

# Introducción al Machine Learning



Alfonso D Blázquez

# Índice de contenidos

- Data Science, Machine Learning e Inteligencia Artificial: aclarando conceptos
- ¿Qué es el Machine Learning?
  - Tipos de algoritmos de Machine Learning
  - Explicabilidad vs Precisión
  - Sesgo vs Varianza
  - Preprocesamiento de Datos
  - Métricas y métodos de validación
- Modelos supervisados
  - KNN, Naive Bayes, SVM...
  - Ensamblado de modelos
- Modelos no supervisados
  - Clustering
  - PCA
- Caso práctico

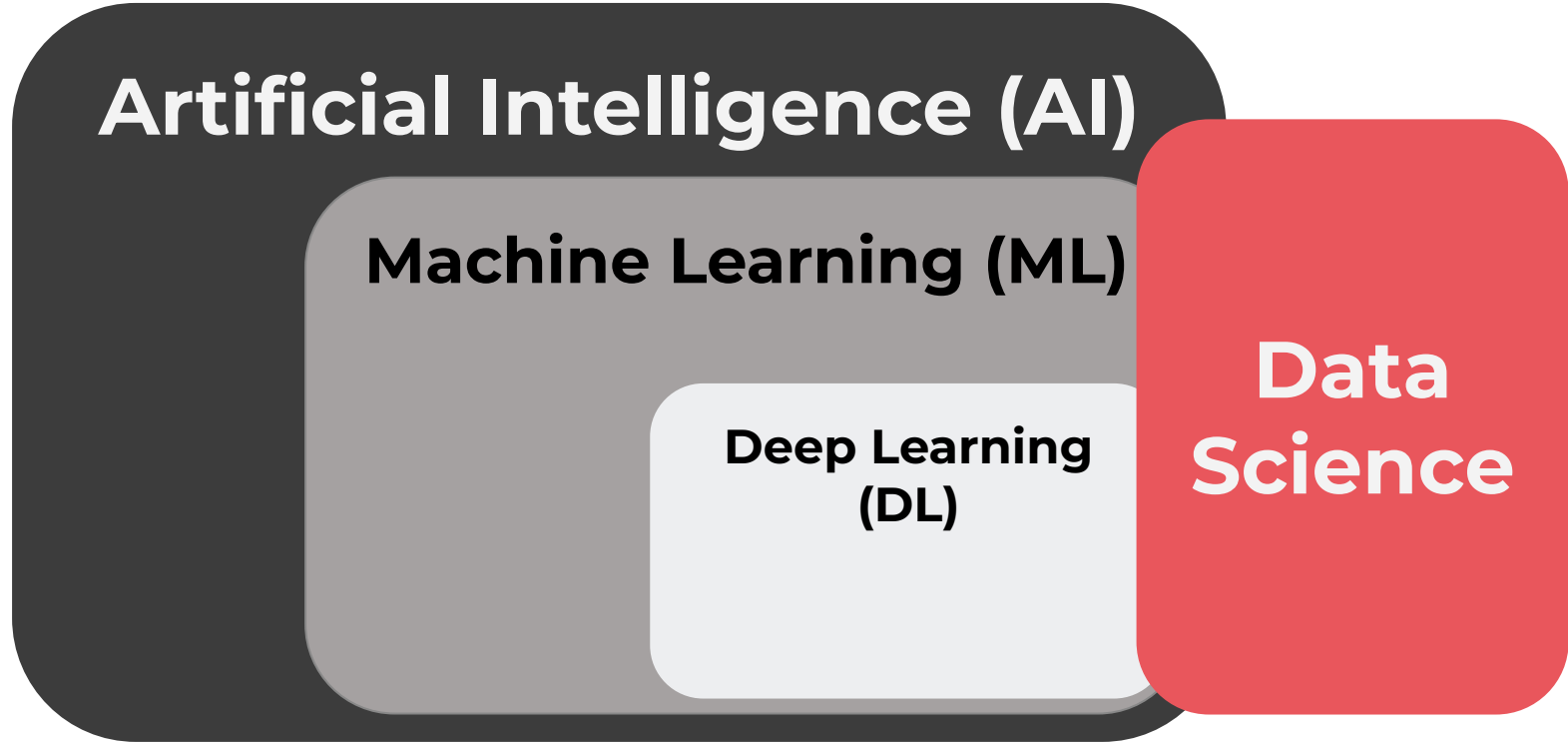
**¿Data Science? ¿Machine Learning? ¿IA?**

**Artificial Intelligence (AI)**

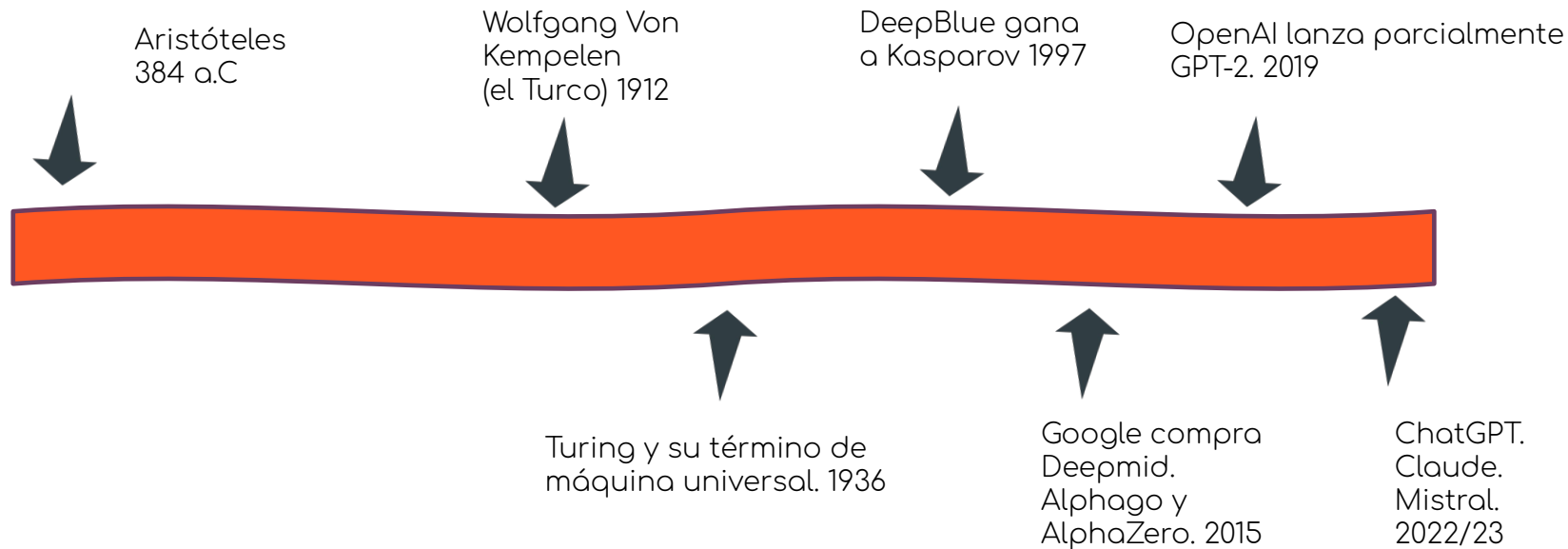
**Machine Learning (ML)**

**Deep Learning  
(DL)**

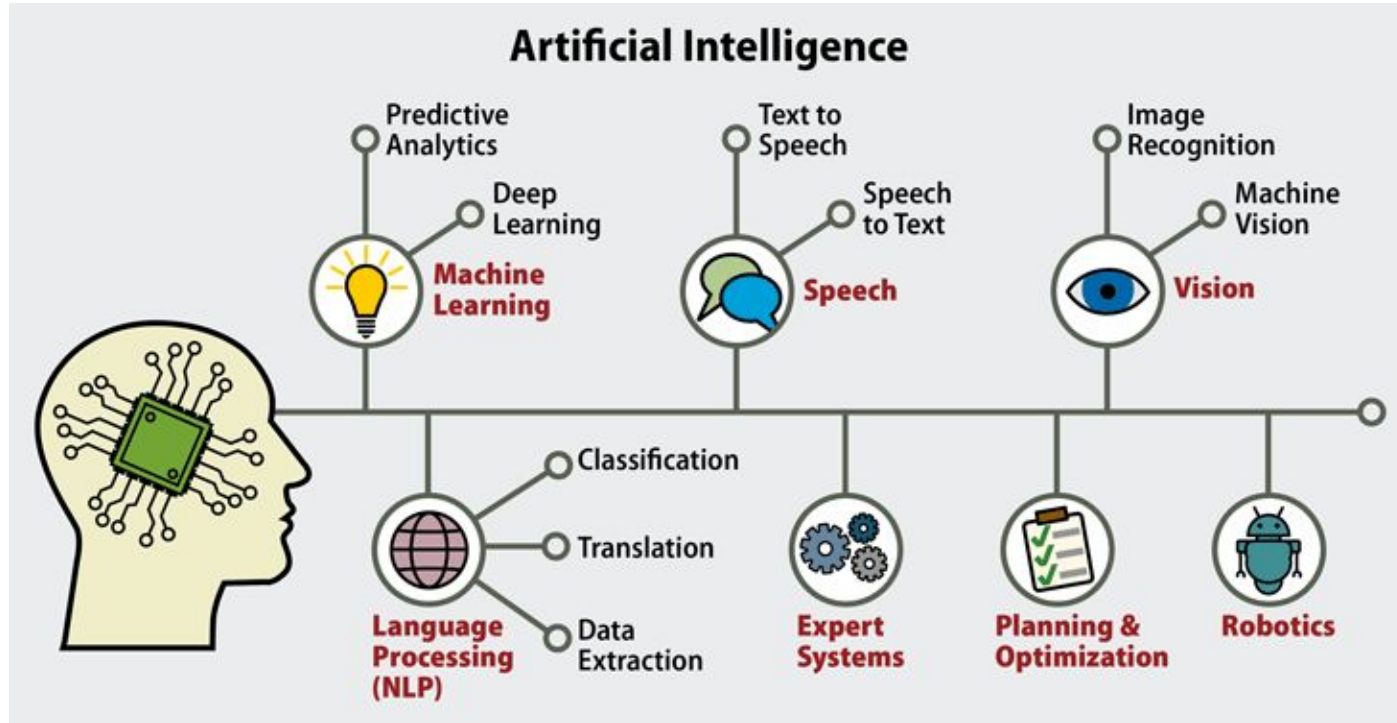
**Data  
Science**



# ¿Data Science? ¿Machine Learning? ¿IA?



# ¿Data Science? ¿Machine Learning? ¿IA?



# ¿Big Data?



- Volumen
- Velocidad
- Variedad
- Variabilidad
- Veracidad
- ¿Visualización?
- ¿Valor?
- etc.

# ¿Qué es el Machine Learning?

Forma parte del área de la inteligencia artificial. Lo que define qué es *machine learning* son las características de los algoritmos que lo complementan.

Un **algoritmo** es un conjunto de instrucciones que solucionan un problema, estas instrucciones deben de ser finitas, ordenadas y lógicas, **en otras palabras:** no pueden ser un número infinito de instrucciones, son secuenciales y deben de carecer de ambigüedad. Cuando se diseña un algoritmo que cumple con estas características, es posible programarlo en un ordenador, el cual podrá seguir dichas instrucciones y solucionar un problema dado.

Entonces estos algoritmos cumplen con las siguientes pautas:

Son capaces de analizar datos, reconocer patrones, aprender de los datos y aplicar lo aprendido para tomar una decisión respecto a nuevos datos.

# ¿Qué es el Machine Learning?

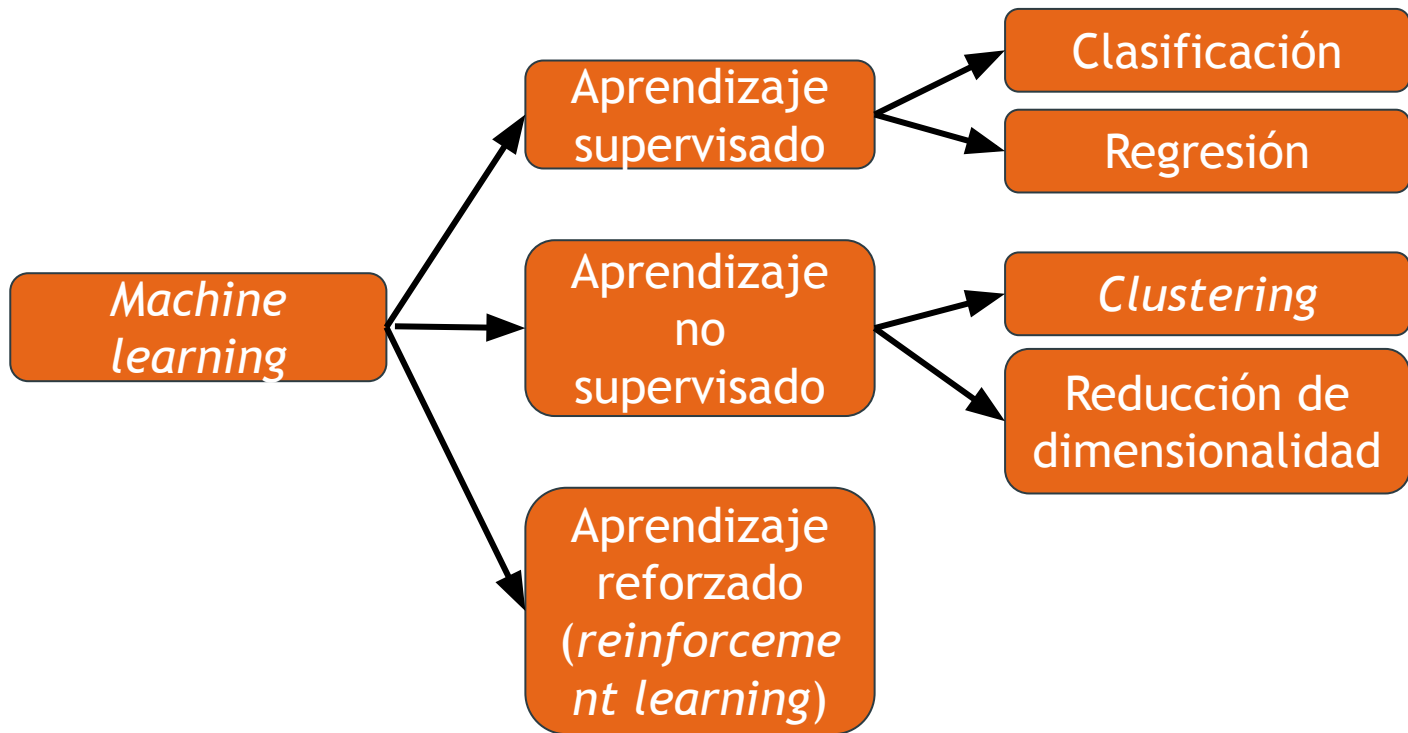
Algunos de estos **algoritmos** fueron creados hace muchos años pero por la capacidad computacional que existía en el momento de su creación **no fue posible su aplicación** como en la actualidad.

- No le decimos a un ordenador que hacer respecto a un escenario dado, sino que lo exponemos a diferentes escenarios donde el ordenador define sus parámetros y puede responder a nuevos escenarios.

Respecto a los datos con los que aprenden los algoritmos, **pueden ser estructurados o no estructurados** (texto, imagen, audio...). Por lo que es necesario un procesamiento para que el algoritmo pueda aprender de dichos datos.

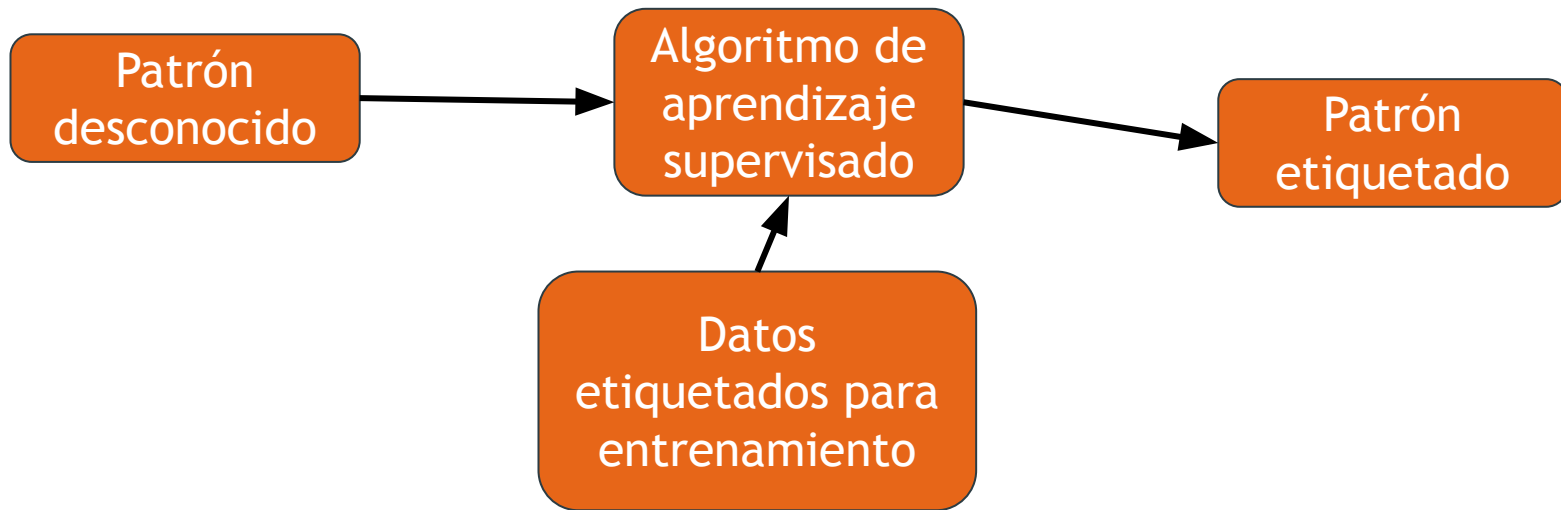


# Tipos de algoritmos de Machine Learning



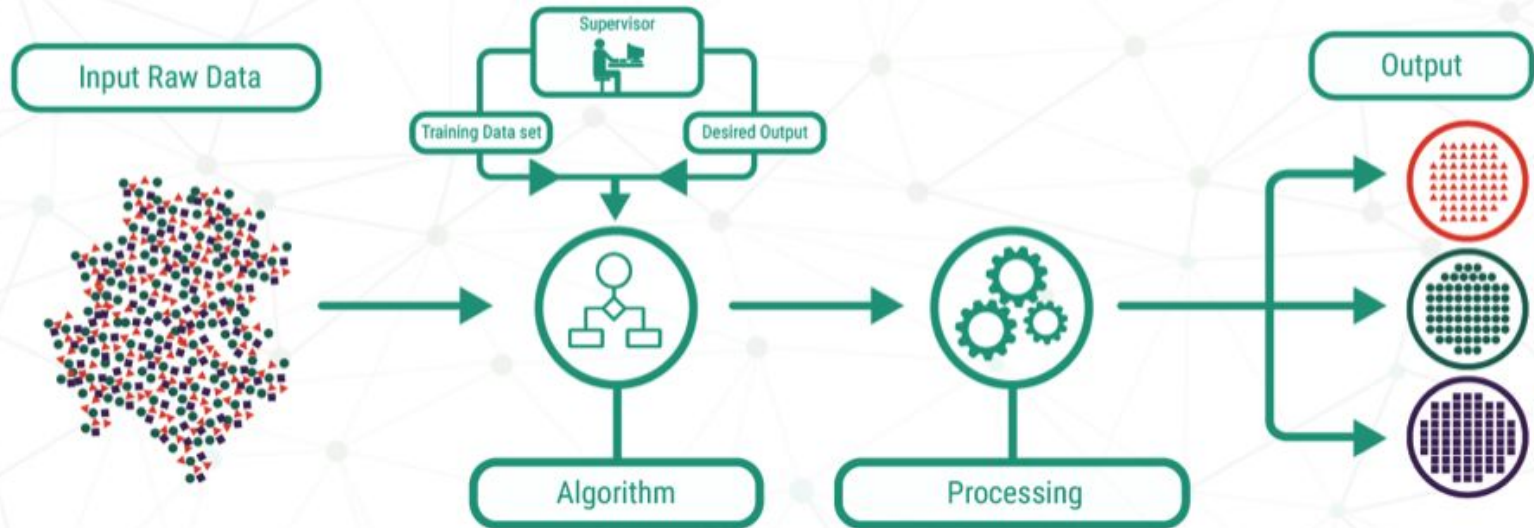
# Algoritmos supervisados

Las características de los algoritmos de aprendizaje supervisado **permiten hacer predicciones a partir de datos etiquetados**. El conjunto de datos etiquetados se utiliza para entrenar a los algoritmos y que estos aprendan.



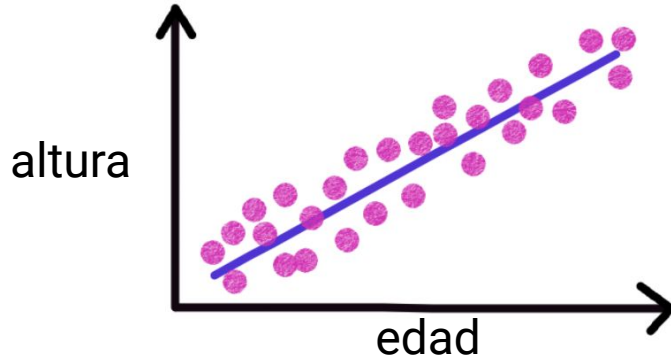
# Algoritmos supervisados

## SUPERVISED LEARNING



# Algoritmos supervisados: REGRESIÓN

La regresión se utiliza para **predecir resultados continuos**, se tienen un número de variables predictoras (explicativas) y una variable de respuesta continua (resultado o destino), se tiene que encontrar una relación entre estas variables que nos permita predecir una variable destino nueva a partir de las predictoras.

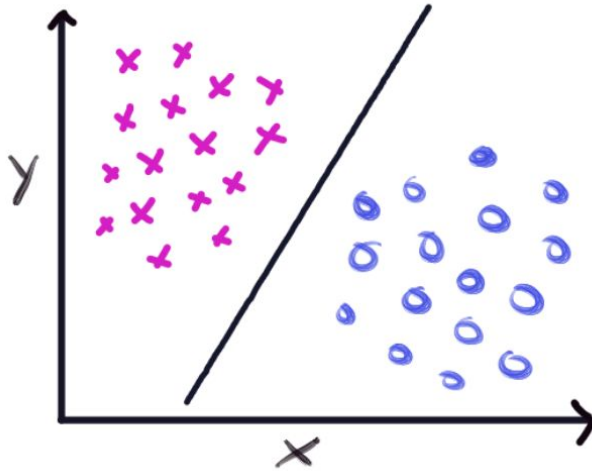


- *Variable respuesta:*  
*Cuantitativa*
- *¿Cómo medimos*  
*nuestro modelo?*

# Algoritmos supervisados: CLASIFICACIÓN

La **clasificación** tiene como objetivo predecir las etiquetas de **clase categórica** de nuevas instancias o patrones. Estas etiquetas de clase son discretas y no necesariamente debe de ser una clasificación **binaria**, puede ser **multiclase**.

✕ Clase 1  
○ Clase 2

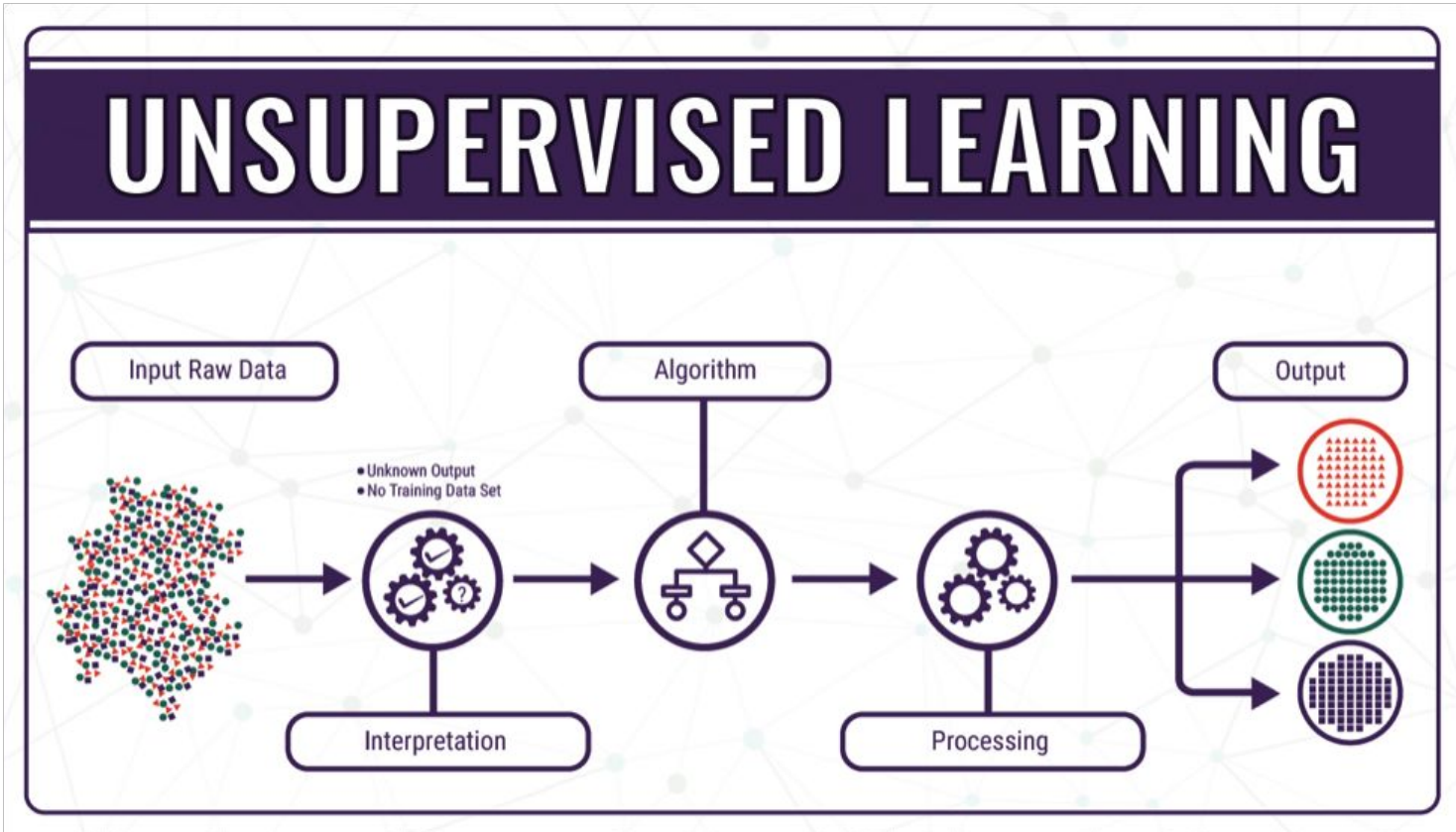


# Algoritmos NO supervisados

Los algoritmos de aprendizaje no supervisado etiquetan a patrones no etiquetados con base en sus características.

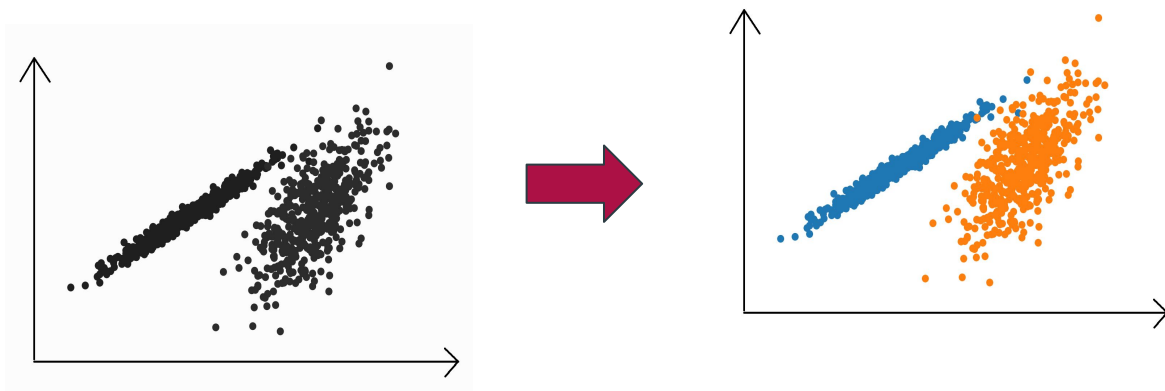


# Algoritmos NO supervisados



# Algoritmos NO supervisados: CLUSTERING

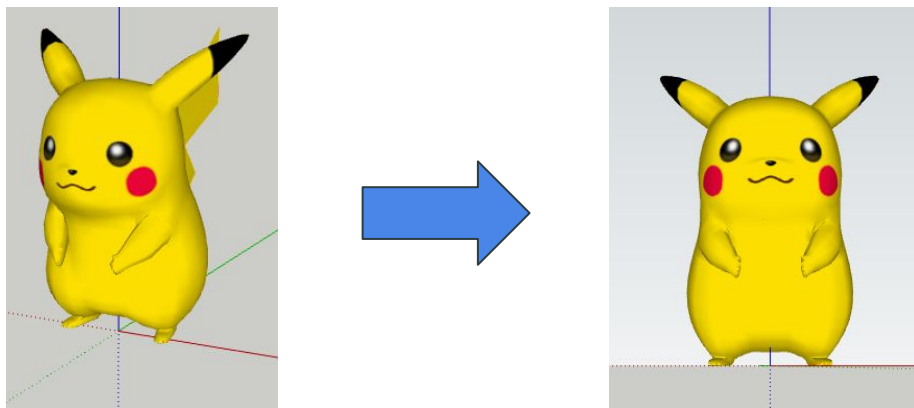
Es una técnica exploratoria de análisis de datos que nos permite **organizar conjuntos de datos no etiquetados en subgrupos significativos** (clusters) sin tener ningún conocimiento previo de los miembros del grupo. Cada cluster que surge durante el análisis define un grupo de objetos que comparten un cierto grado de semejanza y difieren de los objetos de otros clusters, razón por la cual el agrupamiento también se denomina **clasificación sin supervisión**.



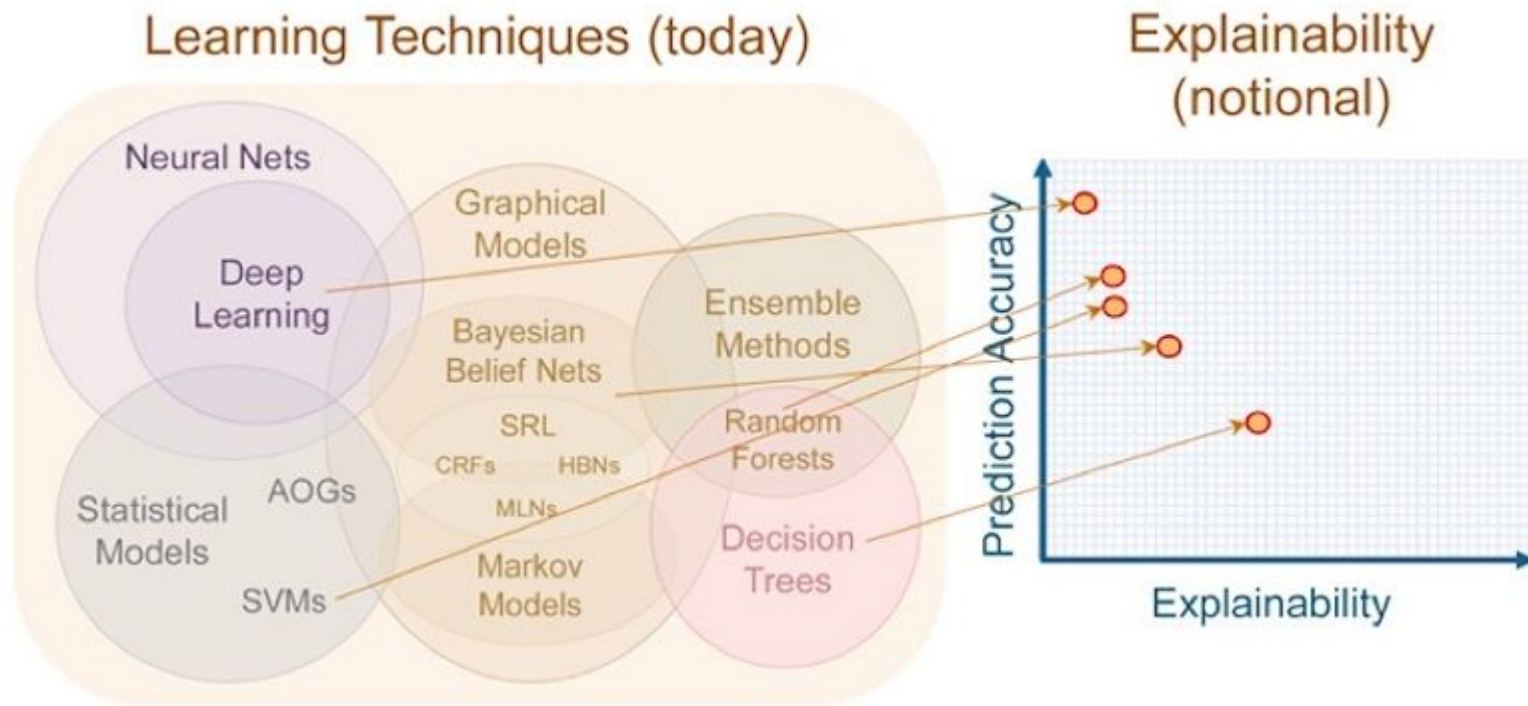


# Algoritmos NO supervisados: REDUCCIÓN DE LA DIMENSIONALIDAD

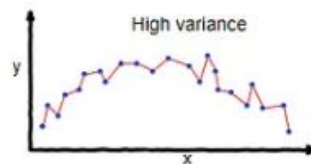
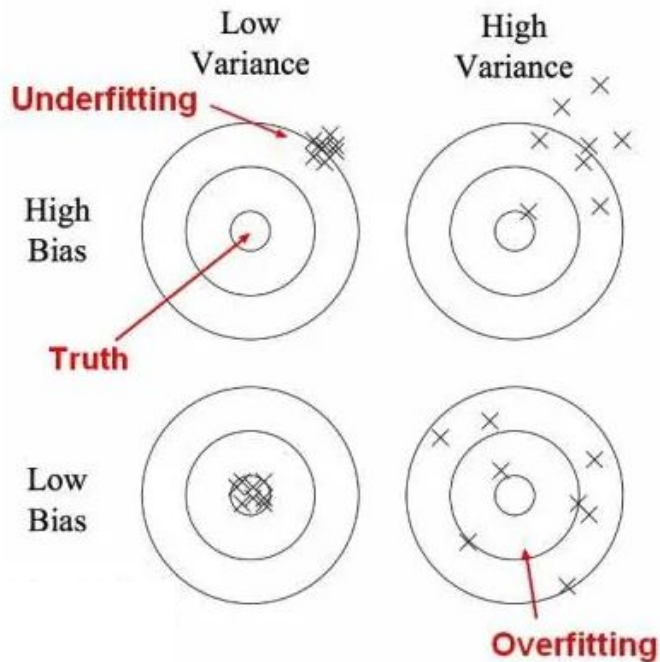
La reducción de dimensionalidad sin supervisión es un enfoque utilizado con frecuencia en el preprocesamiento de características para **eliminar ruido de los datos**; también puede degradar el rendimiento predictivo de ciertos algoritmos y comprimir los datos en un subespacio dimensional más pequeño, manteniendo la mayor parte de la información más importante.



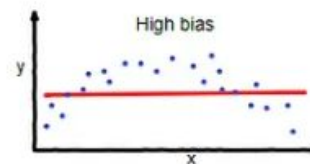
# Explicabilidad vs Precisión



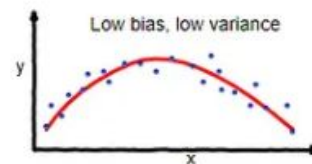
# Sesgo vs Varianza



overfitting



underfitting



Good balance


Es importante encontrar un **equilibrio** entre el **sesgo** y la **varianza** que minimice el error de nuestro modelo.

Si el modelo es simple, entonces puede tener un sesgo alto y una varianza baja.

Si el modelo tiene muchos parámetros (gpt-4), tendrá una varianza elevada y un sesgo bajo.

# Configurando nuestro entorno de trabajo

¿Cuál va a ser nuestro *stack*?

- Python (miniconda + Notebook + bibliotecas de tratamiento de datos y ML)
- VsCode / Jupyter Notebook para la interfaz
- 

Vamos a configurarlo todo!

# Preprocesamiento de DATOS

¿Qué tenemos que tener en cuenta?

- Valores perdidos (missing data)
- Outliers
- Normalización / estandarización de los datos
- Transformaciones
- Desbalanceo de clases
- Alta dimensionalidad

# Valores perdidos

- ¿Eliminar?
- ¿Imputar?
- ¿Selección / adaptación del modelo?

## Mecanismos de pérdida

Existen tres mecanismos:

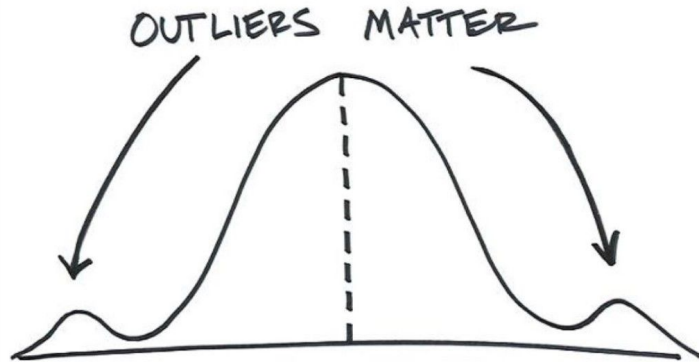
- *Missing Completely At Random (MCAR)*: la probabilidad de observar un valor ausente en una variable no depende de las otras variables ni de ella misma. Los sujetos con y sin valores ausentes tienen las mismas características.
- *Missing At Random (MAR)*: la probabilidad de observar un valor ausente depende de otras variables, no de los valores de la propia variable.
- *Missing Not At Random (MNAR)*: la probabilidad de observar un valor ausente depende de los valores de la propia variable, una vez controladas el resto de las variables. En esta situación no pueden imputarse los valores ausentes.

Es importante identificar el patrón en que aparecen los datos ausentes, ya que esto puede determinar la viabilidad de imputar y, en caso afirmativo, el método más eficiente<sup>3,5,7</sup>.

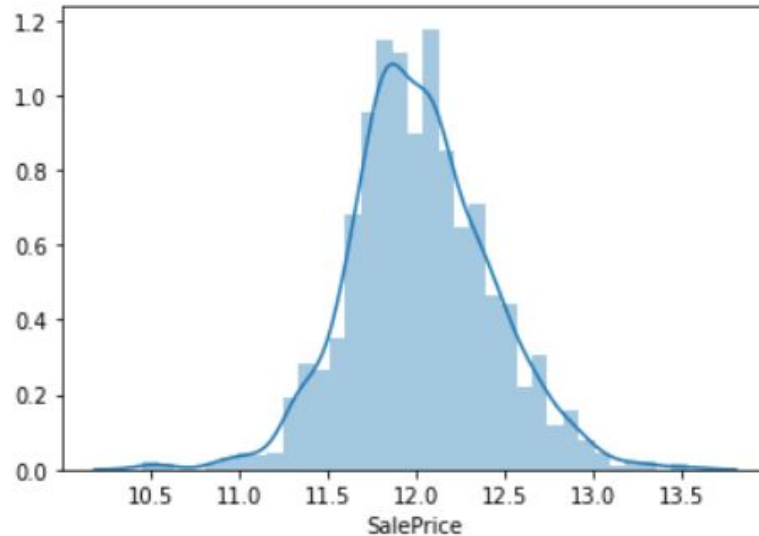
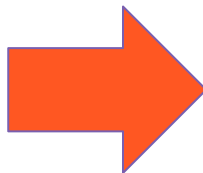
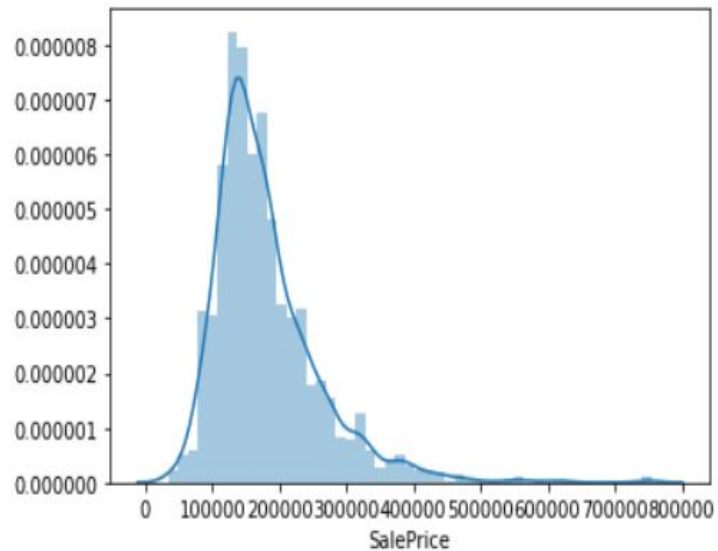
# Outliers

Los valores atípicos son datos que presentan una diferencia significativa del resto de elementos en un conjunto de datos o en una clase en particular.

- Malas mediciones al capturar los datos.
- Mal etiquetado del patrón al asignarle una clase.
- Características propias del atributo.



# Transformaciones





# Alta dimensionalidad

En ocasiones no encontramos con conjuntos de datos con gran cantidad de atributos. No necesariamente toda la información representada por los rasgos resulta relevante para la clasificación.

- Asistencia de un **experto** en el tema.
- Métodos **automáticos** para la **selección** de características.
  - Métodos óptimos: búsqueda exhaustiva, branch and bound Search.
  - Métodos subóptimos: sequential selection, stochastic search techniques (algoritmos genéticos).

# ¿Cómo validamos un modelo?

Antes de pensar en validar el modelo en sí, tenemos que decidir QUÉ datos utilizamos para su entrenamiento.

Lo habitual es coger una muestra aleatoria de nuestros datos, habitualmente entre un 70% y un 80% para entrenar el modelo.

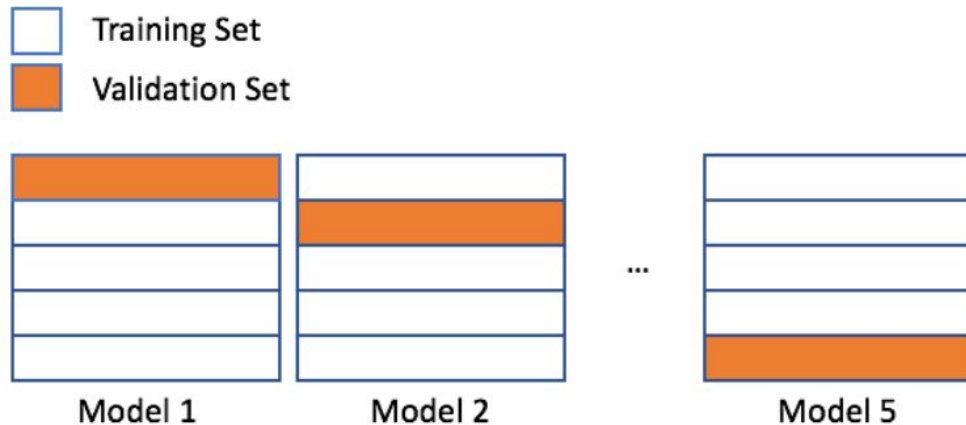
El resto de datos se utilizan para evaluar cómo de bien funciona el modelo entrenado.

Debemos asegurarnos que esta división genera muestras parecidas.



# Métodos de validación

- HoldOut
- K-FOLD Cross Validation
- LOO (Leave One Out)



# Métricas de validación

Una vez que hemos decidido cómo dividir nuestros datos, tenemos que medir.

**IMPORTANTE:** cada tipo de problema tendrá unas métricas que se adapten a la variable objetivo (target)

- Métricas de clasificación
  - Accuracy
  - Precisión
  - Recall (sensibilidad)
  - Especificidad
  - F1 Score
  - AUROC (Area Under ROC)
- Métricas de regresión
  - R2 y variantes
  - MSE (Mean Square Error), MAE (Mean Absolute Error)

*\*adapta siempre las métricas a tu problema\**

# Métricas de validación: matriz de confusión

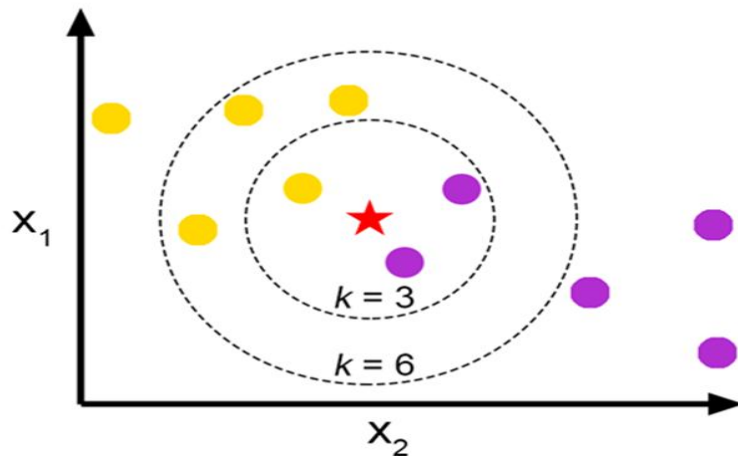
		Predicted class	
		+	-
Actual class	+	<b>TP</b> True Positives	<b>FN</b> False Negatives Type II error
	-	<b>FP</b> False Positives Type I error	<b>TN</b> True Negatives

Métrica	Fórmula	Interpretación
Accuracy	$VP + VN / \text{Total de registros}$	Acierto global
Precisión	$VP / (VP + FP)$	Acierto en la clase True
Recall o Sensibilidad	$VP / (VP + FN)$	Cobertura la clase True
Especificidad	$VN / (VN + FP)$	Cobertura de la clase False
F1 score	$2*VP / (2*VP + FP + FN)$	Métrica útil en desbalance de clases

# KNN

## Pasos en el proceso de entrenamiento de KNN

1. Calcular la distancia entre ese patrón y todos los patrones del conjunto de entrenamiento.
2. Se seleccionan los  $k$  patrones cuyas distancias sean las menores.
3. Verificar cual es la clase más frecuente entre los  $k$  patrones seleccionados y asignar dicha clase.



# Clasificador Naïve Bayes

Consideremos una fábrica de tapones de corcho, cuya producción está restringida a dos clases en la calidad de los corchos: promedio y superior.

Cantidad de corchos de la clase 1 ( $c_1$ ):  $n_1 = 901,420$

Cantidad de corchos de la clase 2 ( $c_2$ ):  $n_2 = 1,352,130$

Cantidad de corchos totales:  $n = 2,253,550$

Con esta información se puede obtener las probabilidades a priori:

$$P(c_1) = n_1/n = 0.4$$

$$P(c_2) = n_2/n = 0.6$$

# Clasificador Naïve Bayes

Supongamos que nos piden adivinar a qué clase pertenece un corcho sin haberlo visto y la única información disponible es la probabilidad a priori.

La opción lógica es decir que el corcho pertenece a la clase 2 ya que con esta decisión esperaríamos estar equivocados solo un 40%.

Supongamos que ahora se nos permite conocer los valores del vector  $\mathbf{x}$  de características del corcho en cuestión. Sea  $P(c_k | \mathbf{x})$  la probabilidad condicional de que el corcho representado por  $\mathbf{x}$  pertenezca a la clase  $c_k$ .

Si somos capaces de estimar  $P(c_1 | \mathbf{x})$  y  $P(c_2 | \mathbf{x})$  podríamos determinar una frontera de decisión con la siguiente regla:

$$\text{Si } P(c_1 | \mathbf{x}) > P(c_2 | \mathbf{x}) \text{ entonces } \mathbf{x} \in c_1 \text{ sino } \mathbf{x} \in c_2$$



# Clasificador Naïve Bayes

$$P(c_k/\mathbf{x}) = \frac{P(c_k) \cdot P(\mathbf{x}/c_k)}{P(\mathbf{x})} \quad \text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

De esta ecuación lo importante es el numerador, ya que el denominador no depende de las clases y los valores de  $\mathbf{x}$  son conocidos, por lo que se le considera una constante.

El numerador es equivalente a la distribución conjunta de probabilidades:

$$P(c_k, x_1, \dots, x_n)$$

Que puede ser re-escrita:

$$\begin{aligned} P(c_k, x_1, \dots, x_n) &= P(c_k) P(x_1, \dots, x_n | c_k) \\ &= P(c_k) P(x_1 | c_k) P(x_2, \dots, x_n | c_k, x_1) \\ &= P(c_k) P(x_1 | c_k) P(x_2 | c_k, x_1) P(x_3, \dots, x_n | c_k, x_1, x_2) \\ &= P(c_k) P(x_1 | c_k) P(x_2 | c_k, x_1) P(x_3 | c_k, x_1, x_2) P(x_4, \dots, x_n | c_k, x_1, x_2, x_3) \end{aligned}$$

# Clasificador Naïve Bayes

Usando la asunción ingenua de la independencia condicional, cada atributo  $x_i$  es condicionalmente independiente de cualquier otro atributo  $x_j$  para  $i \neq j$ , lo que significa:

$$P(x_i | c_k, x_j) = P(x_i | c_k)$$

Por tanto la probabilidad conjunta puede ser escrita:

$$P(c_k, x_1, \dots, x_n) = P(c_k) P(x_1 | c_k) P(x_2 | c_k) P(x_3 | c_k) \dots P(x_n | c_k)$$

$$= P(c_k) \prod_{i=1}^n P(x_i | c_k)$$

# Clasificador Naïve Bayes

Consideremos un conjunto de entrenamiento como sigue:

Número total de instancias	100
Número de instancias en la clase 1	40
Número de instancias en la clase 2	30
Número de instancias en la clase 3	30

Por tanto:

Probabilidad a priori de la clase 1	$\frac{40}{100} = 0.4$
-------------------------------------	------------------------

Probabilidad a priori de la clase 2	$\frac{30}{100} = 0.3$
-------------------------------------	------------------------

Probabilidad a priori de la clase 3	$\frac{30}{100} = 0.3$
-------------------------------------	------------------------

# Clasificador Naïve Bayes

Si de las 40 instancias en la clase 1, un atributo binario toma el valor de 0 en 30 instancias y el valor de 1 en las otras 10 instancias, la probabilidad a priori de que un rasgo tome el valor de 0 es:

$$\frac{30}{40} = 0.75$$

# Clasificador Naïve Bayes

Ejemplo: consideremos el siguiente conjunto de entrenamiento

Example training data set

Cook	Mood	Cuisine	Tasty
Sita	Bad	Indian	Yes
Sita	Good	Continental	Yes
Asha	Bad	Indian	No
Asha	Good	Indian	Yes
Usha	Bad	Indian	Yes
Usha	Bad	Continental	No
Asha	Bad	Continental	No
Asha	Good	Continental	Yes
Usha	Good	Indian	Yes
Usha	Good	Continental	No

Consideremos un nuevo patrón a clasificar:

(cook=sita, mood=bad, cuisine=continental)

# Clasificador Naïve Bayes

Probabilidad a priori de las clases:

Probabilidad a priori de Tasty = yes  $\rightarrow P(\text{Tasty}=\text{yes}) = \frac{6}{10} = 0.6$

Probabilidad a priori de Tasty = no  $\rightarrow P(\text{Tasty}=\text{no}) = \frac{4}{10} = 0.4$

Probabilidad a priori de los atributos:

$$P(\text{Cook}=\text{sita}|\text{Tasty}=\text{yes}) = \frac{2}{6} = 0.33$$

$$P(\text{Cook}=\text{sita}|\text{Tasty}=\text{no}) = 0 = 0.01$$

$$P(\text{Mood}=\text{bad}|\text{Tasty}=\text{yes}) = \frac{2}{6} = 0.33$$

$$P(\text{Mood}=\text{bad}|\text{Tasty}=\text{no}) = \frac{3}{4} = 0.75$$

- Se asigna un valor pequeño para no anular todo el cálculo de la probabilidad.

# Clasificador Naïve Bayes

**Probabilidad a priori de los atributos:**

$$P(\text{Cuisine}=\text{continental}|\text{Tasty}=\text{yes}) = \frac{2}{6} = 0.33$$

$$P(\text{Cuisine}=\text{continental}|\text{Tasty}=\text{no}) = \frac{3}{4} = 0.75$$

**Por tanto las probabilidades a posteriori son:**

$$P(\text{Tasty}=\text{yes}|X) = 0.6 \times 0.33 \times 0.33 \times 0.33 = 0.0216$$

$$P(\text{Tasty}=\text{no}|X) = 0.4 \times 0.01 \times 0.75 \times 0.75 = 0.00225$$

El nuevo patrón es clasificado como perteneciente a la clase Tasty = yes

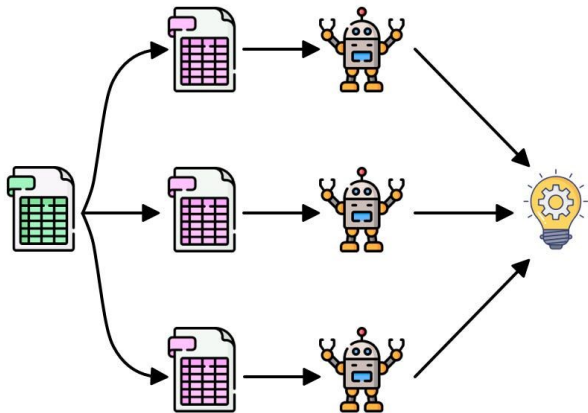
# Otros Modelos

- Support Vector Machine (muy interesantes con alta dimensionalidad)
- Árboles de decisión (fácilmente explicables y visuales)
- Ensamblado de modelos (bagging y boosting)
- ...



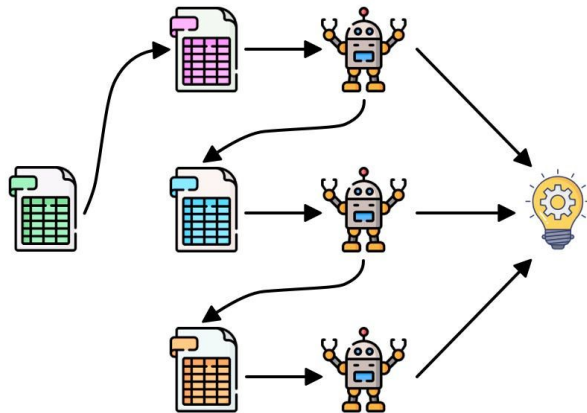
# Ensamblado de Modelos

Bagging



Parallel

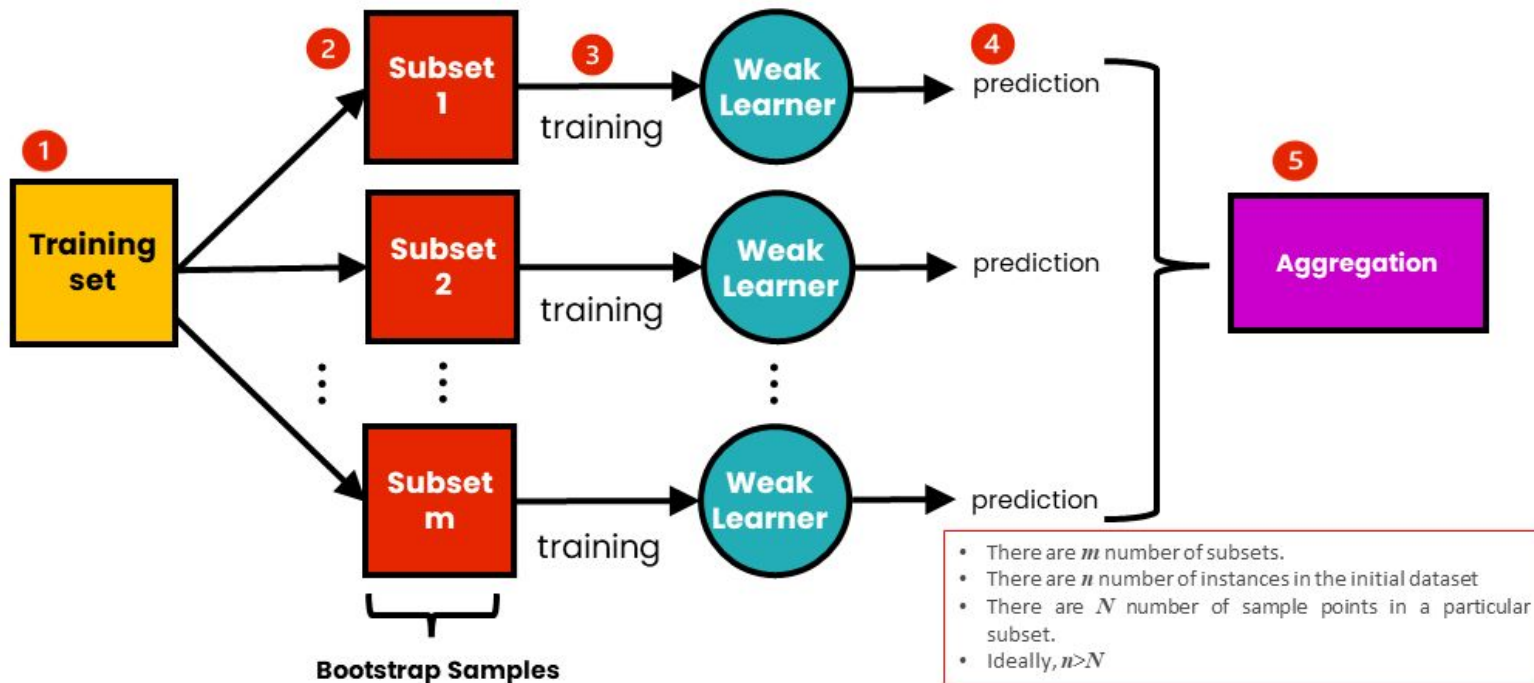
Boosting



Sequential

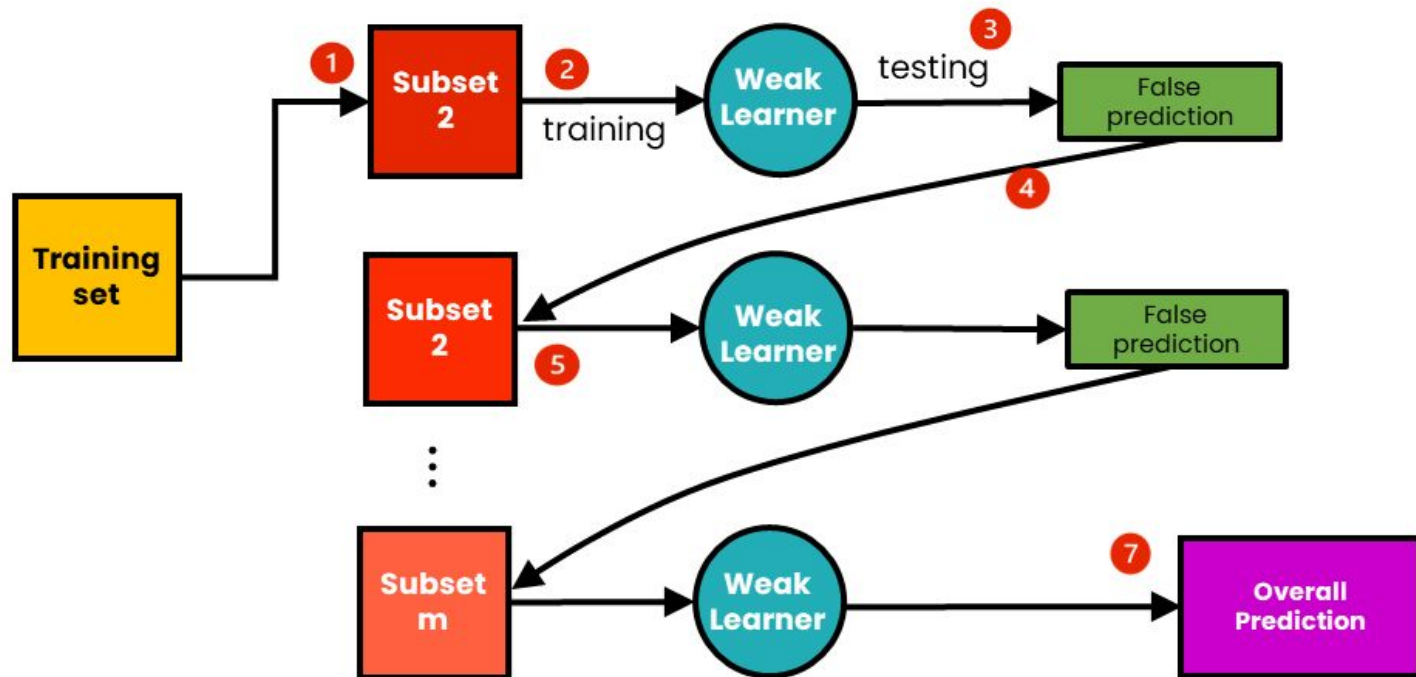
# Ensamblado de Modelos

## The Process of Bagging (Bootstrap Aggregation)



# Ensamblado de Modelos

## The Process of Boosting

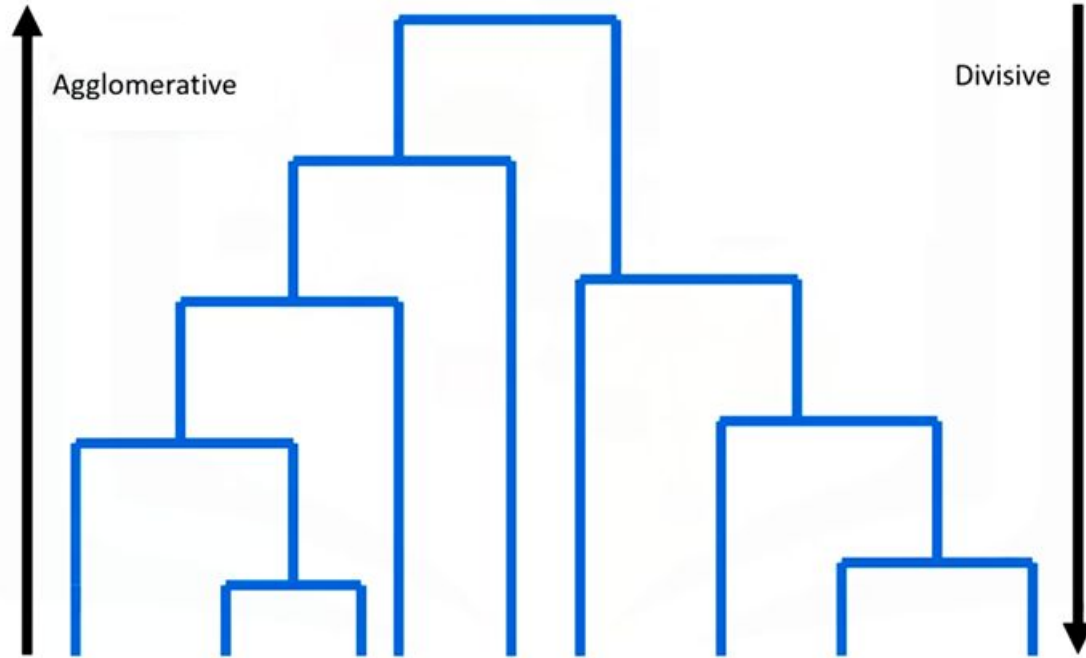


# Clustering

Técnica que tiene como objetivo **organizar patrones en grupos**, de modo que los patrones que pertenecen al mismo grupo son lo suficientemente similares como para inferir que son del mismo tipo y los patrones que pertenecen a diferentes grupos son lo suficientemente diferentes como para inferir que son de otra clase.

Los grupos pueden ser exclusivos, con traslapes, probabilísticos, jerárquicos.

# Clustering Jerárquico



# Clustering Jerárquico: Aglomerativo

	TO	OT	VA	MO	WI	ED
TO		351	3363	505	1510	2699
OT			3543	167	1676	2840
VA				3690	1867	819
MO					1824	2976
WI						1195
ED						



	TO	OT	VA	MO	WI	ED
TO		351	3363	505	1510	2699
OT			3543	167	1676	2840
VA				3690	1867	819
MO					1824	2976
WI						1195
ED						

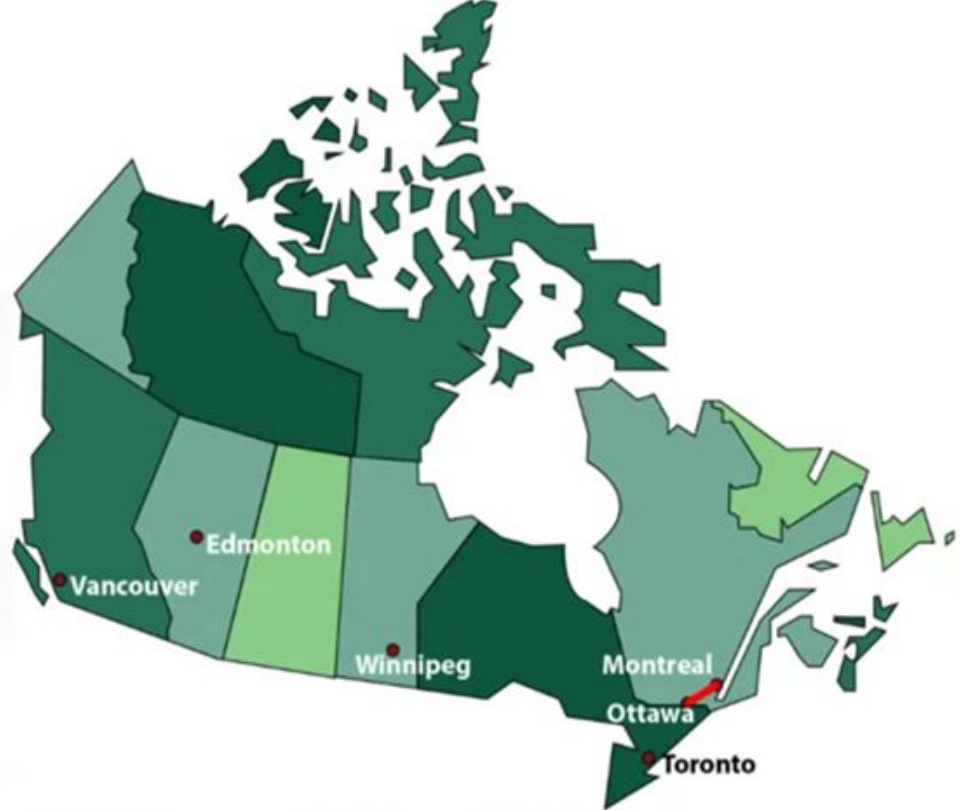




TO OT MO VA ED WI

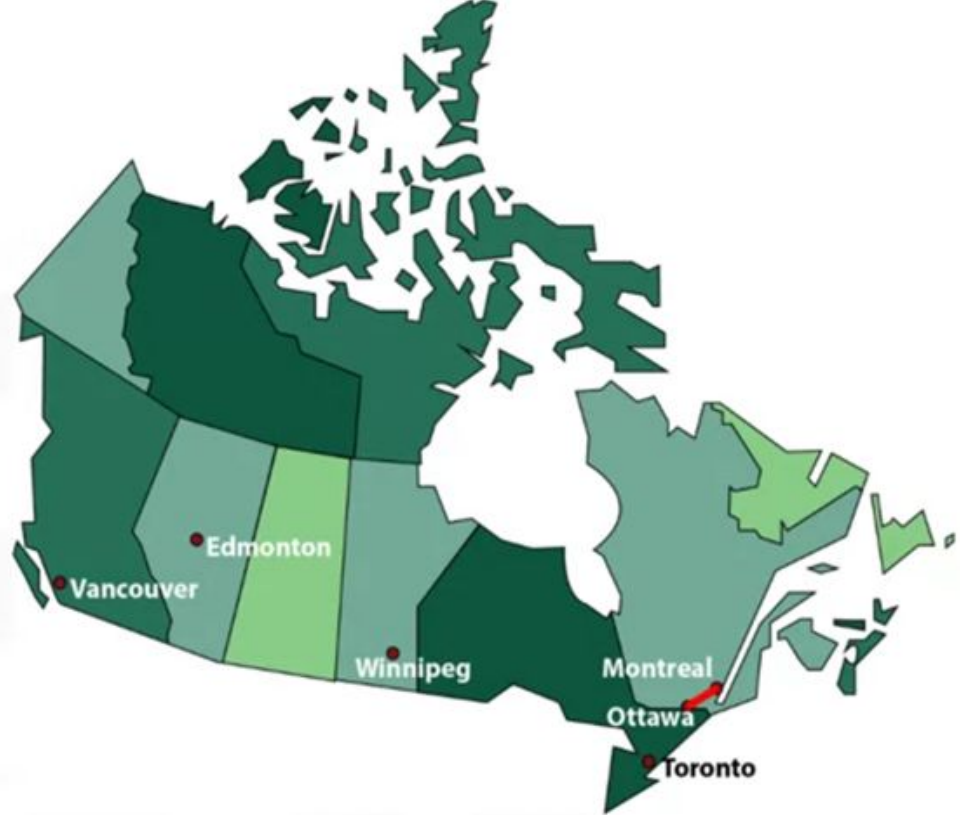
	TO	OT/MO	VA	WI	ED
TO		351	3363	1510	2699
OT/MO			3543	1676	2840
VA				1867	819
WI					1195
ED					





TO OT MO VA ED WI

	TO	OT/MO	VA	WI	ED
TO		351	3363	1510	2699
OT/MO			3543	1676	2840
VA				1867	819
WI					1195
ED					



	TO	OT/MO	VA	WI	ED
TO		351	3363	1510	2699
OT/MO			3543	1676	2840
VA				1867	819
WI					1195
ED					



	TO/OT/MO	VA	WI	ED
TO/OT/MO		3543	1676	2840
VA			1867	819
WI				1195
ED				





	TO/OT/MO	VA	WI	ED
TO/OT/MO		3543	1676	2840
VA			1867	819
WI				1195
ED				



	TO/OT/MO	VA/ED	WI
TO/OT/MO		2840	1676
VA/ED			1667
WI			

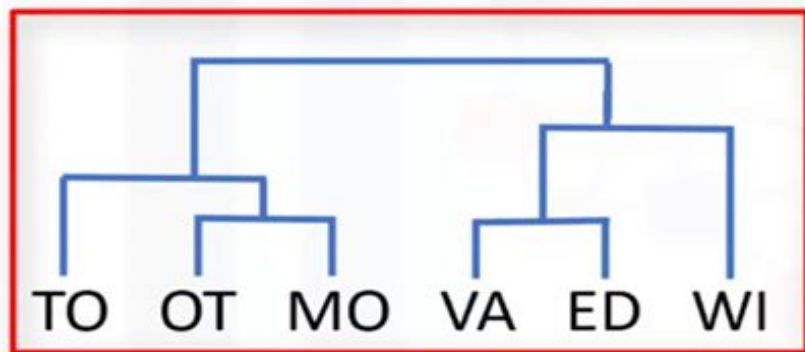




	TO/OT/MO	VA/ED	WI
TO/OT/MO		2840	1676
VA/ED			1667
WI			



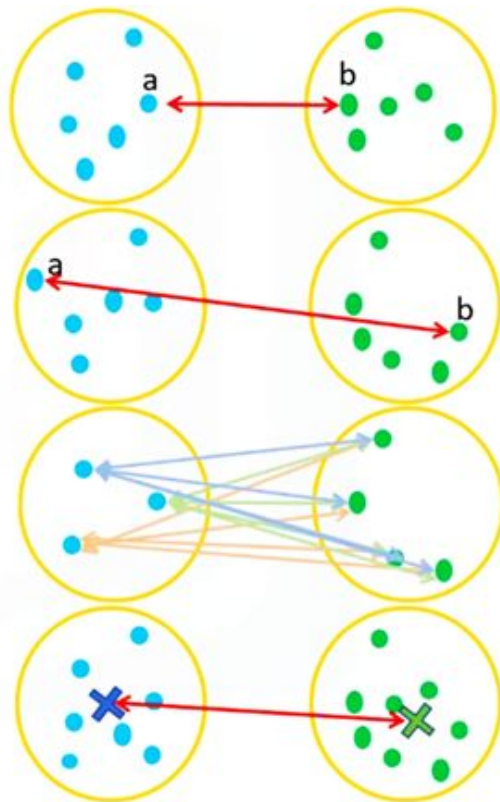




	TO/OT/MO	VA/ED/WI
TO/OT/MO		1676
VA/ED/WI		

# Maneras de calcular distancias

- **Single-Linkage** Clustering
  - La distancia mínima entre clusters
- **Complete-Linkage** Clustering
  - Máxima distancia entre clusters
- **Average Linkage** Clustering
  - Distancia promedio entre clusters
- **Centroid Linkage** Clustering
  - Distancia entre el centroide de cada cluster





# K-means

1. Elegir  $k$  centroides, y posicionarlos en el conjunto de datos en un lugar aleatorio.
2. Calcular la distancia entre cada patrón desde los centroides.
3. Asignar a cada patrón al centroide más cercano, la clase del centroide.
4. Una vez que todos los puntos o patrones fueron asignados, recalcular la posición de los centroides
5. Repetir los pasos 2 al 4 hasta que los centroides se mantengan en la misma posición, llegar a un número máx. de interacciones o aceptar una tolerancia definida.

# K-Means vs Jerárquico

K-Means	Hierarchical clustering
Más eficiente	Es lento para dataset con una gran cantidad de datos
Requiere de especificar el número de clusters	No requiere de especificar el número de clusters
Solo genera las particiones de los datos a través del número de clústeres definidos	Puede dividir el dataset un múltiples particiones
Puede generar diferentes resultados	Siempre generará el mismo resultado

# Otros algoritmos

- DBSCAN y HDBSCAN (probabilísticos, muy útiles con presencia de outliers)
- Gaussian Mixtures (mezcla de gaussianas)
- K-Medians, K-Modes, K-Prototypes...
- SOM (Redes Neuronales)
- ...

# ¿Nos ponemos a prueba?



Caso práctico con seguimiento