_i)_{i=1 \dots n}$ asociados a una etiqueta de clase $y_i \in \{-1, 1\}$, separables mediante un hiperplano.

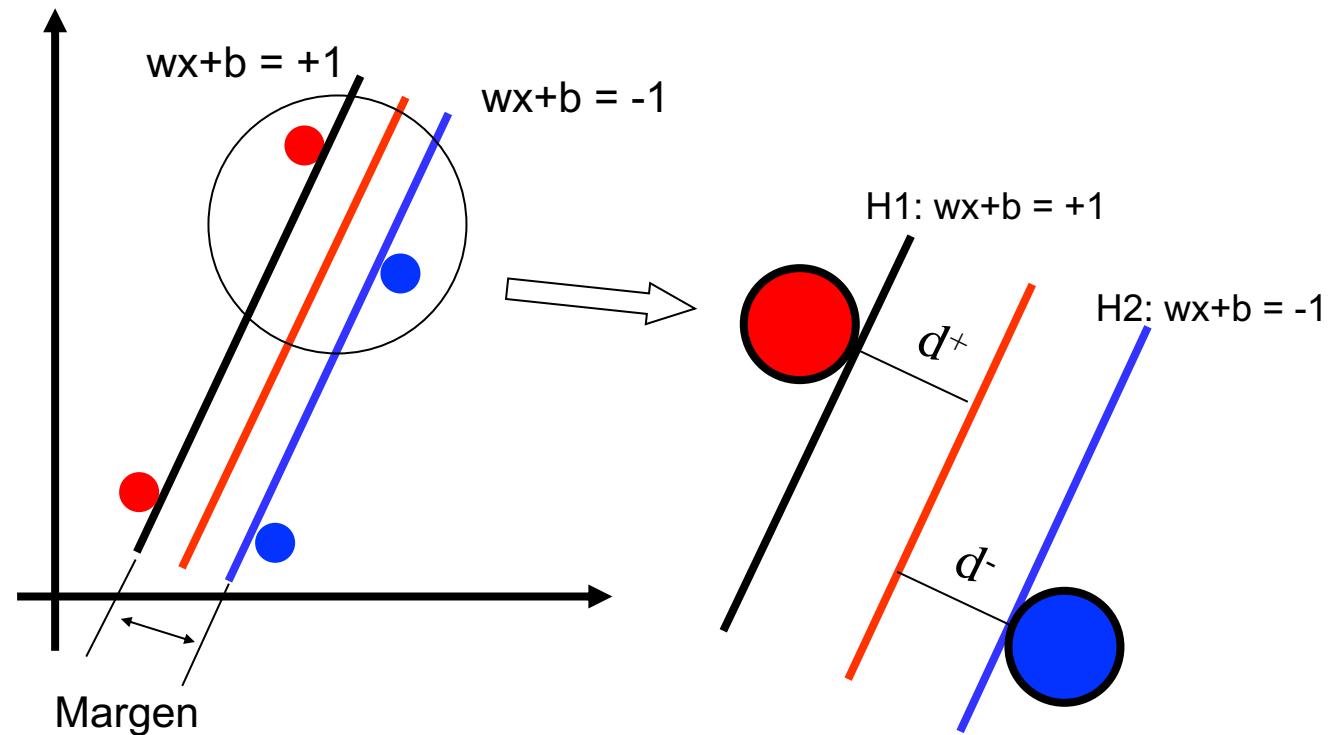


Sobre las diversas líneas de separación, se encuentra la que maximiza el margen (óptimo).

Máquinas de soporte vectorial

Distancia

- El mejor hiperplano de separación debe estar situado en la posición más neutra posible con respecto a las clases.
- Se considera los puntos que están en la frontera de la región de decisión, dado que es la zona donde puede haber dudas sobre a qué clase pertenece el elemento.



d^+ = la distancia más corta al punto positivo más cercano.

d^- = la distancia más corta al punto negativo más cercano.

El **margin** de un hiperplano de separación es $d^+ + d^-$.

Máquinas de soporte vectorial

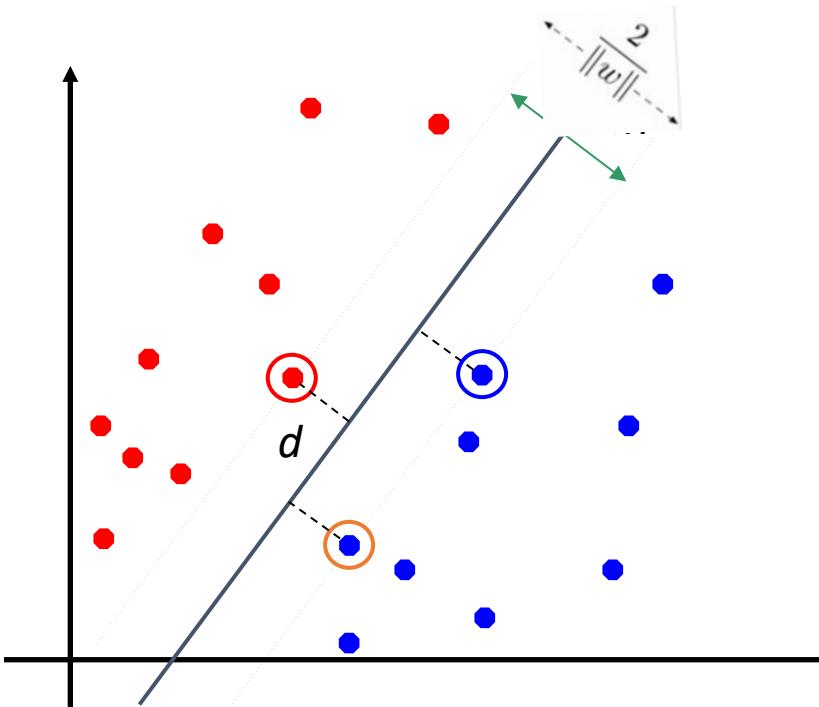
Distancia

La distancia del ejemplo \mathbf{x}_i al separador $\mathbf{H}(\mathbf{w}, \mathbf{b})$ es:

$$d = \frac{\mathbf{w}\mathbf{x}_i + b}{\|\mathbf{w}\|} = \frac{\pm 1}{\|\mathbf{w}\|}$$

longitud euclíadiana $\sqrt{\mathbf{w} \cdot \mathbf{w}}$

El **margin** del separador es la distancia entre los vectores de soporte: $M = 2d = \left| \frac{1}{\|\mathbf{w}\|} - \frac{-1}{\|\mathbf{w}\|} \right| = \frac{2}{\|\mathbf{w}\|}$



Máquinas de soporte vectorial

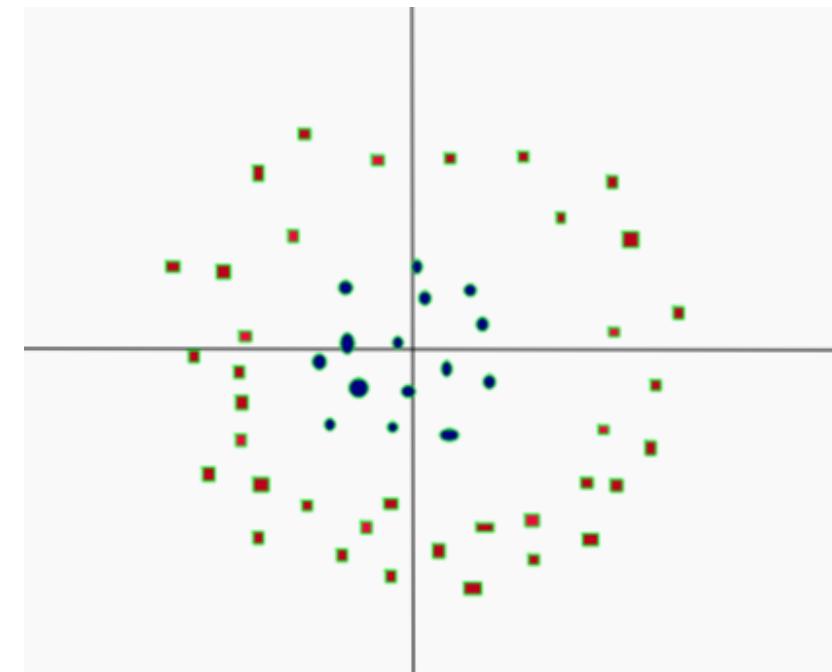
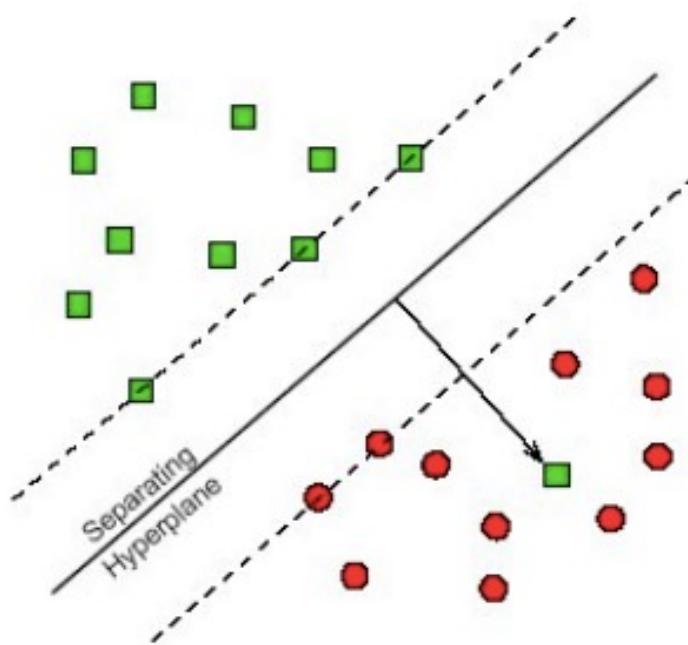
Complejidad

- La complejidad depende de la cantidad de elementos (objetos) de entrenamiento.
- Al ser un algoritmo de clasificación para datos lineales y no lineales, con un mapeo no lineal apropiado, los datos de entrenamiento pueden ser separados por un hiperplano.
- SVM encuentra ese hiperplano óptimo usando vectores de soporte (datos de entrenamiento) para medir los **márgenes de separación**.

Máquinas de soporte vectorial

Complejidad

- Ruido (clases no linealmente separables).

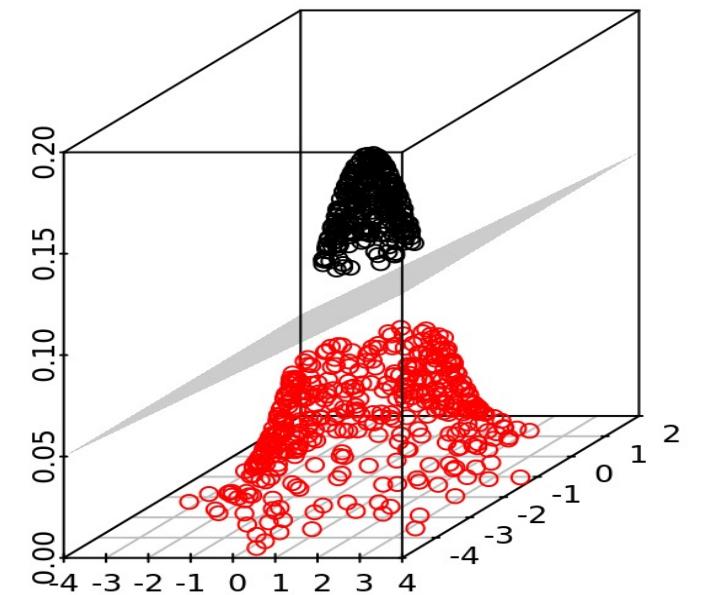
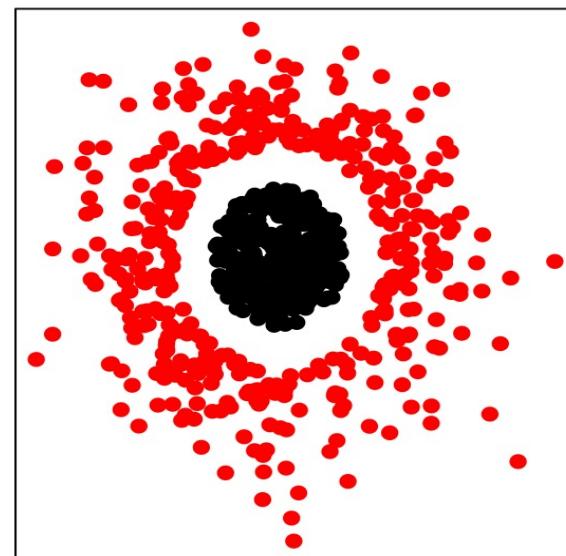
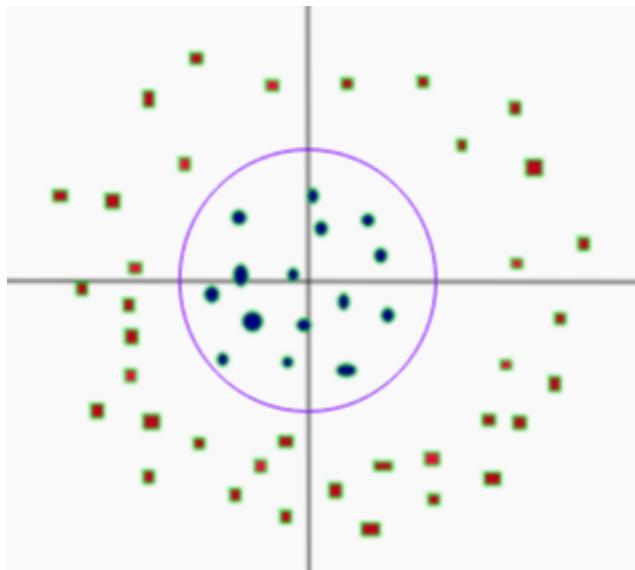


Estos datos no se pueden separar linealmente, pero se puede aumentar la dimensión (función matemática) para convertirlos en datos linealmente separables.

Máquinas de soporte vectorial

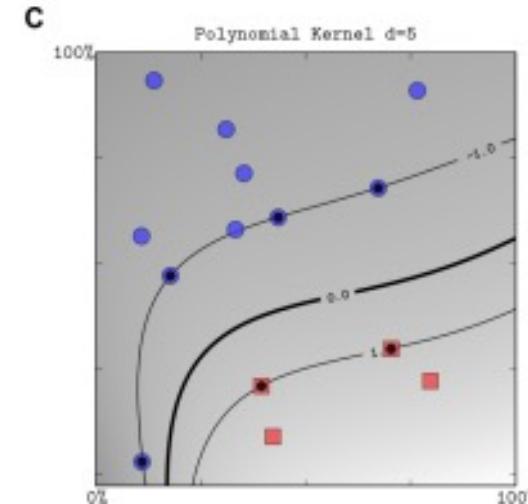
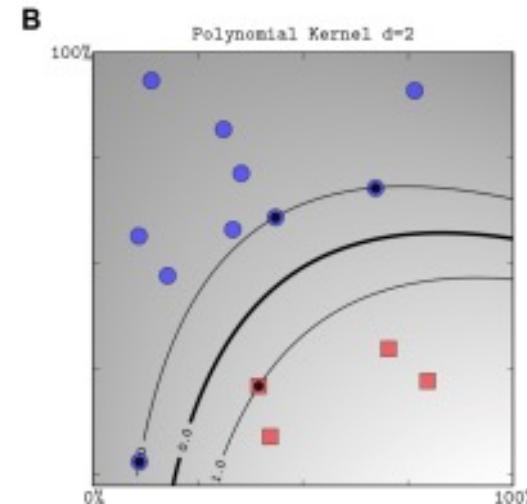
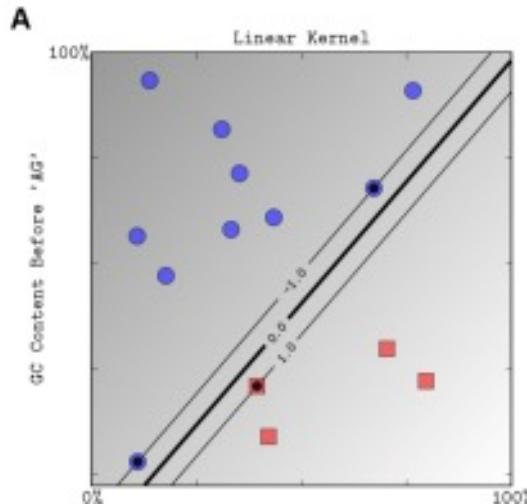
Aumento de la dimensionalidad

Al aumentar la dimensionalidad, los datos pueden ser claramente separables. Por ejemplo, a través de la ecuación de un círculo (radio): $r^2 = x^2 + y^2$ se puede proyectar un separador en una dimensión superior.

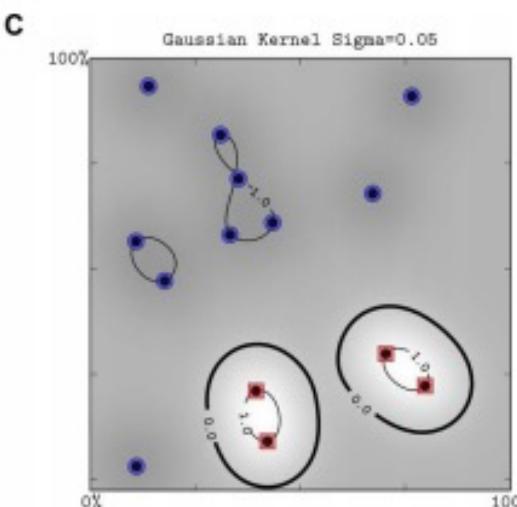
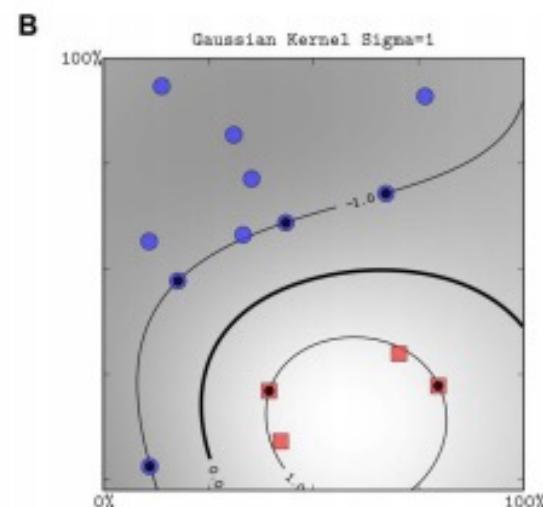
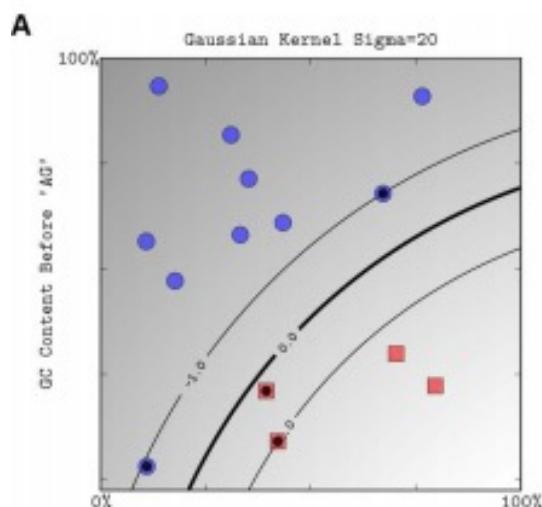


Máquinas de soporte vectorial

- Ruido (clases no linealmente separables).



**Efecto polinomial
(grado)**



Efecto del radio

Máquinas de soporte vectorial

Kernels (funciones)

- Mediante transformaciones matemáticas, se mapean los datos en un mejor espacio de representación por una determinada función, denominada **kernel**.
- Encontrar la transformación correcta para un conjunto de datos no es una tarea fácil. Por lo que, se usan diferentes kernels en una implementación de SVM. Algunos **kernels** utilizados son:

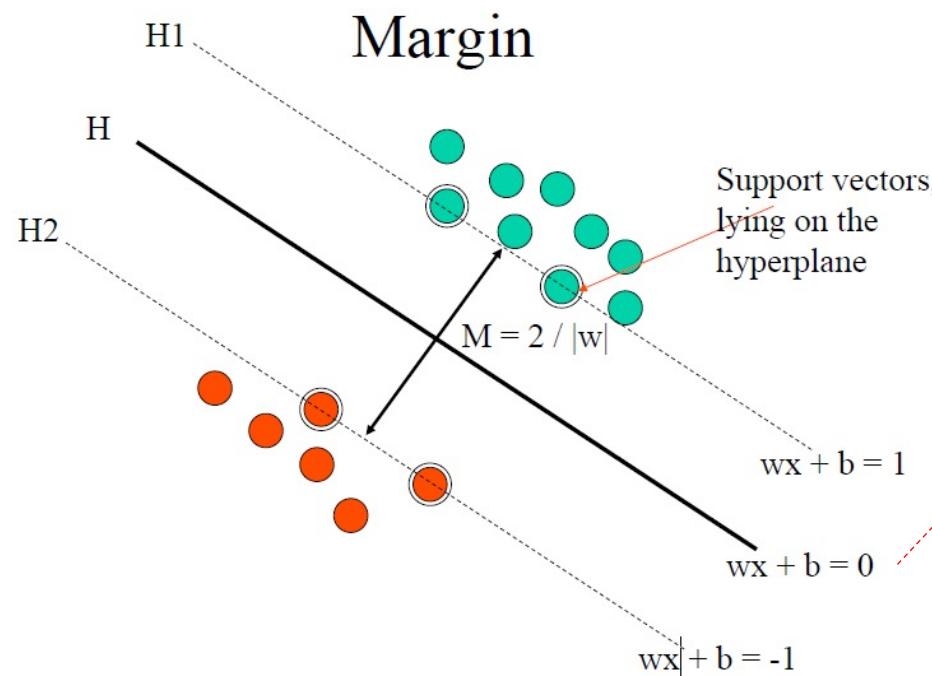
Kernel	Formula
linear	$k(x, y) = x.y$
sigmoïd	$k(x, y) = \tanh(ax.y + b)$
polynomial	$k(x, y) = (1 + x.y)^d$
RBF	$k(x, y) = \exp(-a\ x - y\ ^2)$
exponential RBF	$k(x, y) = \exp(-a\ x - y\)$

Kernel with Moderate Decreasing (KMOD)
$$K(x, y) = a \left[\exp\left(\frac{\gamma}{\|x-y\|^2+\sigma^2}\right) - 1 \right]$$

Máquinas de soporte vectorial

Kernels (funciones)

El objetivo es encontrar el hiperplano $f(x) = \mathbf{w}x + b$ que mejor clasifique los datos (la que minimiza el error de clasificación).



$$f(x) = \begin{cases} K(x, y) = x \cdot y \\ K(x, y) = \tanh(ax \cdot y + b) \\ K(x, y) = (1 + x \cdot y)^d \\ K(x, y) = a \left[\exp\left(\frac{\gamma}{\|x-y\|^2 + \sigma^2}\right) - 1 \right] \\ K(x, y) = \exp(-a\|x - y\|^2) \\ K(x, y) = \exp(-a\|x - y\|) \end{cases}$$
$$Y_i \in \begin{cases} +1 & | 1 & \text{wx} + b > 0 \\ -1 & | 0 & \text{wx} + b \leq 0 \end{cases} \begin{matrix} \text{Positivos} \\ \text{Negativos} \end{matrix}$$

Todos los hiperplanos de separación están expresados por: $\mathbf{w}x + b = 0$

Máquinas de soporte vectorial

Ventajas

- Es altamente eficaz cuando las clases son separables (espacios de gran dimensión).
- Las SVM son adecuadas para la clasificación binaria.
- Pueden ser utilizados con un amplio número de variables y gran cantidad de datos.
- Se pueden especificar diferentes funciones del núcleo para la función de decisión.
- Es estable con nuevas muestras.
- SVM se basa en propiedades geométricas de los datos, mientras que la regresión logística se basa en enfoques estadísticos.
- El riesgo de sobreajuste es menor en SVM, en comparación con los Árboles de Decisión.

Desventajas

- Para conjuntos de datos grandes requiere mayor tiempo para procesarse (alto costo computacional).
- Puede no ser efectivo en el caso de clases múltiples.
- Seleccionar la función de kernel adecuada puede ser una tarea compleja.

Máquinas de soporte vectorial

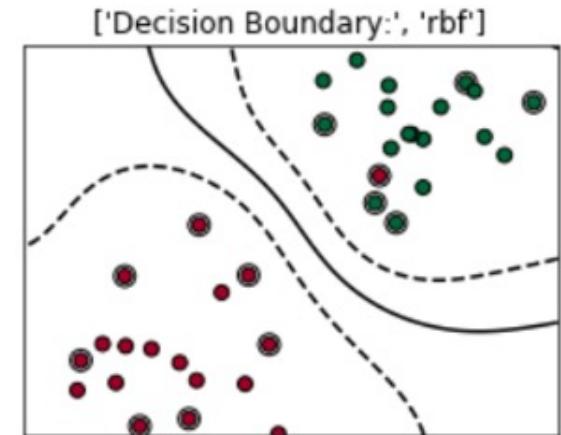
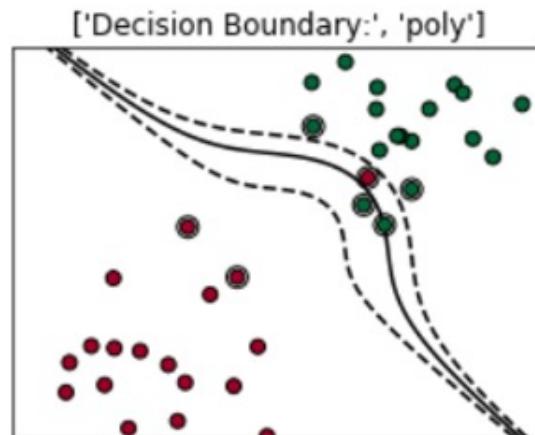
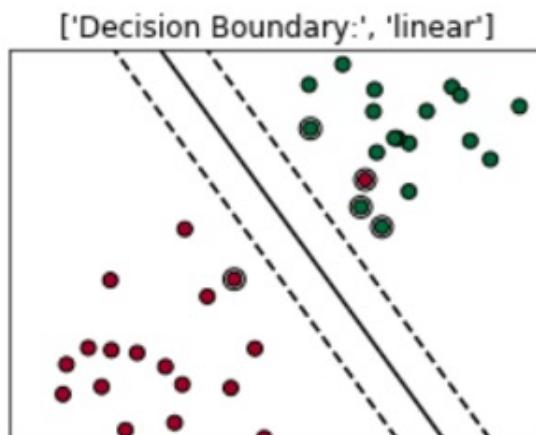
Kernels (funciones)

En Python a través de Sklearn es posible utilizar las funciones nativas existentes, como:

- linear
- poly
- rbf
- sigmoid



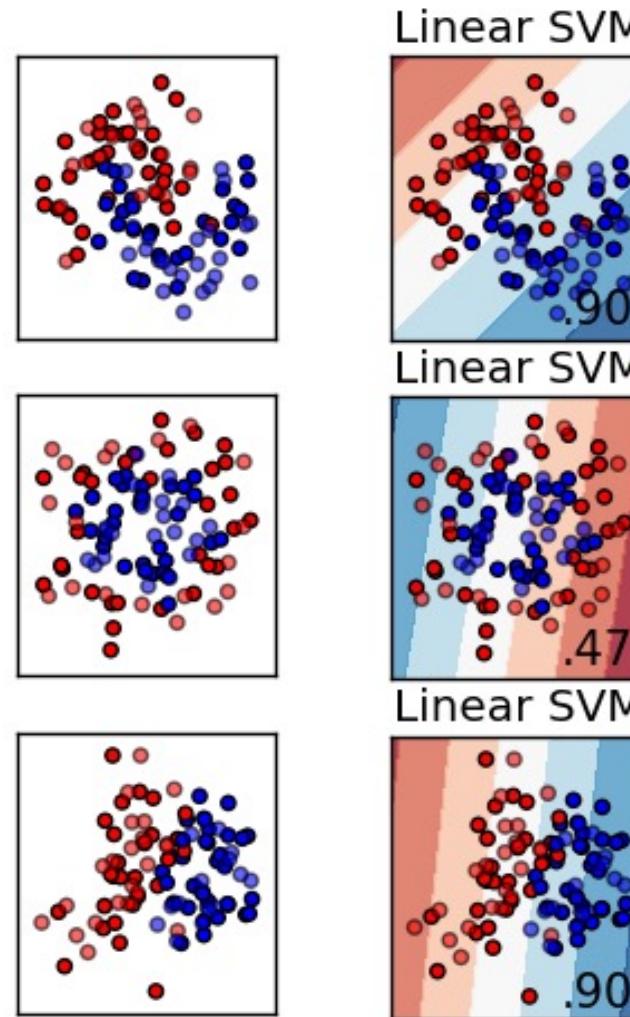
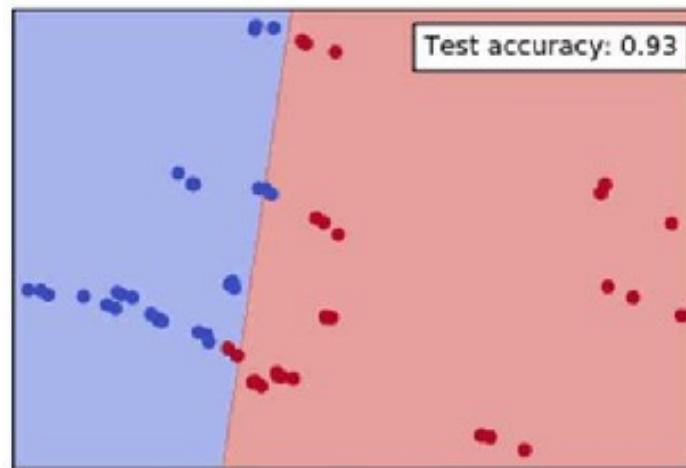
```
from sklearn.svm import SVC #Support vector classifier  
from sklearn import model_selection
```



Máquinas de soporte vectorial

Kernels (funciones)

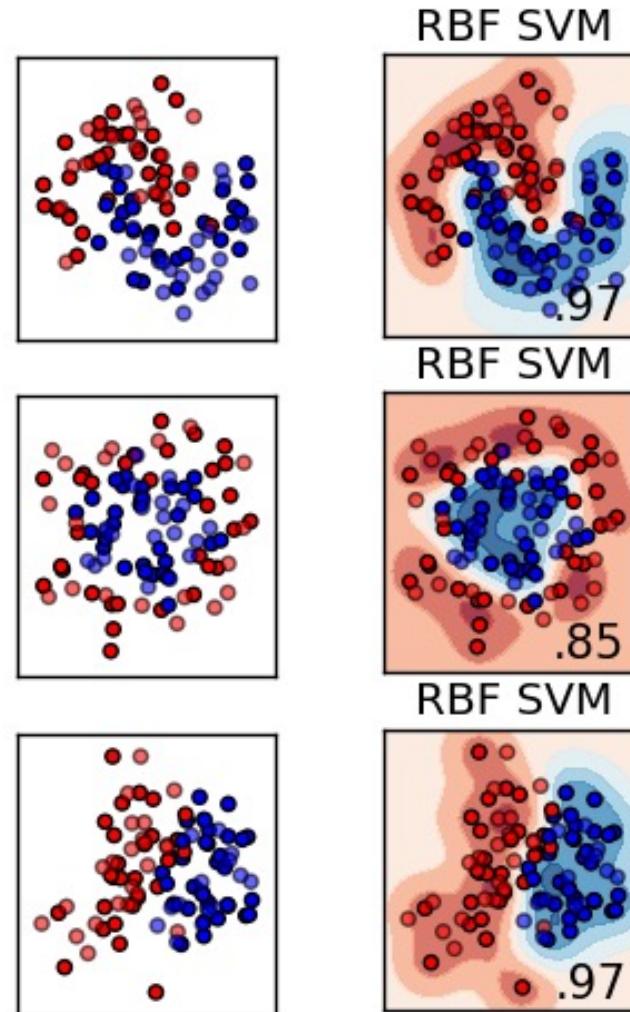
Linear: $K(x, y) = x \cdot y$



Máquinas de soporte vectorial

Kernels (funciones)

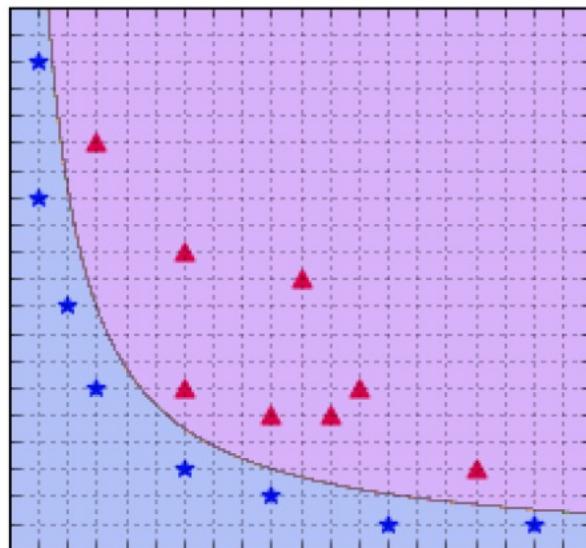
RBF: $K(x, y) = \exp(-a|x - y|^2)$



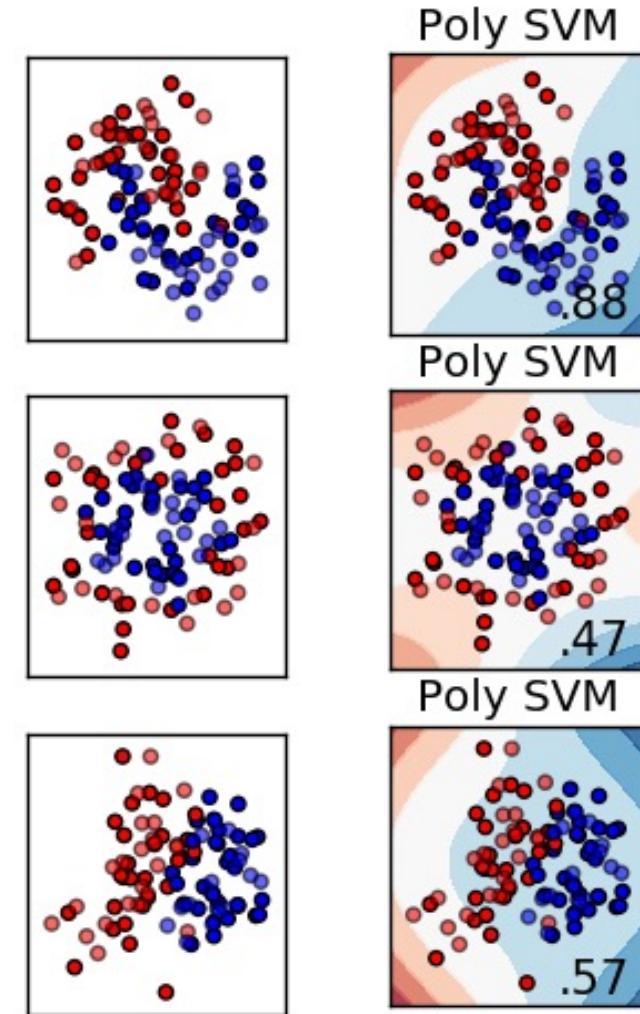
Máquinas de soporte vectorial

Kernels (funciones)

Polinomial: $K(x, y) = (1 + x \cdot y)^d$



Grado 2



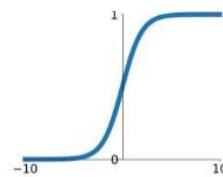
Máquinas de soporte vectorial

Kernels (funciones)

Sigmoide: $K(x, y) = \tanh(ax.y + b)$

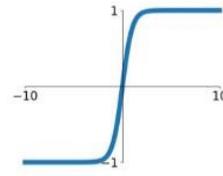
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



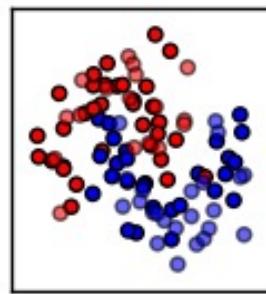
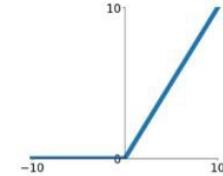
tanh

$$\tanh(x)$$

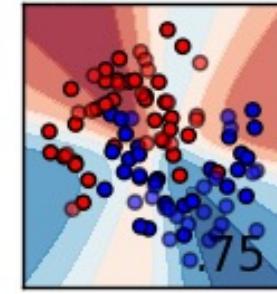


ReLU

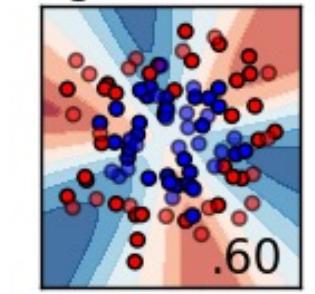
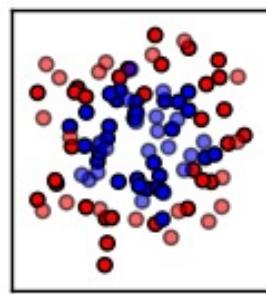
$$\max(0, x)$$



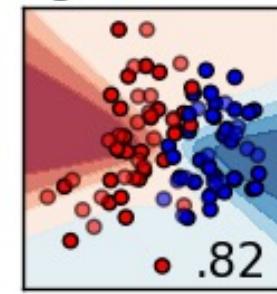
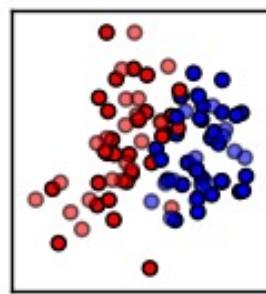
Sigmoid SVM



Sigmoid SVM



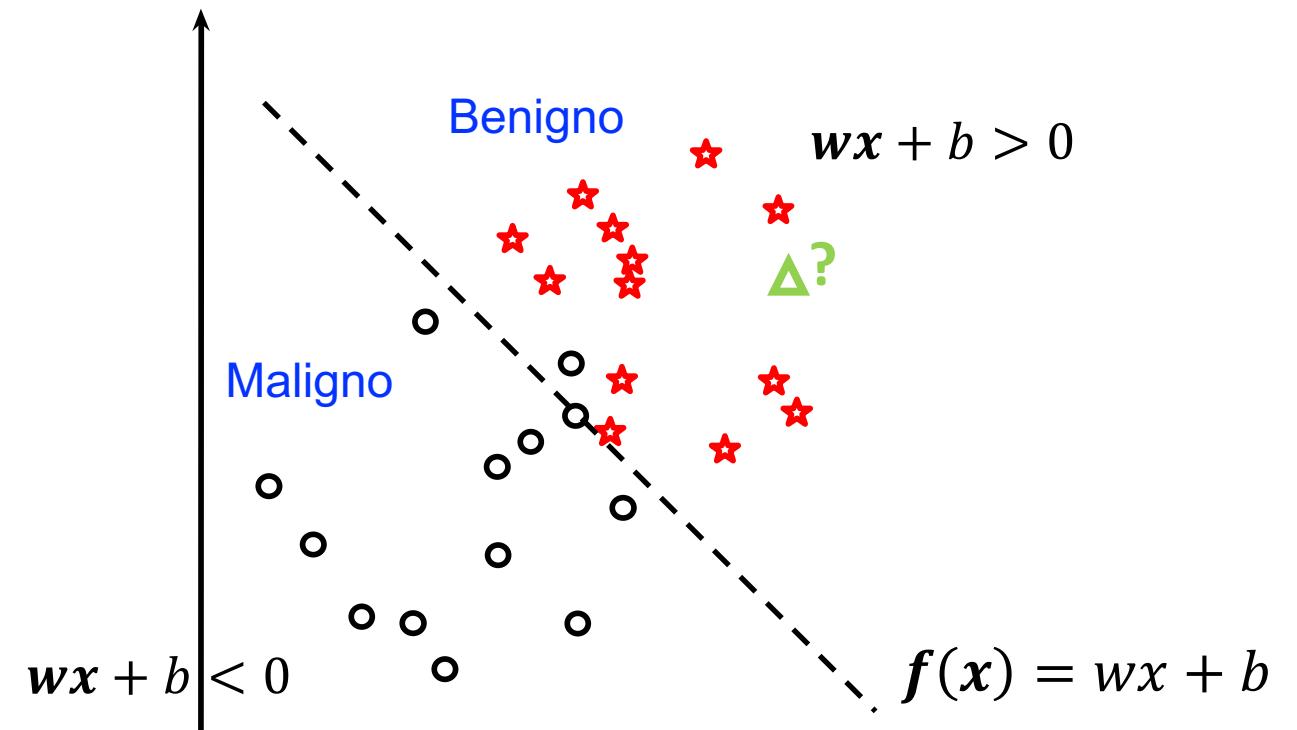
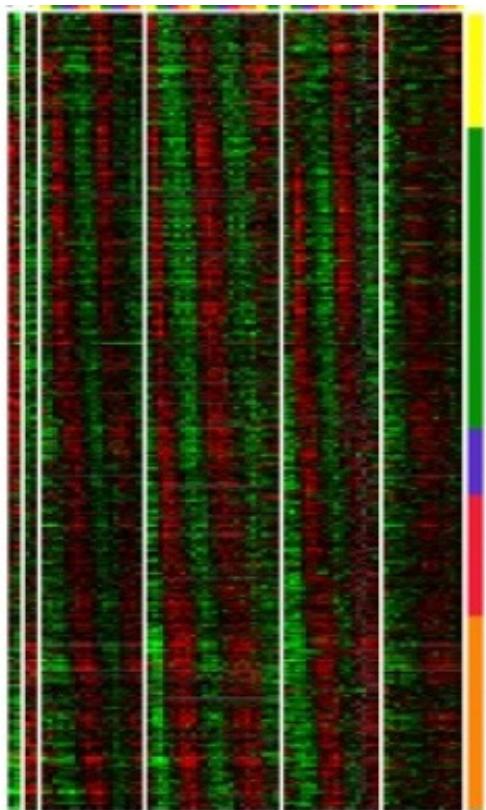
Sigmoid SVM



Máquinas de soporte vectorial

Ejemplo

$$X = [x_1, x_2, \dots, x_n] \quad Y$$



Máquinas de soporte vectorial

Consideraciones finales

- Las SVM actualmente son uno de los algoritmos que mejor desempeño tienen en problemas de clasificación.
- Se pueden aplicar a tipos de datos complejos mediante varios tipos de funciones (kernel).
- Las SVM se puede utilizar para problemas de clasificación mediante vectores de soporte (SVC) y regresión (SVR).
- El ajuste de las SVM sigue siendo un arte: la selección del kernel y parámetros específicos generalmente se hace de manera a prueba y error.