# What is AI?

IBM Research defines Artificial Intelligence (AI) as Augmented Intelligence, helping experts scale their capabilities as machines do the time-consuming work.

AI learns by creating machine learning models based on provided inputs and desired outputs.

AI should not attempt to replace human experts.

Exists three types of AI:

* **Weak or narrow AI:** It’s applied to a specific domain, they cannot learn new tasks, making decision based on programmed algorithms or training data. They only can do a specific task. Examples: language translators, recommendation engines…
* **Strong or Generalized AI**: This AI can operate with independent/unrelated tasks. It can learn new tasks to solve new problems, can perform at a human level intelligence.
* **Super IA or Conscious IA:** Is AI with human-level consciousness, it’s the future of IA.

AI-powered advances in speech-to-text technology have made real time transcription a reality.

# What is Cognitive technology?

**Cognitive computing systems** differ from conventional computing systems in that they can:

* Read and interpret unstructured data, understanding not just the meaning of words but also the intent and context in which they are used.
* Reason about problems in a way that humans reason and make decisions.
* Learn over time from their interactions with humans and keep getting smarter

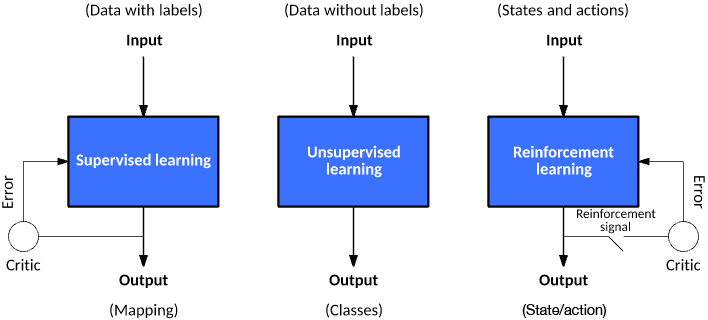
First, **cognitive technology** is often used to **enable innovation and discovery** by understanding new patterns, insights[[1]](#endnote-1) and opportunities. Second, it is often used to **optimize operations** to provide better awareness, continuous learning, better forecasting and optimization. Third, to **augment and scale expertise** by capturing and sharing the collective knowledge of the organization. Finally, to **create adaptive, personalized experiences,** including individualized products and services, to better engage customers and meet their needs.

# Terminology and related concepts

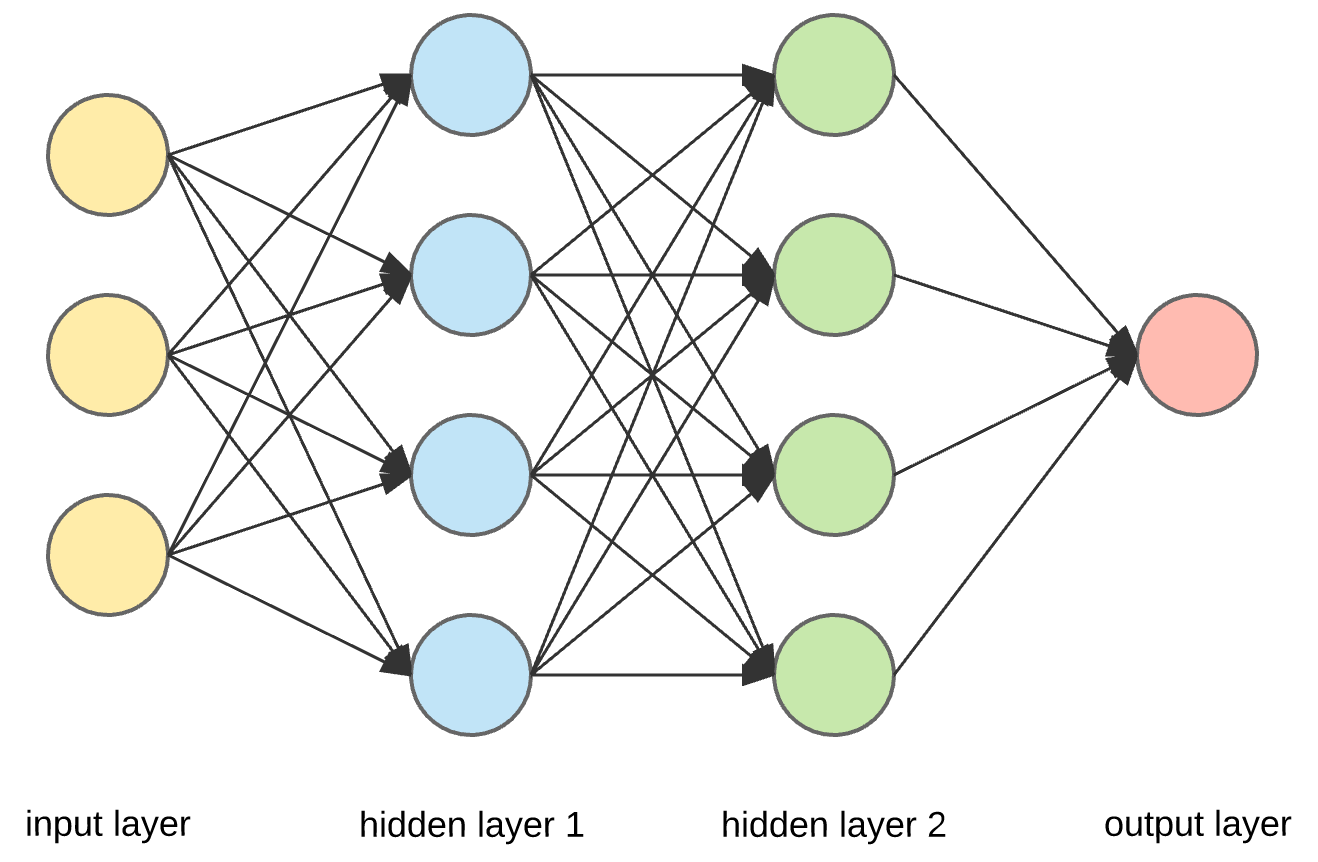
**Artificial intelligence** is a branch of computer science dealing with a simulation of intelligent behavior. AI systems will typically demonstrate behaviors associated with **human intelligence** such as planning, learning, reasoning, problem-solving, knowledge representation, perception, motion, and manipulation.

**Machine learning** is what enables machines to solve problems on their own and make accurate predictions using the provided data.

* **Supervised learning**: Is trained on human-labeled data. You provide input and labels.
* **Unsupervised learning**: Giving unlabeled data and letting it find patterns by itself. You provide input.
* **Reinforcement learning**: Providing with a set of rules and constraints and letting it learn how to achieve its goals. Uses a reward function to penalize bad actions or reward good actions.



**Deep learning** is a specialized subset[[2]](#endnote-2) of Machine Learning that uses layered neural networks to simulate human decision-making. Deep learning algorithms can label and categorize information and identify patterns. It is what enables AI systems to continuously learn on the job, and improve the quality and accuracy of results by determining whether decisions were correct

A **neural network** in AI is a collection of small computing units called neurons that take incoming data and learn to make decisions over time. Neural networks are often layered deep and are the reason deep learning algorithms become more efficient as the datasets increase in volume, as opposed to other machine learning algorithms that may plateau as data increases.

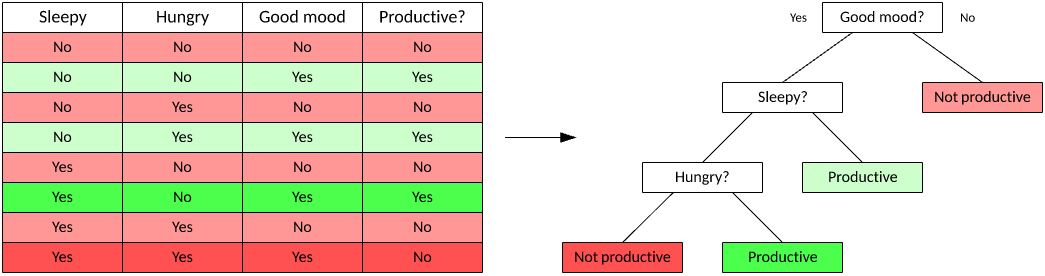
**Neural networks** are trained from examples rather than being explicitly programmed**. Neural networks** learn through a process called **backpropagation**. **Backpropagation** uses a set of training data that match known inputs to desired outputs. First, the inputs are plugged into the network and outputs are determined. Then, an error function determines how far the given output is from the desired output. Finally, adjustments are made in order to reduce errors.

Types of **neural networks**:

* **Perceptrons** are the simplest and oldest types of neural networks. They are single-layered neural networks consisting of input nodes connected directly to an output node
* **Convolutional neural networks** or **CNNs** are multilayer neural networks. **CNNs** are useful in applications such as image processing, video recognition, and natural language processing. A **convolution** is a mathematical operation, where a function is applied to another function and the result is a mixture of the two functions. Convolutions are good at detecting simple structures in an image, and putting those simple features together to construct more complex features.
* **Recurrent neural networks** or **RNNs**, are recurrent because they perform the same task for every element of a sequence, with prior outputs feeding subsequent stage inputs.

## Decision trees

A decision tree is a supervised learning method for classification. Create trees that predict the result of an input vector based on decision rules.



Two types of models exist for decision trees: **classification trees**, where the target variable is a discrete value and the leaves represent class labels (as shown in the example), and **regression trees(search it)**, where the target variable can take continuous values. You use a data set to train the tree, which then builds a model from the data. You can then use the tree for decision-making with unseen data

## Types of dataset in machine learning

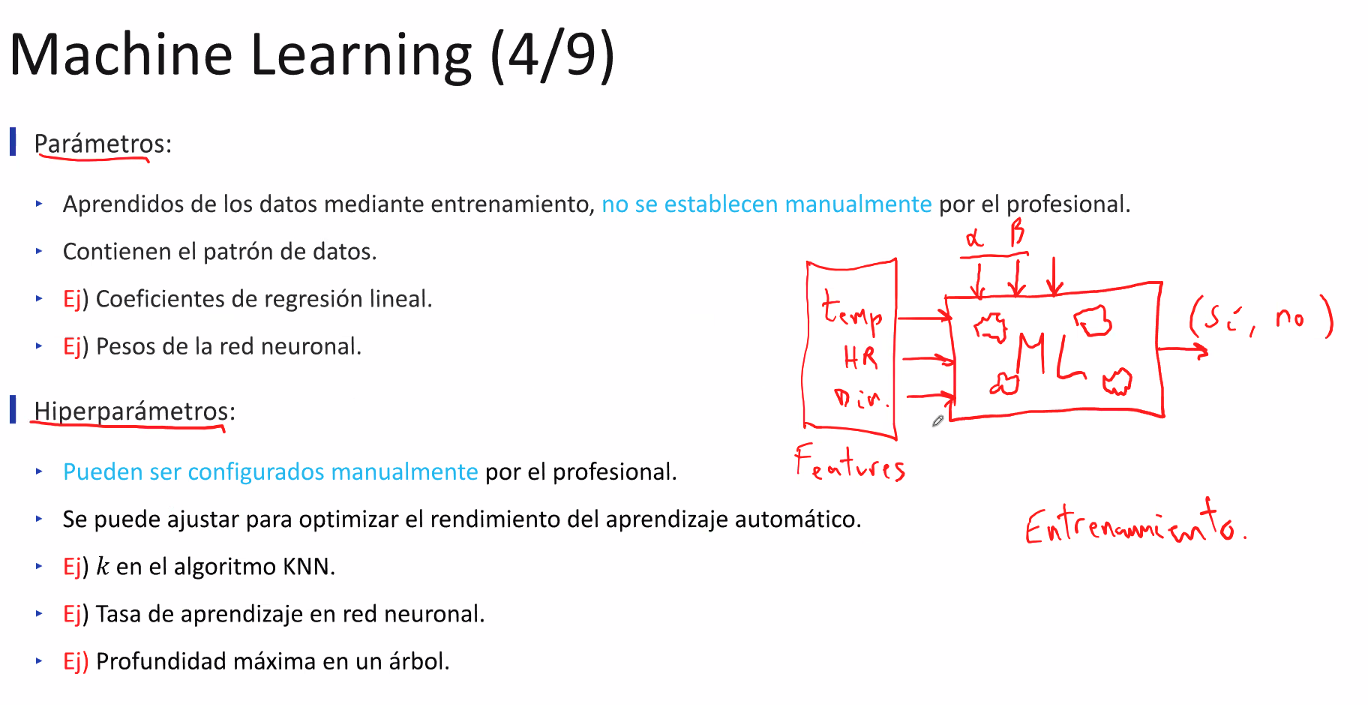
* The **Training subset** is the data used to train the algorithm.
* The **Validation subset** is used to validate our results and fine-tune the algorithm's parameters.
* The **Testing data** is the data the model has never seen before and used to evaluate how good our model is. We can then indicate how good the model is using terms like, accuracy, precision and recall.

## Application areas

* **Natural Language Processing** (NLP) is a subset of **artificial intelligence** that enables computers to understand the meaning of human language, including the intent and context of use.
* **Speech-to-text** enables machines to convert speech to text by identifying **common patterns** in the different pronunciations of a word, mapping new voice samples to corresponding words.
* **Speech Synthesis** enables machines to create natural sounding voice models.
* **Computer Vision** enables machines to identify and differentiate objects in images the same way humans do.

# Machine learning

## Aprendizaje supervisado



En la salida, si es una variable discreta (es decir que se pueden contar los resultados posibles) se llama **Clasificación,** si se trata de una variable real es decir continua se llama **Regresión.**

Dentro de la caja se consideran parámetros.

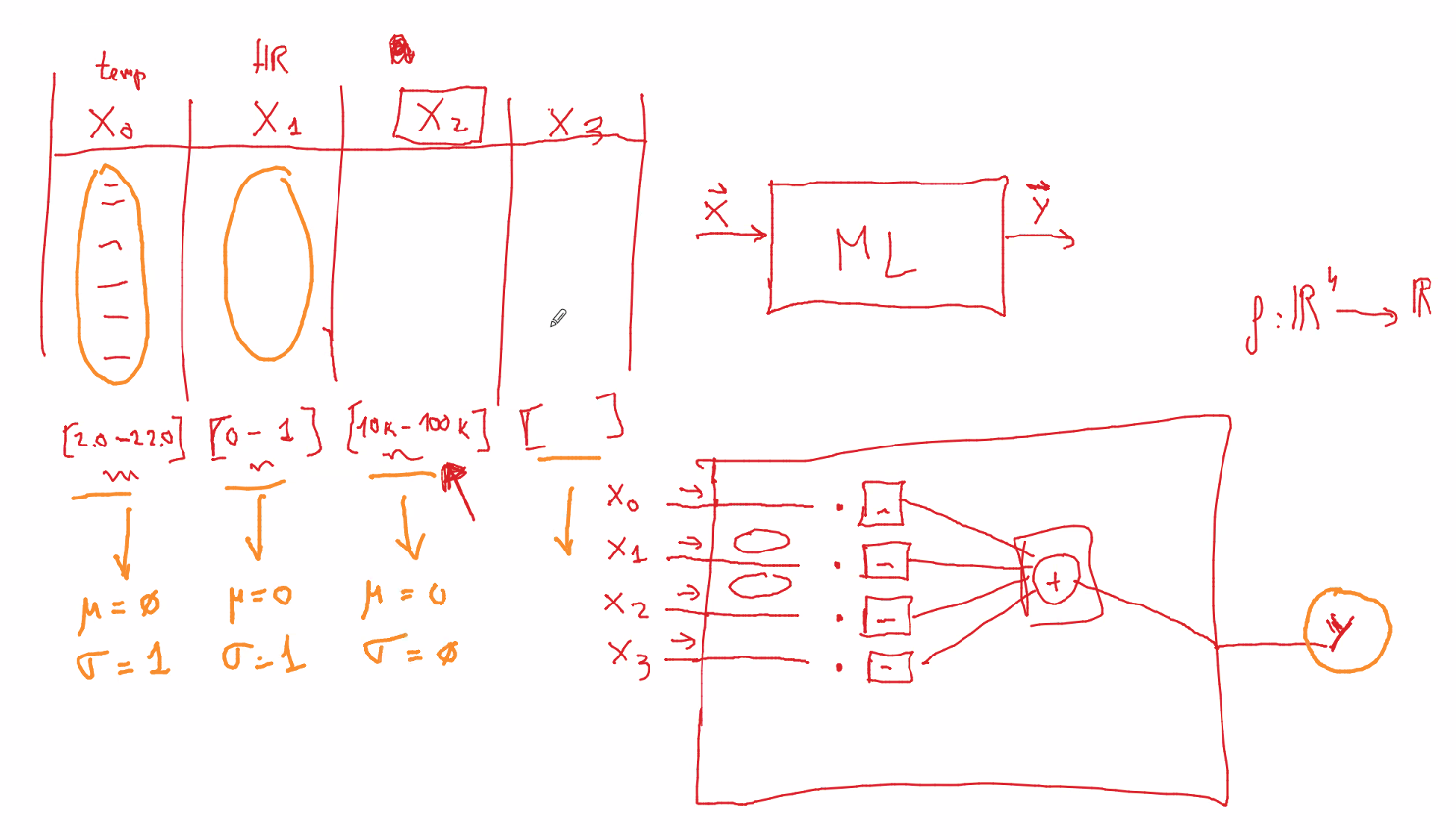
Las features son los valores constantes.

### Flujo de aprendizaje

La **Ingeniería de características,** es la fase donde creamos nuestros propios datos.

### Regresión lineal con variables continuas

El problema sería cuando los rangos son muy diferentes, en el caso de tener variables continuas.



Entonces sería necesario realizar un estándar o un escalado de las mismas. Solo se puede realizar una **estandarización** si las variables se distribuyen en una normal o si tenemos un dataset gigante, debido al **teorema central del límite**.

Para detectar si nuestras variables son **normales** debemos usar el test de Kolmogorov-Smirnov.

Si no cumplimos estas características para realizar la estandarización, se puede realizar un reescalado, es decir una regla de tres (**minmaxscaler**)

Mientras estén en el mismo **orden de magnitud** funcionará, teniendo en cuenta que por ejemplo podemos poner la variable1 y la 2 en estandarización y las demás en minmaxscaler.

Los problemas que tiene el minmaxscaler, es que, si nuestros datos no son lineales, podemos tener un mínimo con mucha desviación típica de los datos centrales y con los máximos iguales, para eso tenemos el **robustscaler**.

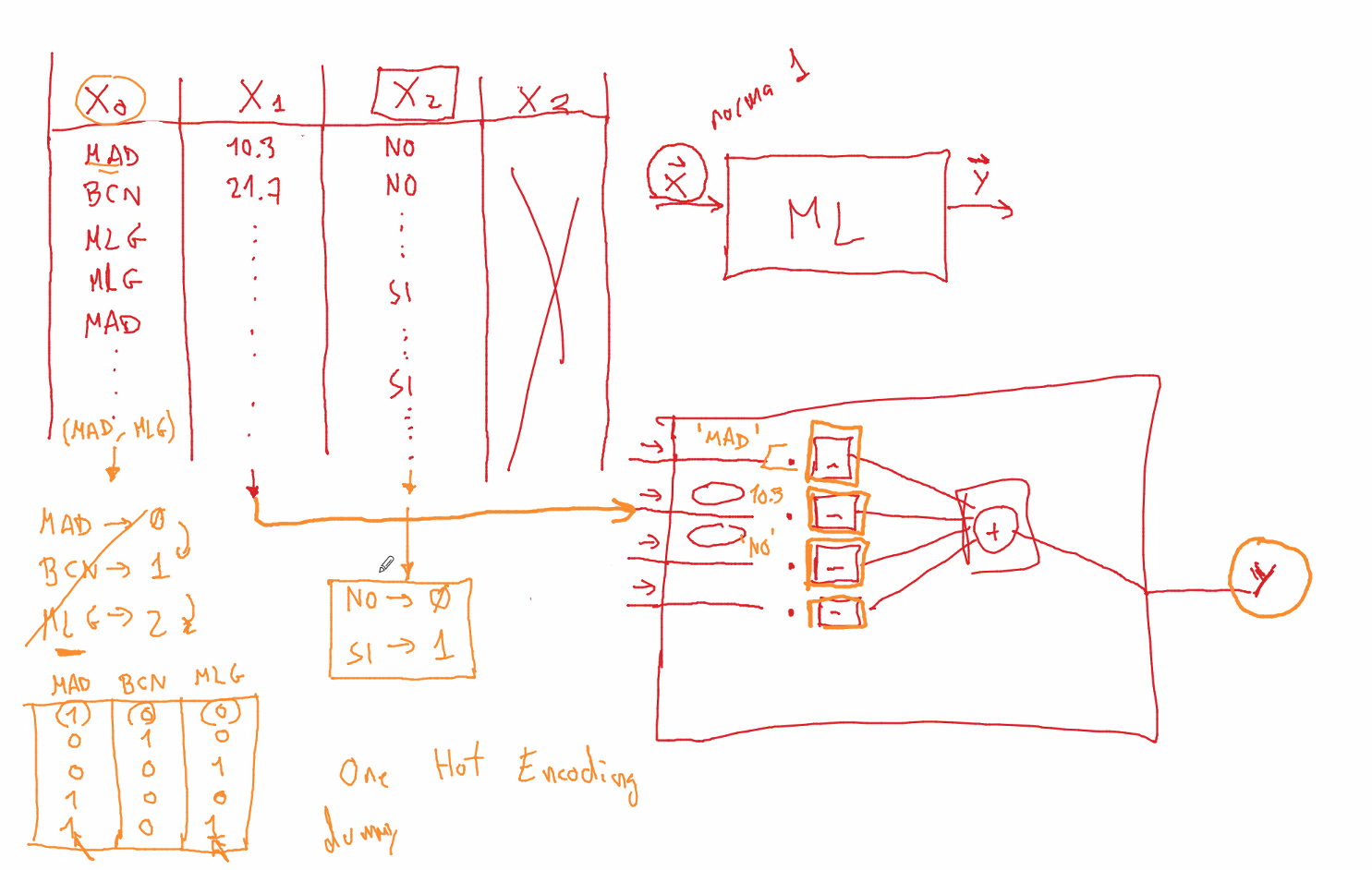
### Discretización

Debe ser binaria, de un valor pasamos a dos.

Un ejemplo es la edad, que puede ser una variable continua (es decir 37.5) y pasarla a por ejemplo adolescente, joven adulto, adulto…

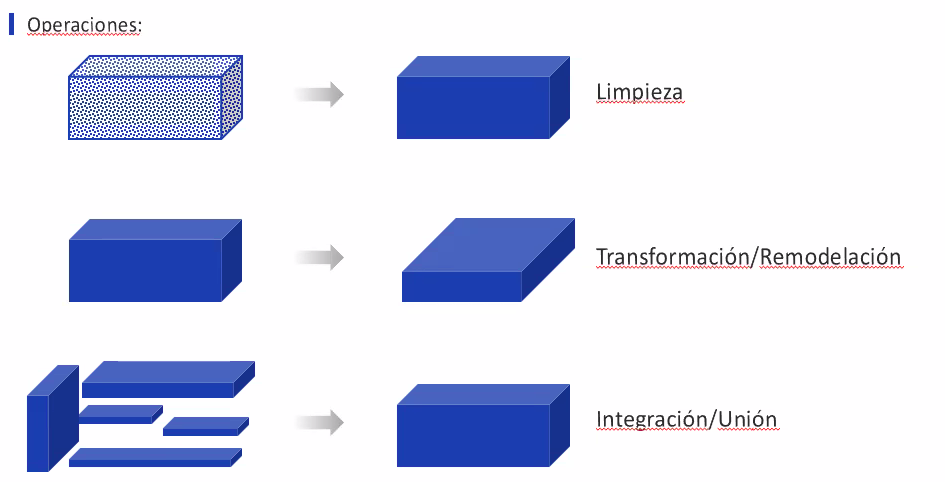
Es decir, clasificamos los datos en ciertas labels.

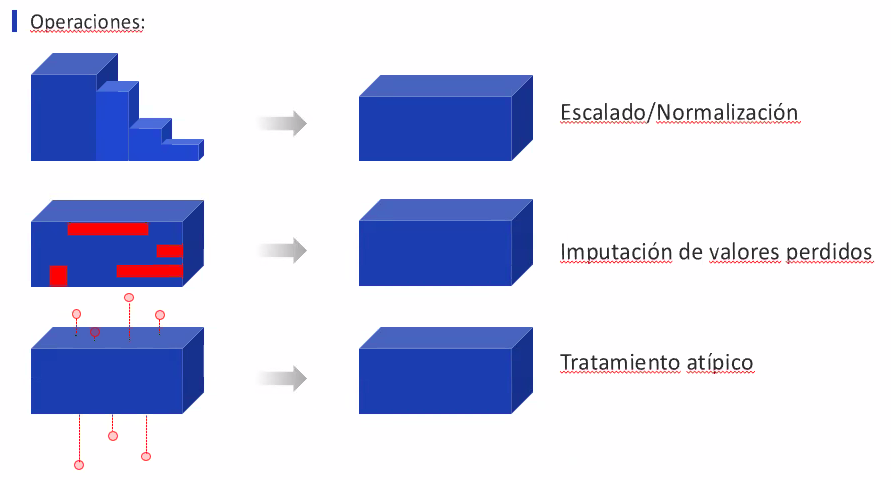
### Regresión lineal con variables discretas



## Procesamiento de datos

Se divide en distintas fases:





La imputación de valores perdidios, tenemos tres formas de enfrentarnos a los huevos en los datos:

* Si falta algún valor de alguna de sus variables, eliminar la fila, esto obviamente es demasiado drástico, debido a que cada dato cuenta y puede ser un dato mayor
* En imputación de valores perdidos, intentaremos recomponer los datos para dejar de tener el hueco.

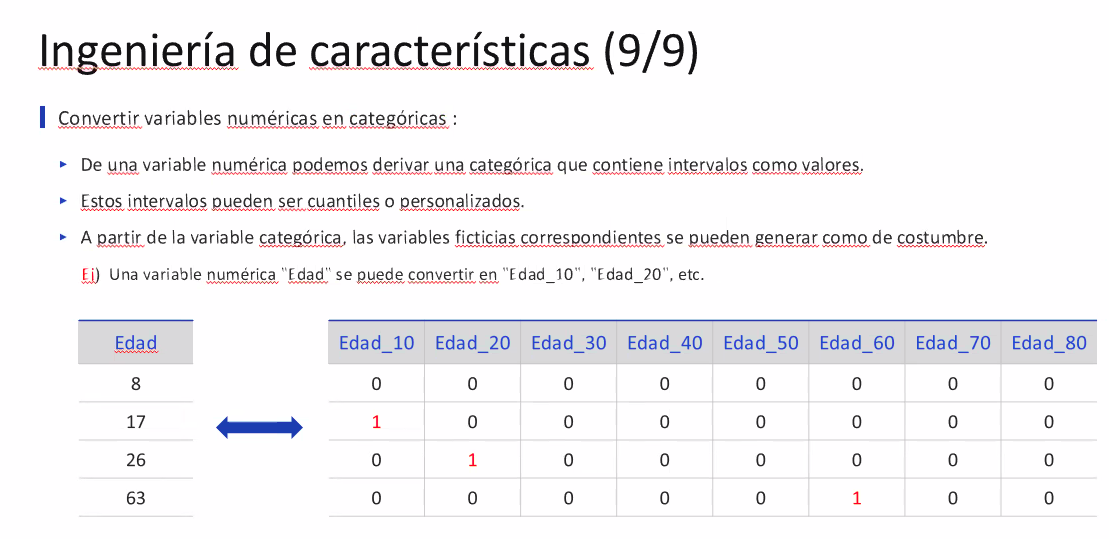
En el tratamiento de datos atípicos se debe tomar la decisión de eliminarlos o dejarlos, si decidimos dejarlos el modelo no va a ser capaz de generalizar, el cuál es el objetivo final de ML.

### Ingeniería de características

Es el proceso de inventarnos nuevas variables, nuevas columnas, con el dataset que ya tenemos, para así enriquecerlo, teniendo en cuenta que las transformaciones lineales no suelen aportar mucho, como por ejemplo tener, unidades y precio por unidad y saber cual es el costo total.

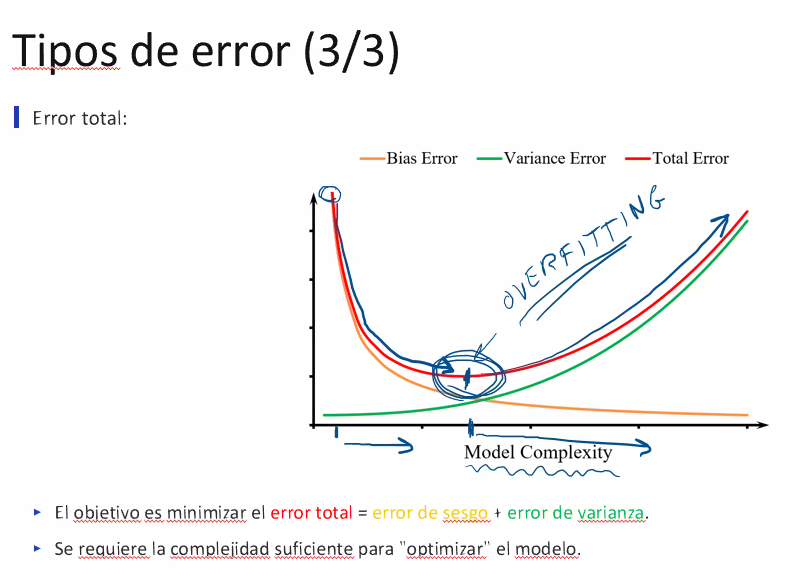
### One hot encoding

Se basa en transformar una serie de variables en características, por una serie de 0s y 1s para evitar dar más peso a unas variables que ha otras.

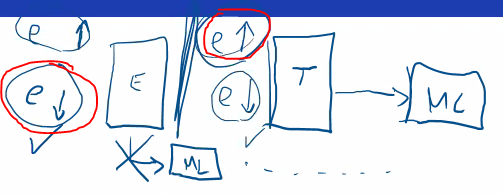


### Overfitting

El overfitting es lo que sucede cuando nuestro proceso se empieza a aprender de memoria el dataset, a poco que le preguntes una cosa ligeramente distinta a lo aprendido, no sabrá solucionarlo.



Podremos detectarlo porque, el modelo ya entrenado obviamente tendrá un error muy bajo si le preguntamos sobre el dataset que ha aprendido, para identificar el overfitting lo pasaremos a un conjunto de datos tipo test, para ver si está generalizando o no. Si el error es bajo en entrenamiento y alto en test, tenemos problemas de overfitting.



### Regresión lineal

De manera sencilla, intenta pasar con una línea por la mayoría de puntos.

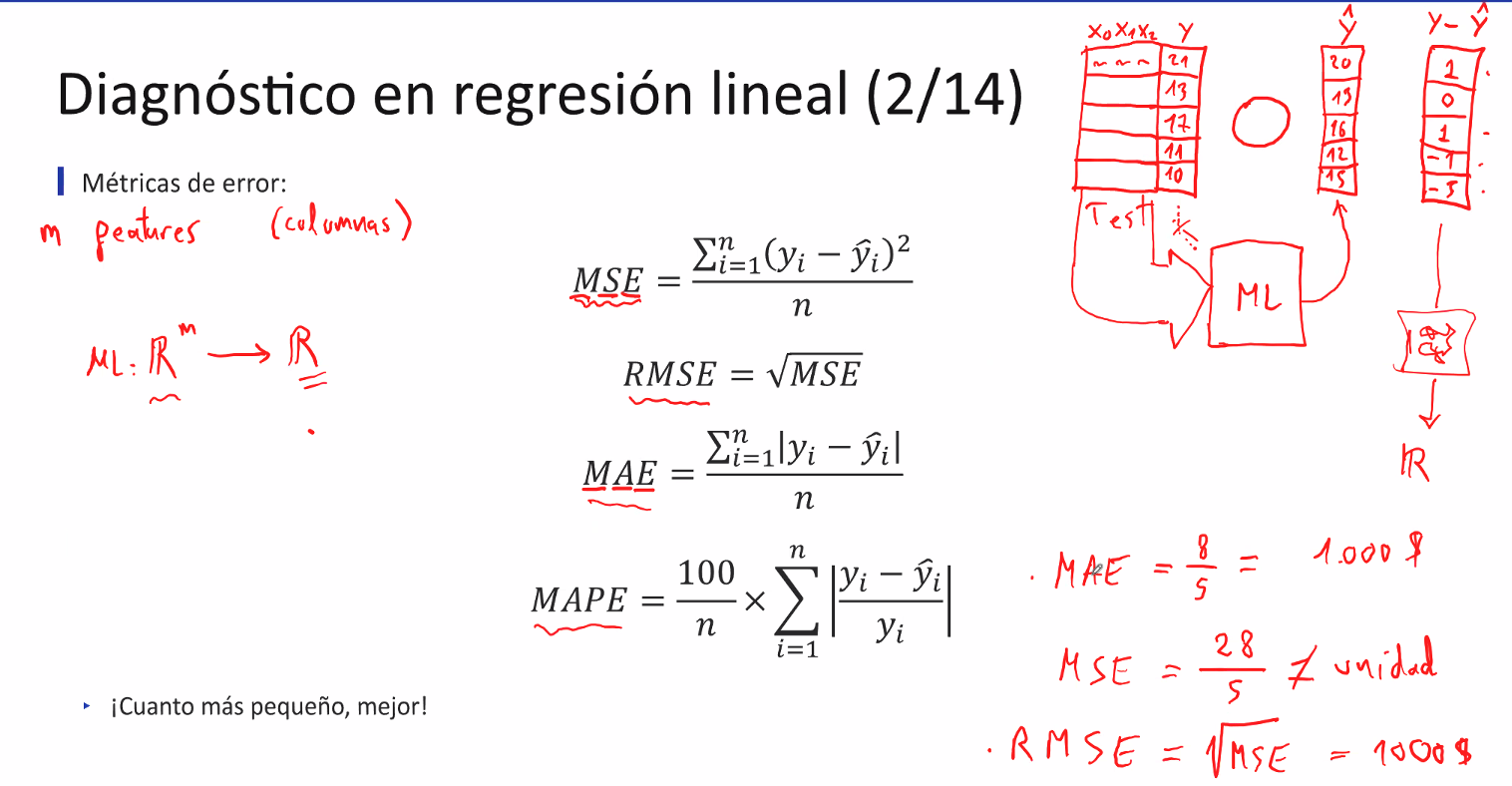
Sirve para determinar si existe una correlación entre los datos.

Fortalezas es que es un modelo de caja blanca, y es sencillo.

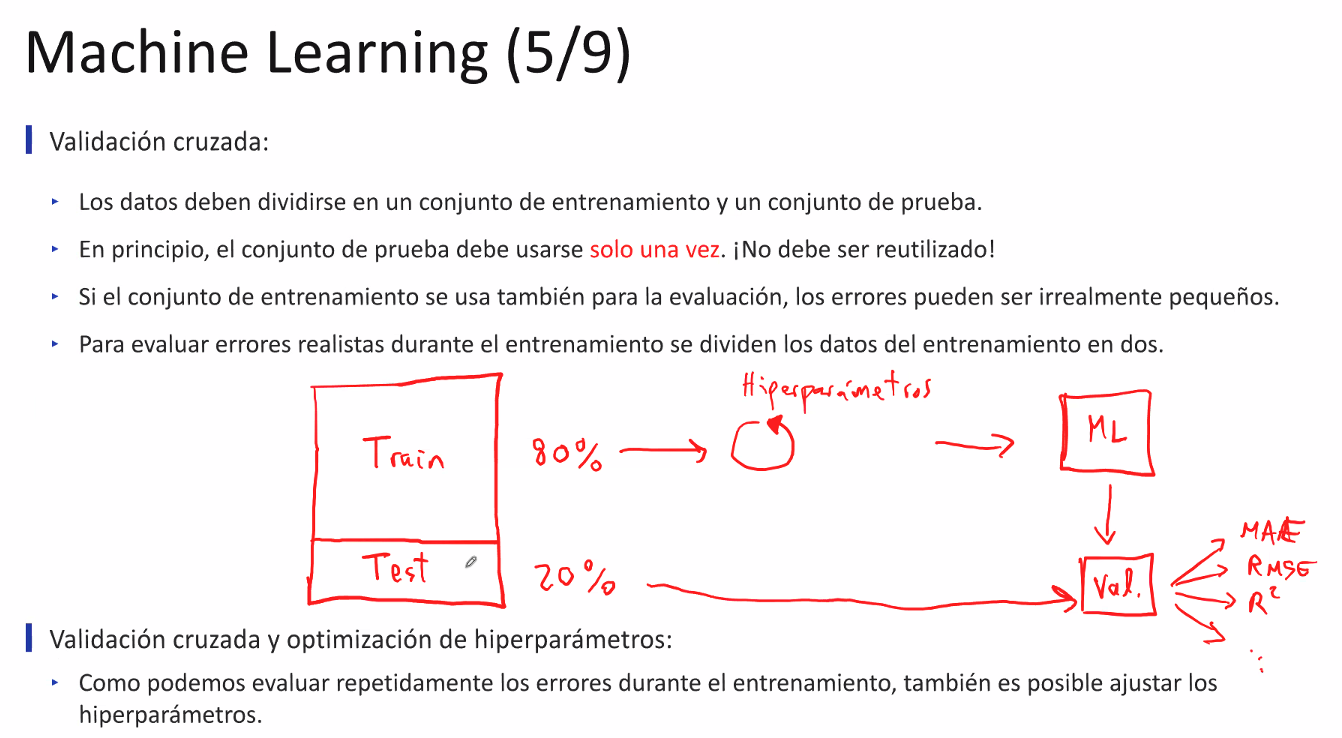
Inconvenientes:

* **Sobreajuste**, Es el problema que surge cuando sobreentrenas un modelo, porque se “memoriza” los datos, por lo tanto, cada vez que sale de su zona de confort no sabrá responder correctamente. – Se arregla con la normalización
* **Outlayers**, son los datos que tenemos que están o muy por encima de la media o muy por debajo, este tipo de modelo es muy sensible a estos datos, por lo tanto es mejor no tenerlos en cuenta en la Regresión lineal – Se arregla con la normalización
* **Escalabilidad**, cuando tenemos datasets más grandes tenemos un problema de coste computacional. – Se arregla con la sgd
* **Linealidad**, este modelo asume que hay una correlación entre la variable objetivo y el resto de variables, casi nunca nos encontraremos con este tipo de relación, se puede comprobar si existe una linealidad entre variables usando – Se arregla con la polinómica

### Diagnóstico de regresión lineal

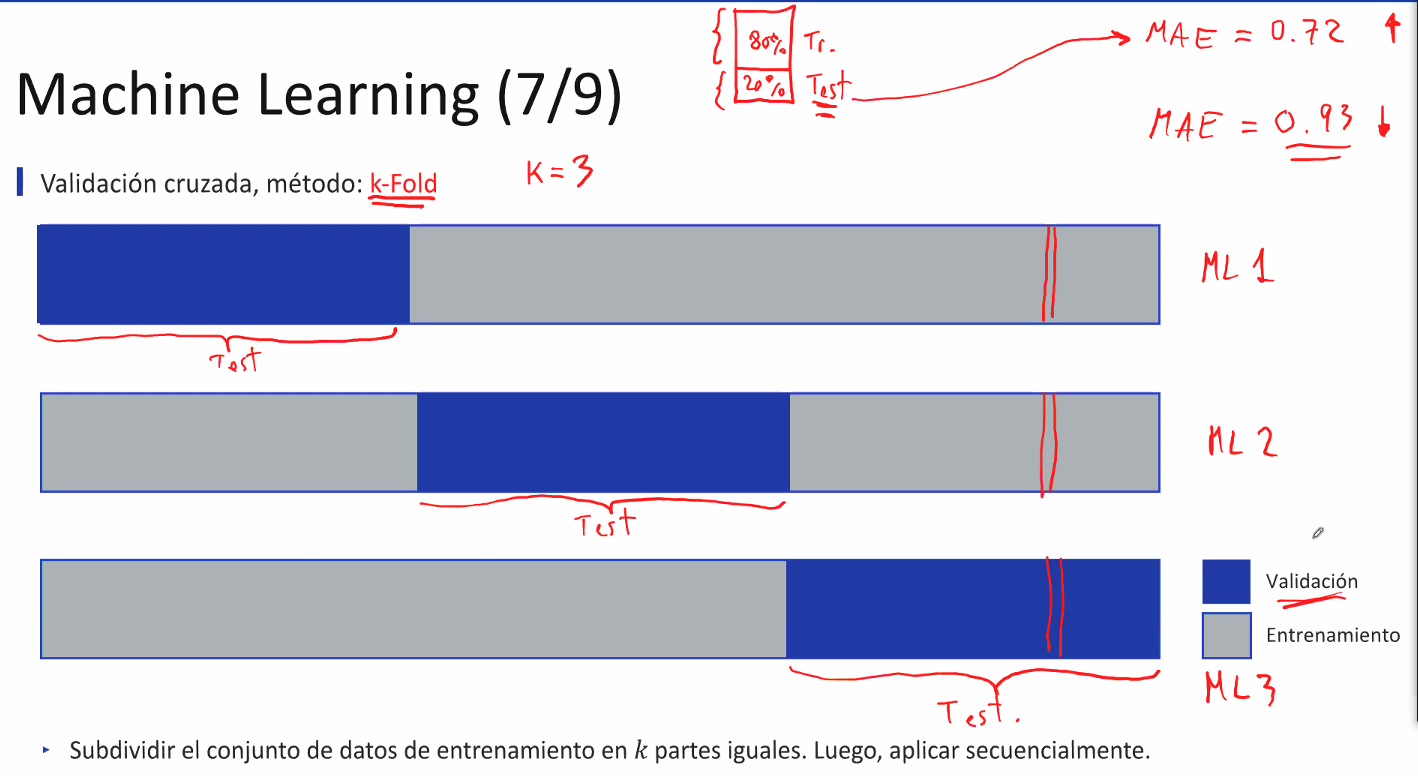


# Validaciones



## Validación cruzada

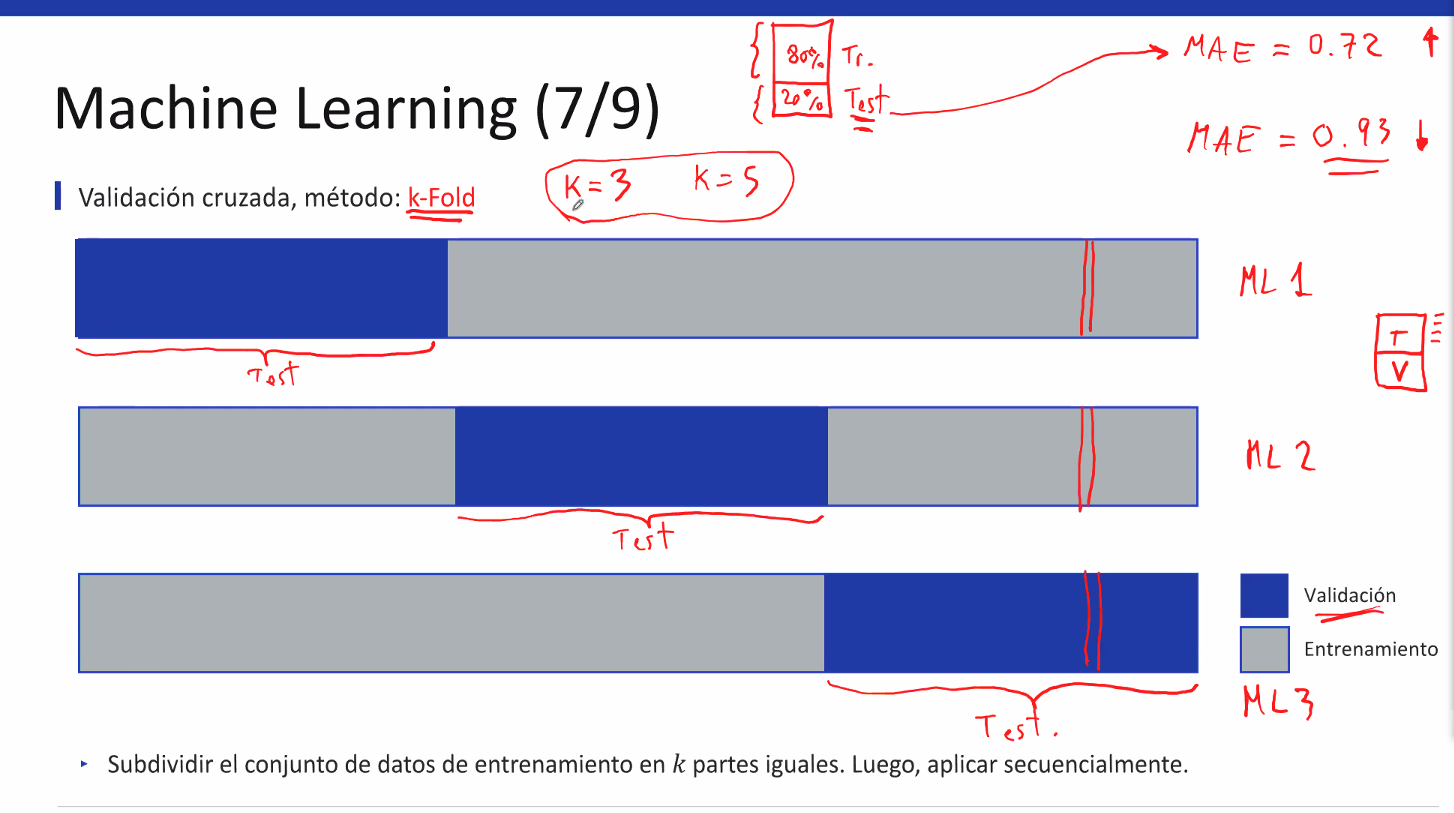
### k-fold



Son tres modelos iguales desentrenados cada uno independiente del otro, en el cual generaremos de manera aleatoria los sets de entrenamiento, para mitigar así la “buena o mala suerte”.

### Leave one out

Haces tantos experimentos como observaciones tengas, nos lleva un tiempo de computo elevadísimo.



## Ridge regression

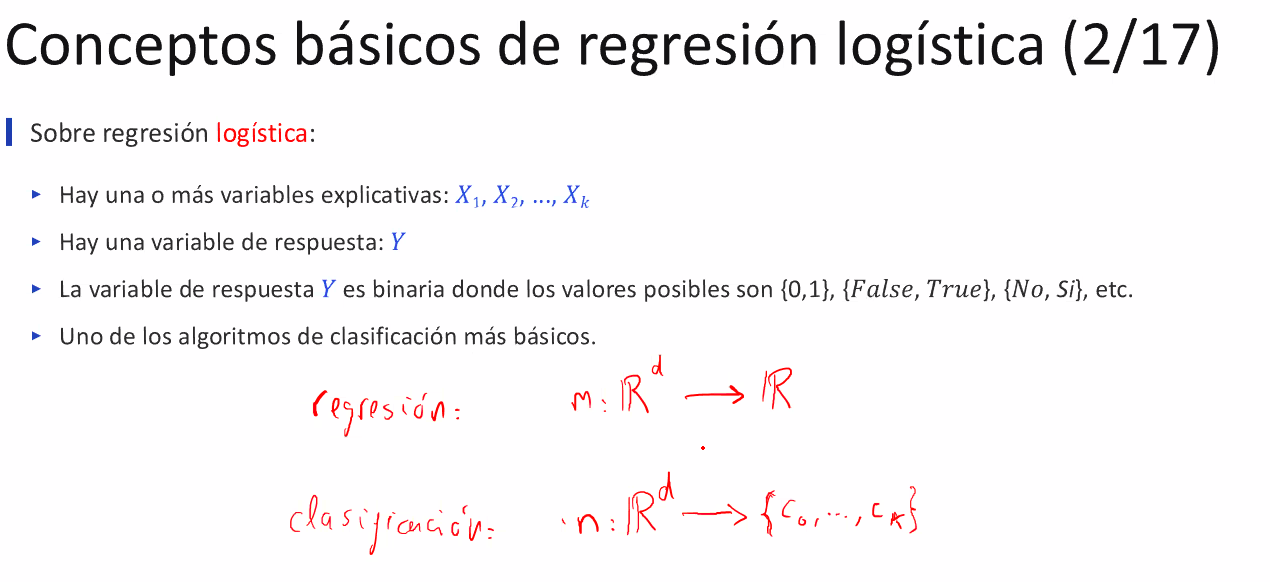
Es una variante de la regresión lineal que evita el sobreajuste o el overfitting añadiendo una penalización para menguarla.

Se basa en el alfa, que deberemos de calcular el valor optimo realizando experimentos, y quedándonos con el valor que sea más optimo.

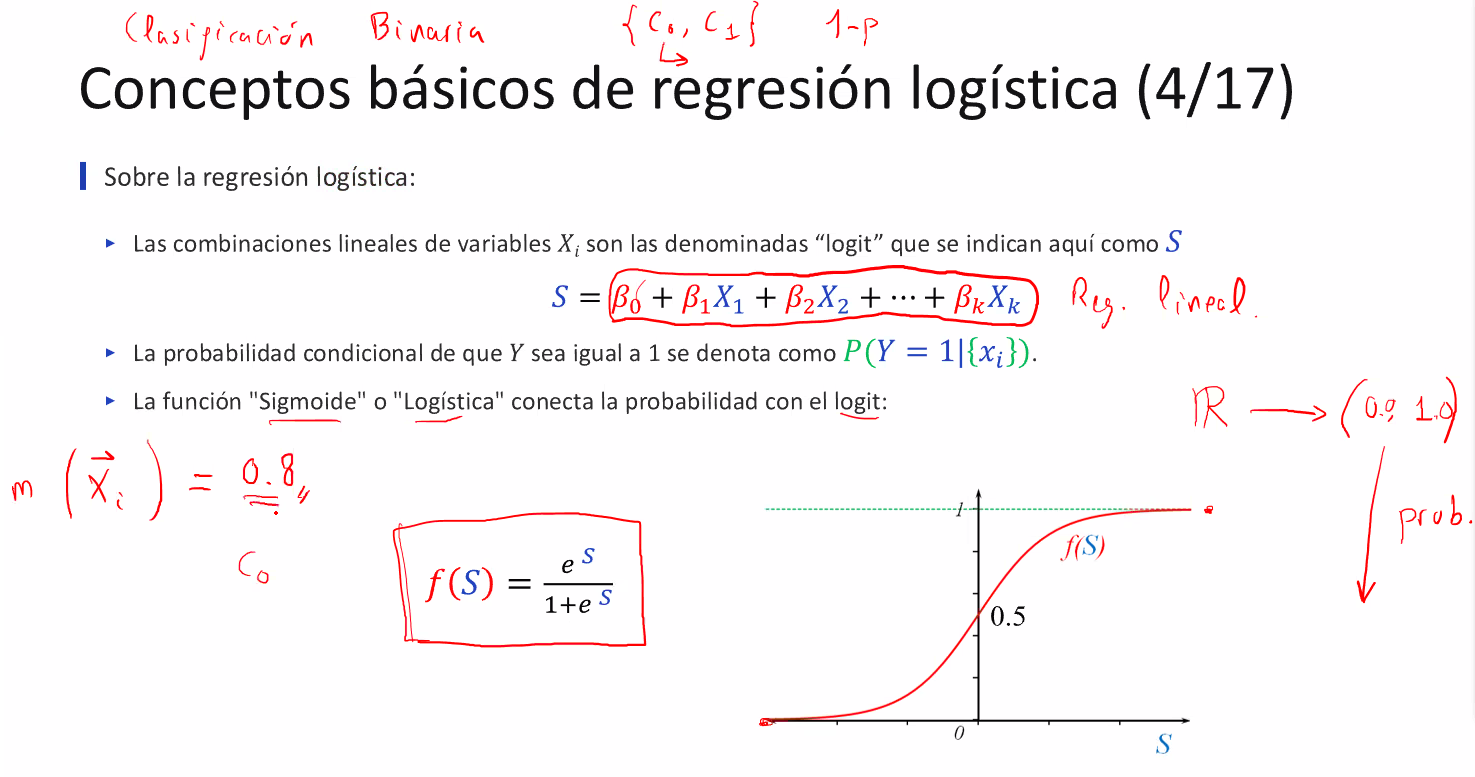
O en ingles  **loss function.**

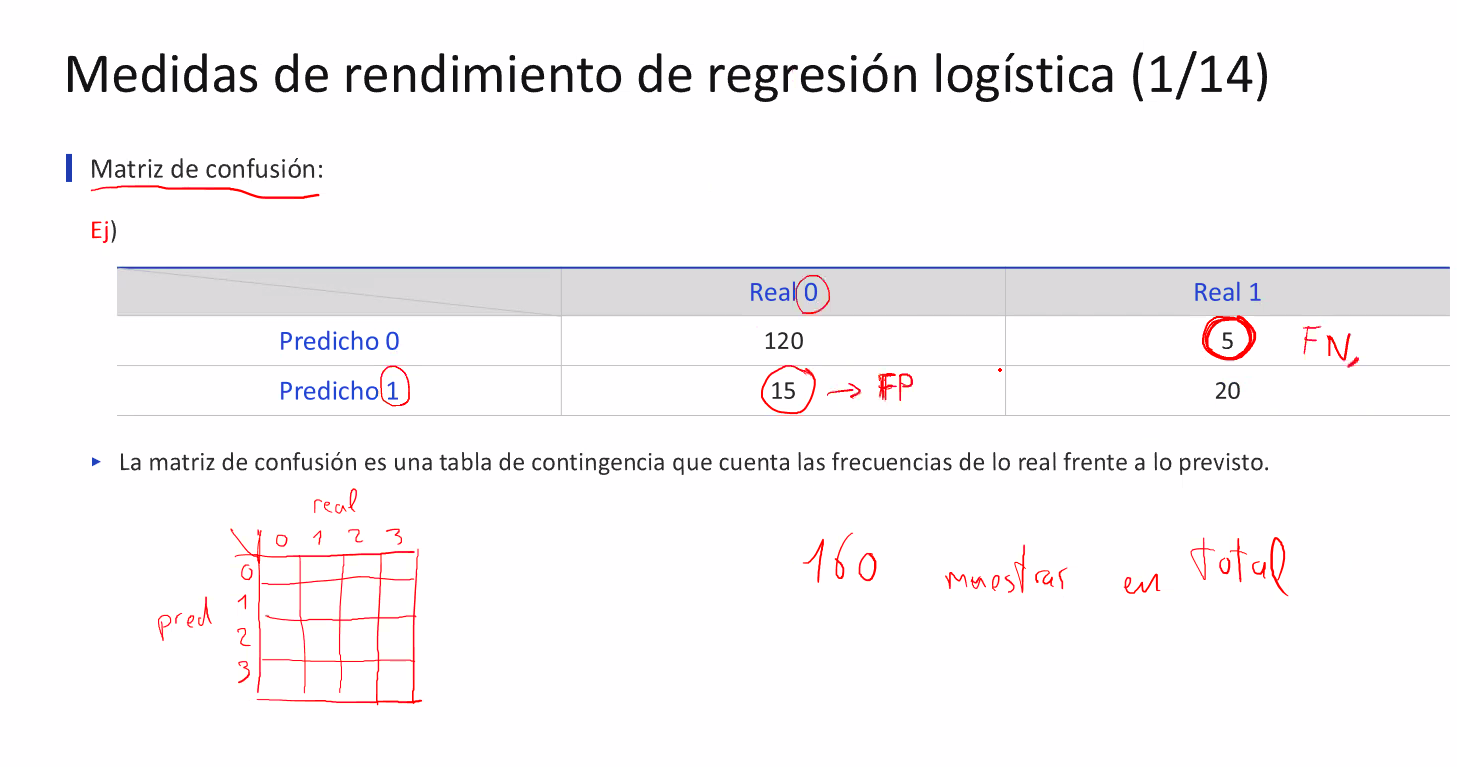
# Clasificación

## Regresión logística



Aplicaremos la regresión logística para convertir nuestro modelo de regresión lineal en uno de clasificación, la predicción dará un número entre 0 – 1 que será la probabilidad (p)





Es necesario tener todas las medidas de rendimiento, porque el dataset puede tener un bias importante, como por ejemplo:



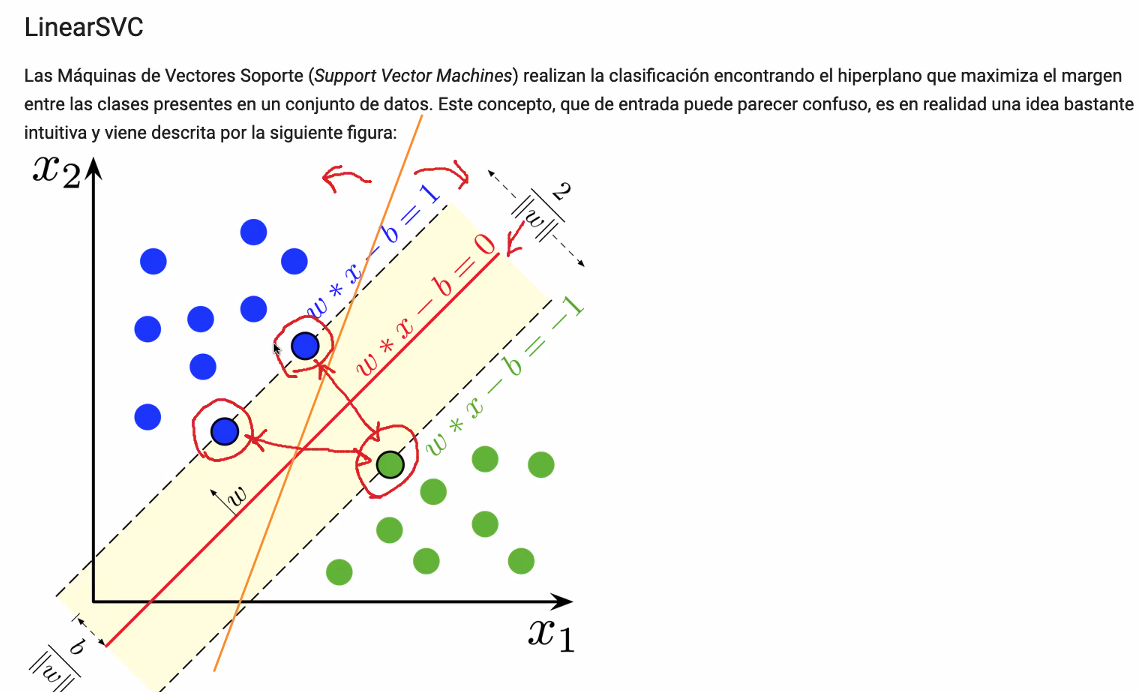
Donde el accuracy sería del 99%, pero eso no significa que nuestro modelo sea válido, hay que comprobar todas las demás medidas.

Es importante en clasificación, tener datos iguales de cada tipo de clasificación, es decir si tengo un dataset de 150 datos y tengo una clasificación con 3 opciones, lo ideal sería que fuesen 50,50,50.

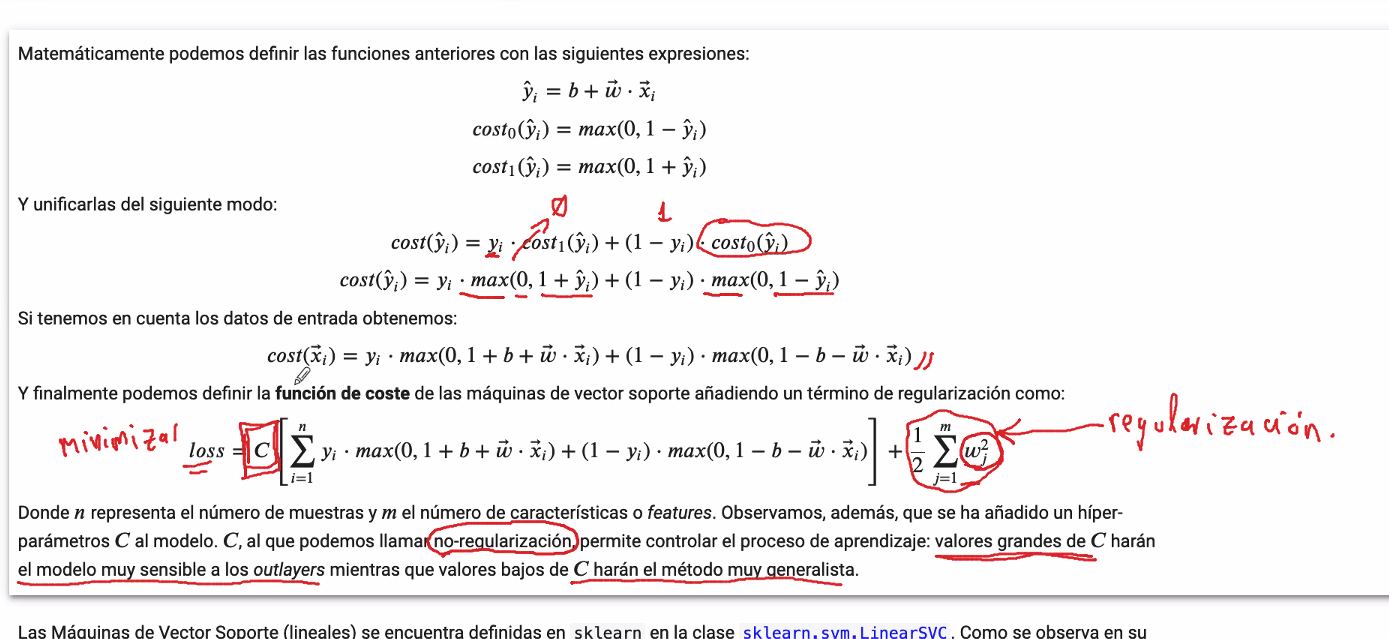
## Support vector machines (SVM)

Es un modelo configurable.

No solo encuentra la recta que separa las clases, sino que además maximiza la distancia, es decir busca las observaciones de cada clase que están más cerca una de las otras, y realiza así una frontera.

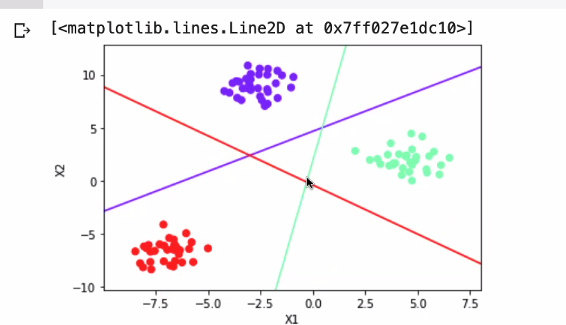


Las observaciones más cercanas entre sí se llaman **vectores soporte**.



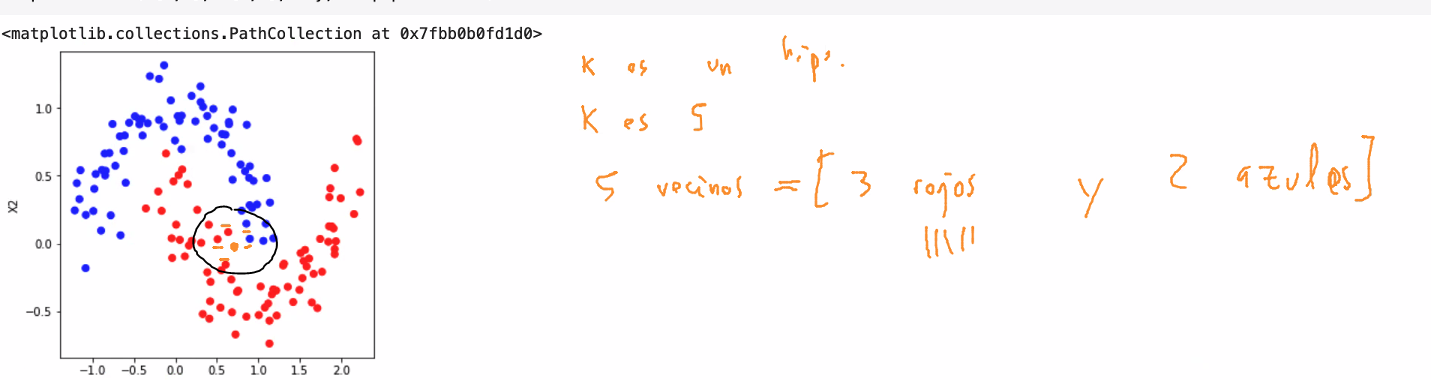
El parámetro C si lo ponemos muy alto vamos a ser muy sensibles a los outlayers, y muy bajo aprenderá peor.

Si existe un punto entre las fronteras, que es un poco terreno de nadie, calculará la probabilidad de que pertenezca a una clase con la cercanía de las fronteras. Si se encuentra solo entre dos fronteras, seguirá calculando la probabilidad de la tercera, pero dará un porcentaje cercano a 0.



## KNN

Calcula los 5 vecinos del punto a predecir, y según la mayoría clasificará ese nuevo punto de una manera o de otra.



El método como tal no hace un proceso de entrenar, ya que solo se guarda la matriz de puntos del entrenamiento, cuando le pidamos predecir un punto se fijará en los de alrededor.

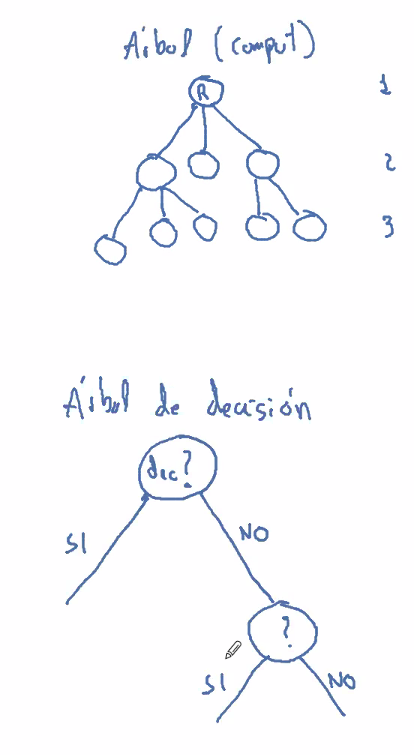
## Naïve Bayes Classifier

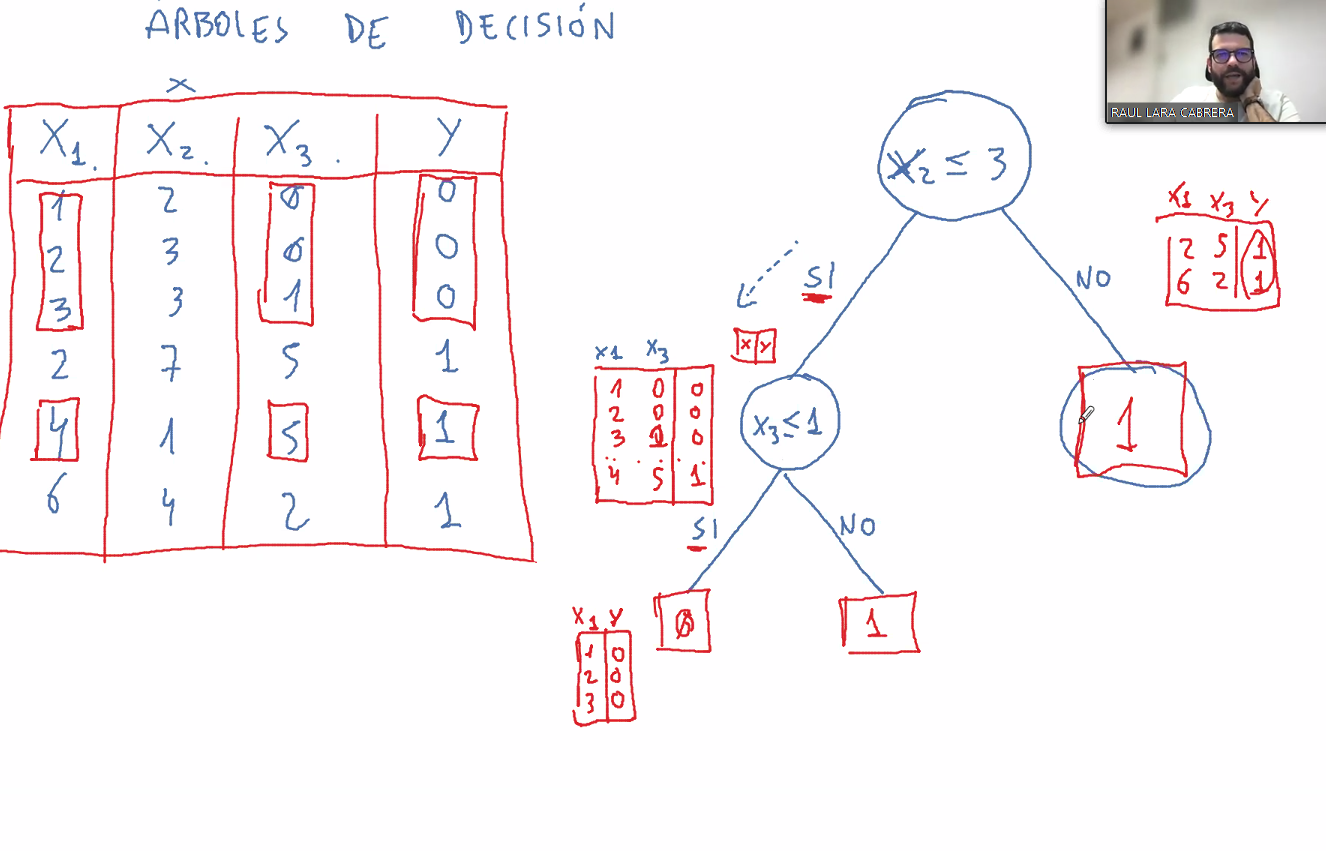
Se hace la suposición de que las variables implicadas no están relacionadas entre sí, es lo que se llama variables independientes.

## Árboles de decisión

Hasta ahora todos los modelos que hemos visto necesitaban que las variables estuvieran estandarizadas, normalizadas, que fuesen numéricas…

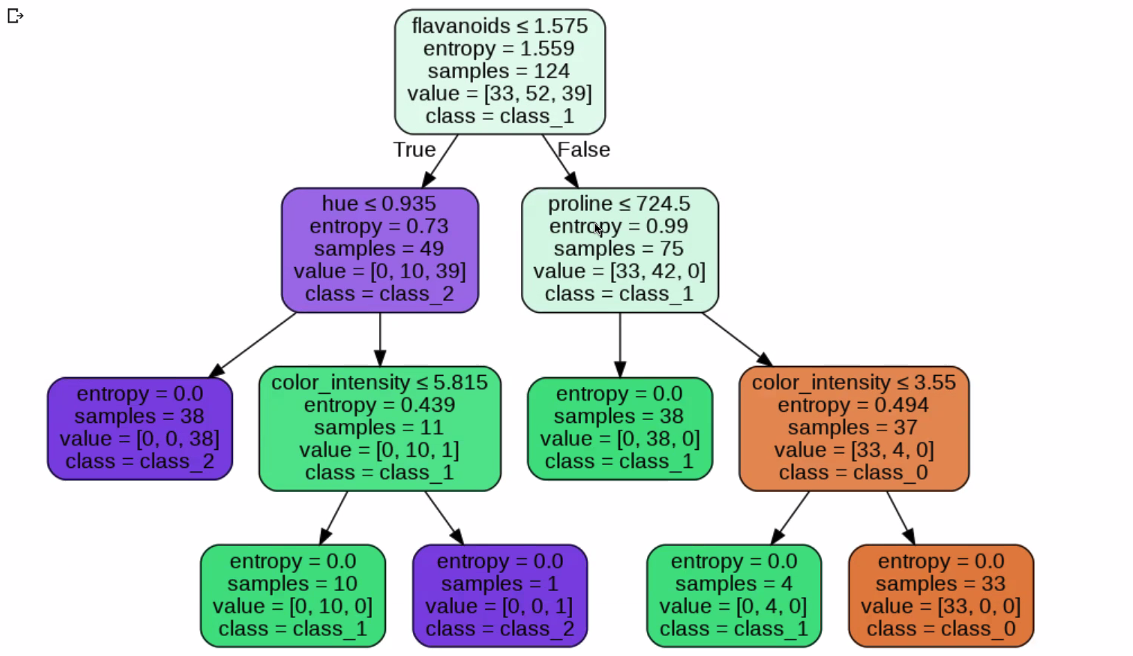
Este modelo NO necesita cumplir con estas restricciones.





Este modelo es extremadamente propenso a cometer overfitting.

Es un proceso altamente explicativo:



La **entropía** es la incertidumbre digamos, perdemos entropía cada vez que obtenemos información.

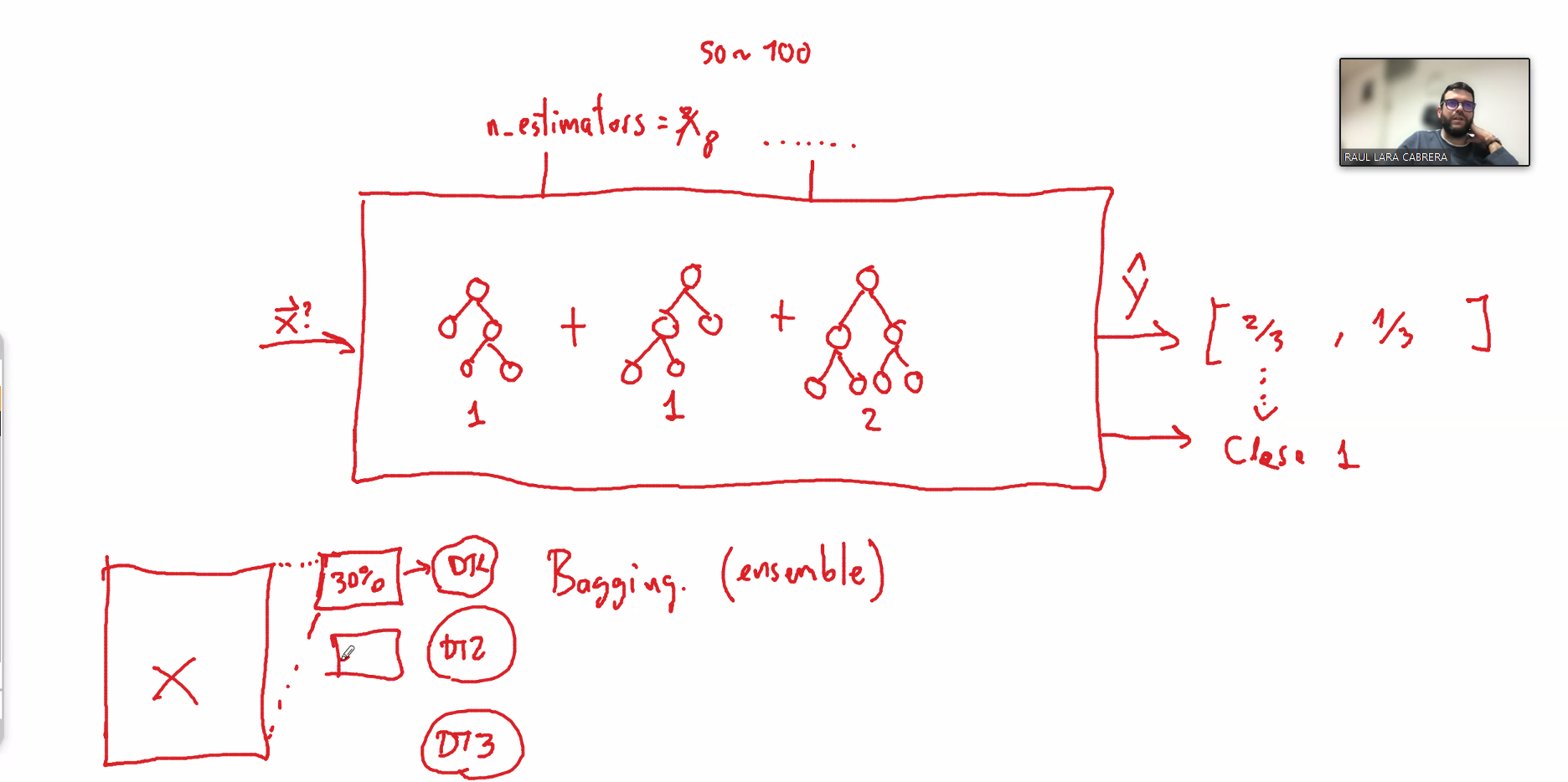
**Samples** nos indica el número de total filas en nuestro dataset.

**Value[clase\_0, clase\_1, clase\_2]** 🡪 Es decir que ese array nos demuestra cuantas samples de cada tipo hay en ese fragmento del dataset.

## Random fotest

Examina muchos árboles de decisión que tengan overfitting, se realiza un agrupamiento final por mayoría en el resto de árboles de decisión, es decir si quiero calcular que x parámetro sean que clase, si el 80% decide que es clase\_1, obtendremos ese valor como correcto.

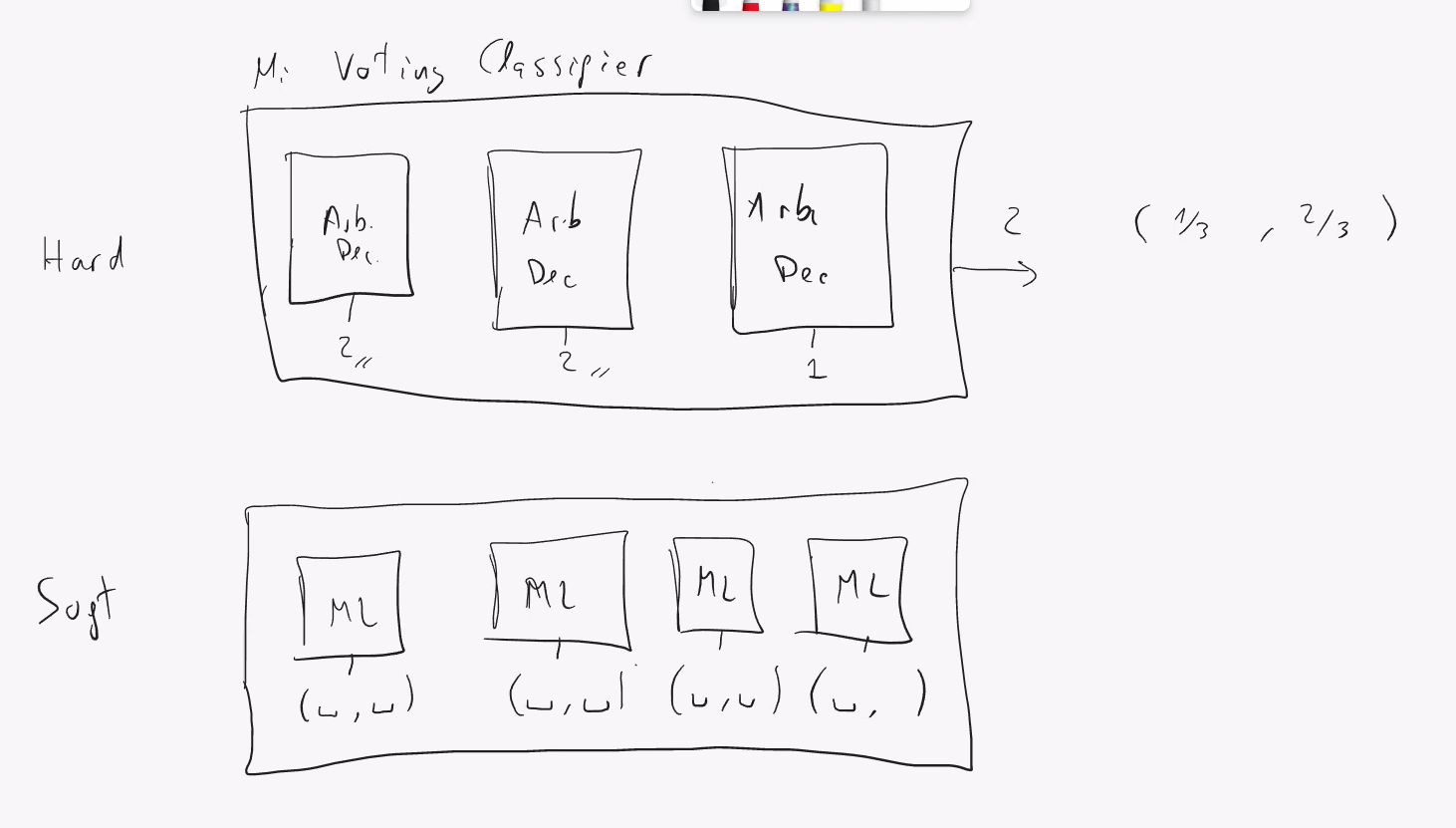
Cada árbol se entrena con un conjunto aleatorio de nuestro dataset de entrenamiento.



# Ensembes

Están formados por N algoritmos de clasificación.

## Voting Classifier

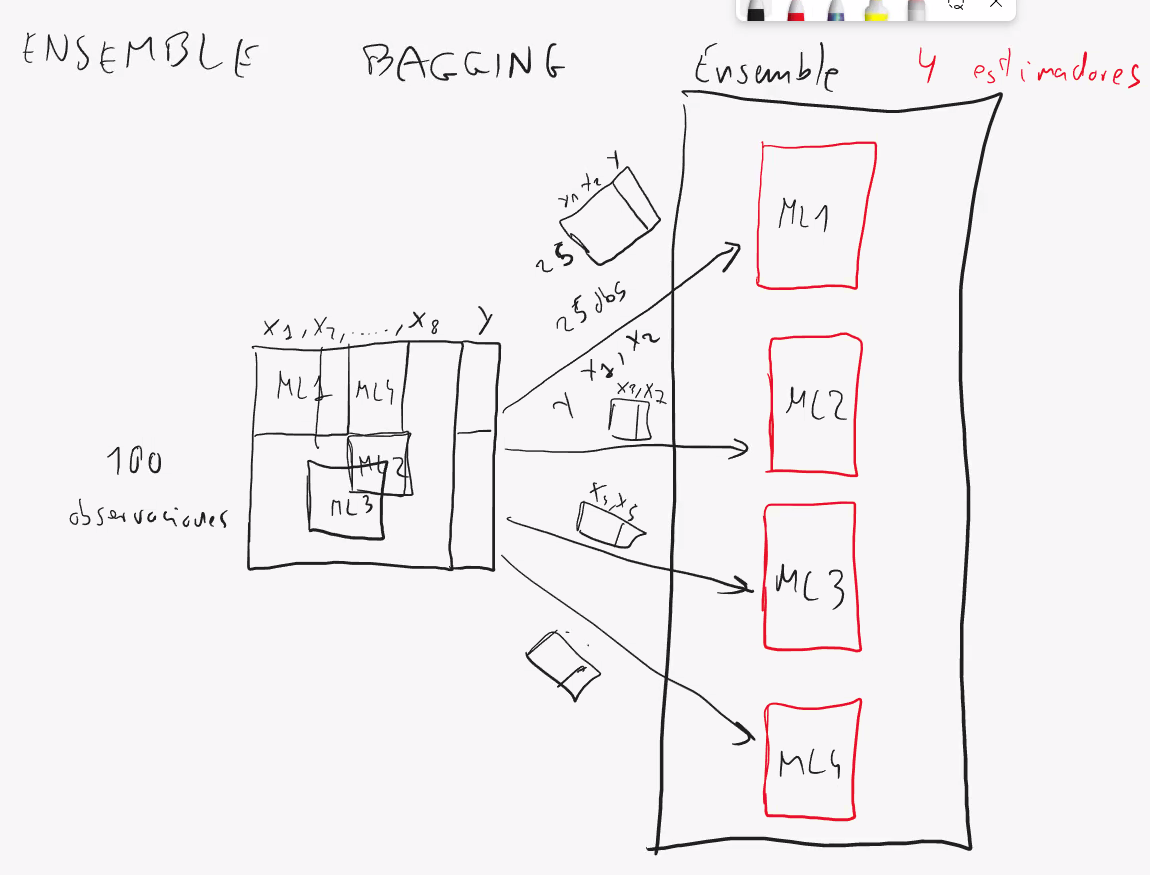


El **soft** se pueden convinar distintos métodos de ML (Regresión Lineal, Random forest..)

Normalmente es más interesante utilizar **soft** voting más que hard voting.

## Bagging

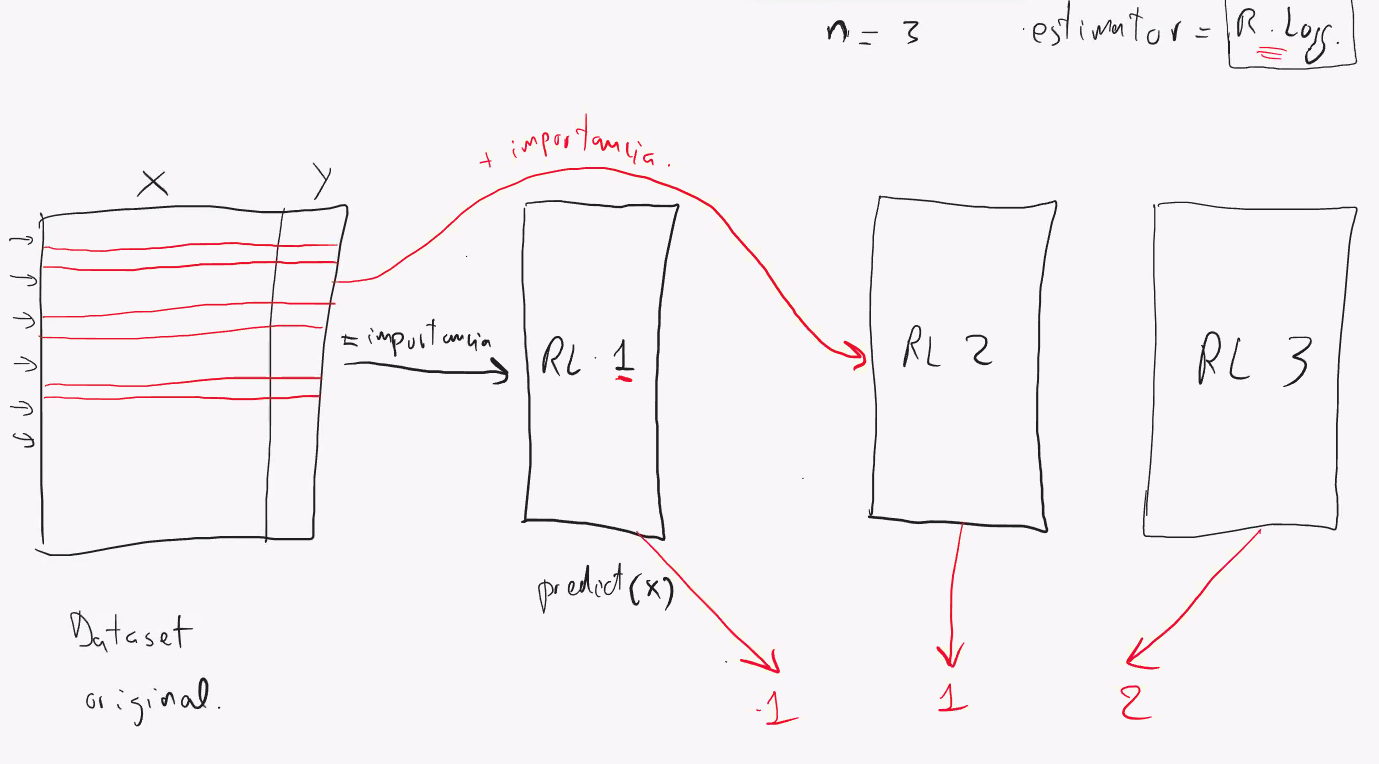
Desordena las filas y para cada algoritmo de clasificación dividirá los datos que tenemos entre ellos, además selecciona aleatoriamente las variables que se les pasa.



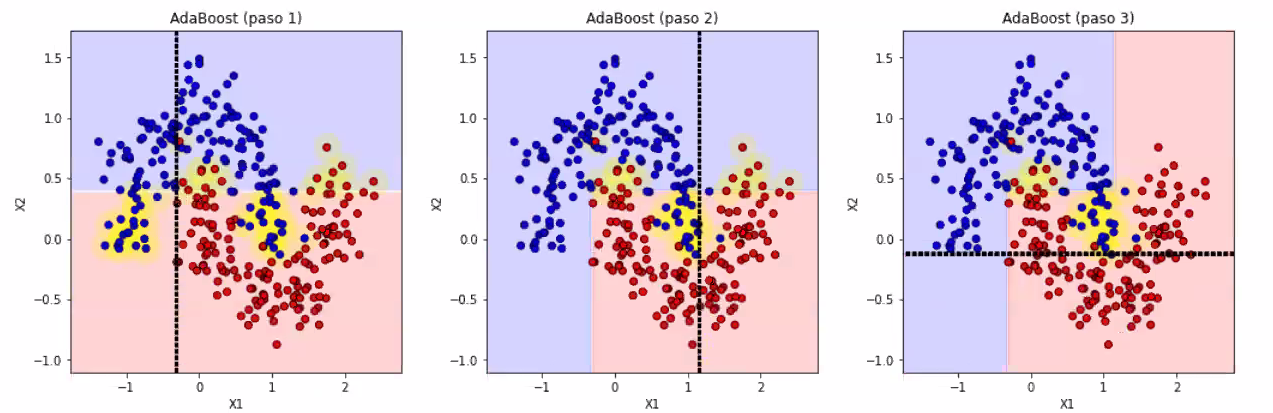
El modelo interno tiene que ser el mismo repetido por el número de estimadores.

## Boosting

El modelo aprende de lo que falla el anterior

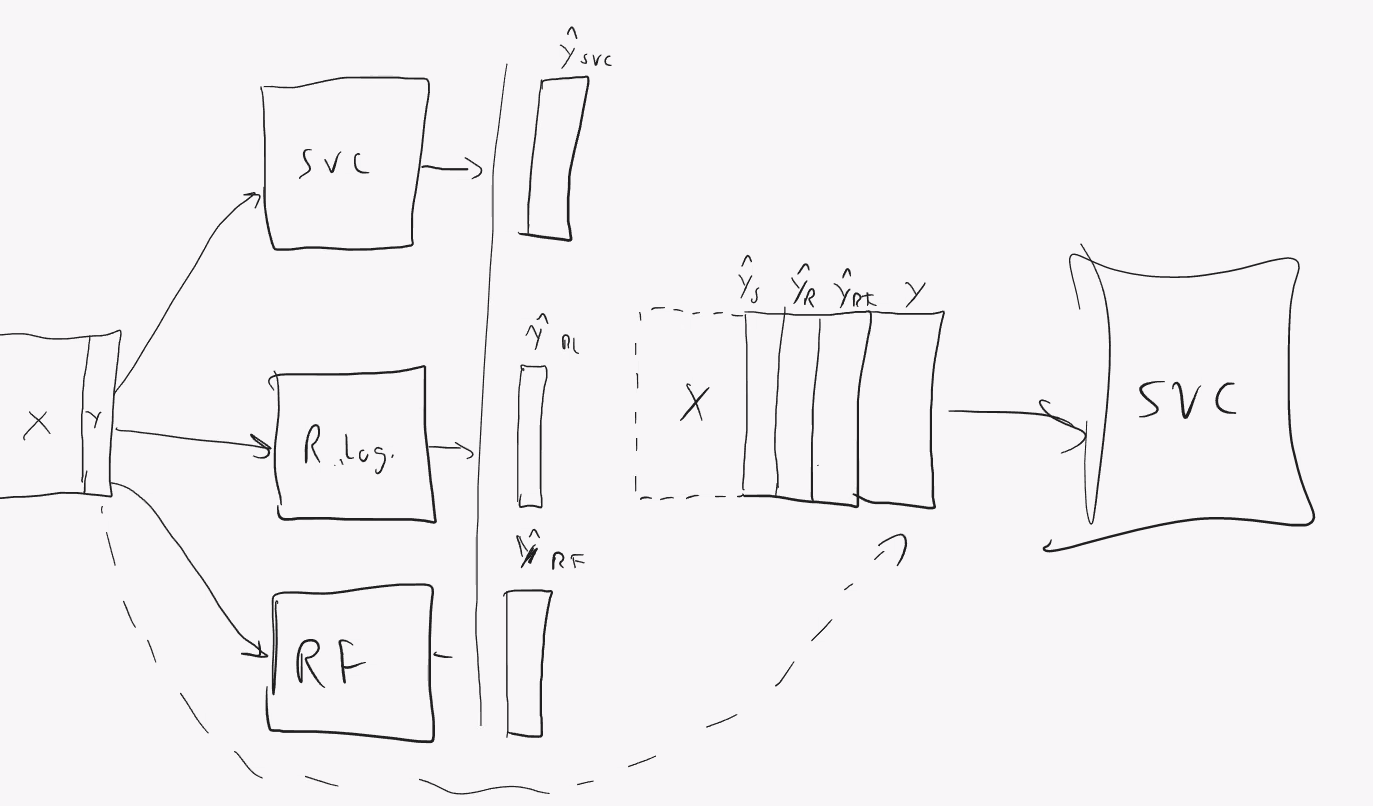


Aquí podemos ver por ejemplo que se va nutriendo de cada modelo a partir de lo que ha fallado en los anteriores.



## Staked

Entrenas un clasificador cuya entrada son las salidas de otros clasificadores entrenados previamente.



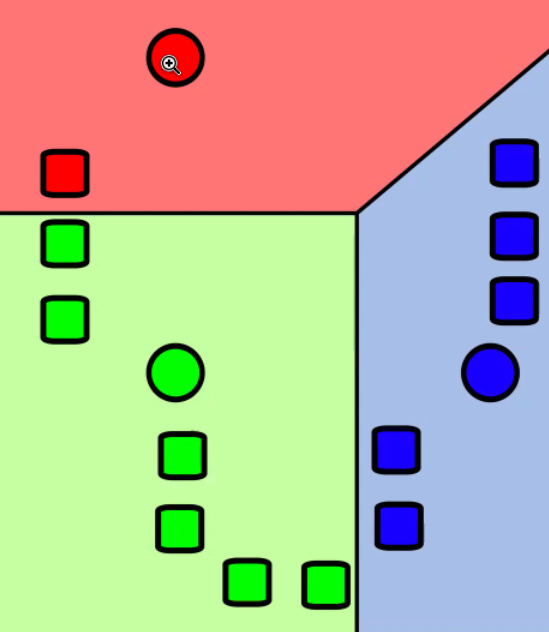
# Unsupervised learning

## K-Means

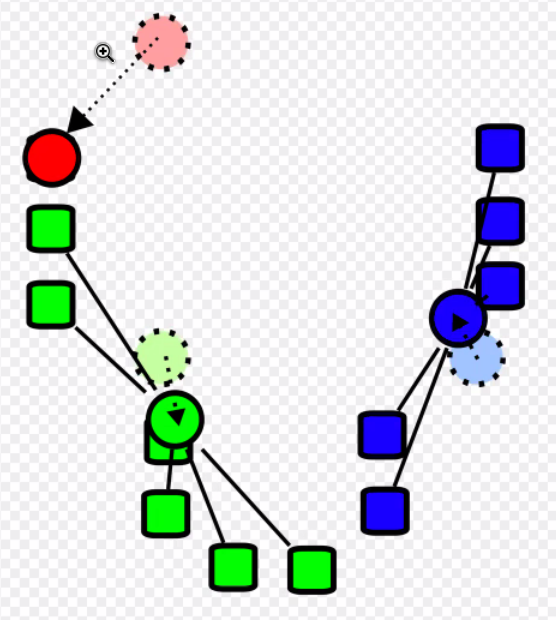
Es un algoritmo lineal, separa por un plano. Es el algoritmo más básico y el que mejores resultados da.

En K-Means se tiene que especificar de antemano la cantidad de grupos que tenemos que dividir el dataset. Este sería nuestro **hiperparametro,** el cual se le llama K.

Primero este método lo que hace es asociar cada observación con su media más cercana, siendo las medias los círculos. Estos centroides (es así como se llaman los puntos), se colocan aleatoriamente escogiendo datos de nuestro dataset.

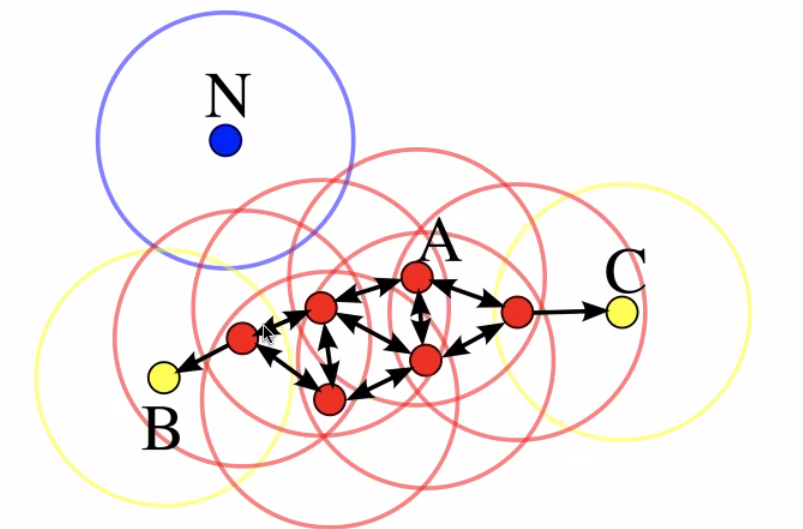


Después de esto, las medias se mueven a una media de las observaciones asociadas a ellas. Colocándose así en un sitio más acertado cada vez. Se repite este proceso n veces.



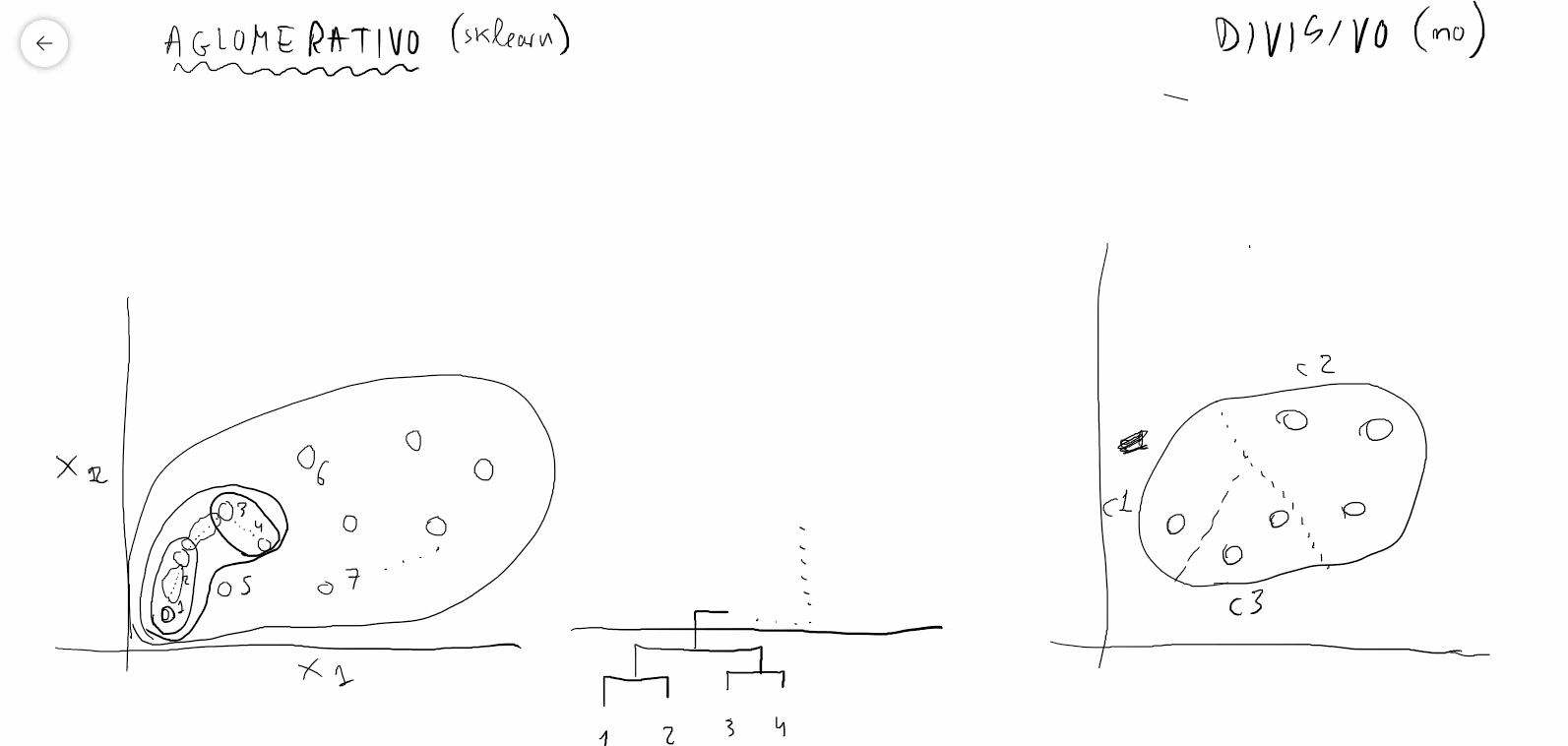
## DBSCAN

Este algoritmo tiene dos hiperparametros, el radio del círculo y el número mínimo de puntos que debe de haber dentro de ese radio para considerarse de la misma clase.



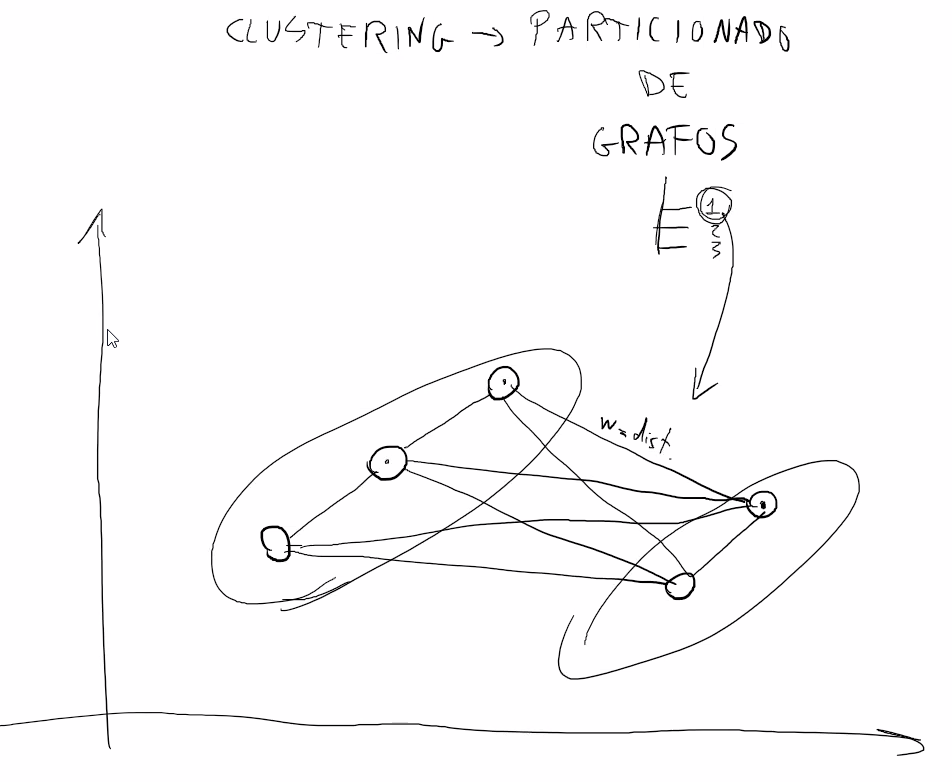
En este caso N sería un outlayer. Es decir un punto descartado normalmente porque se encuentra muy desplazado del computo global de puntos.

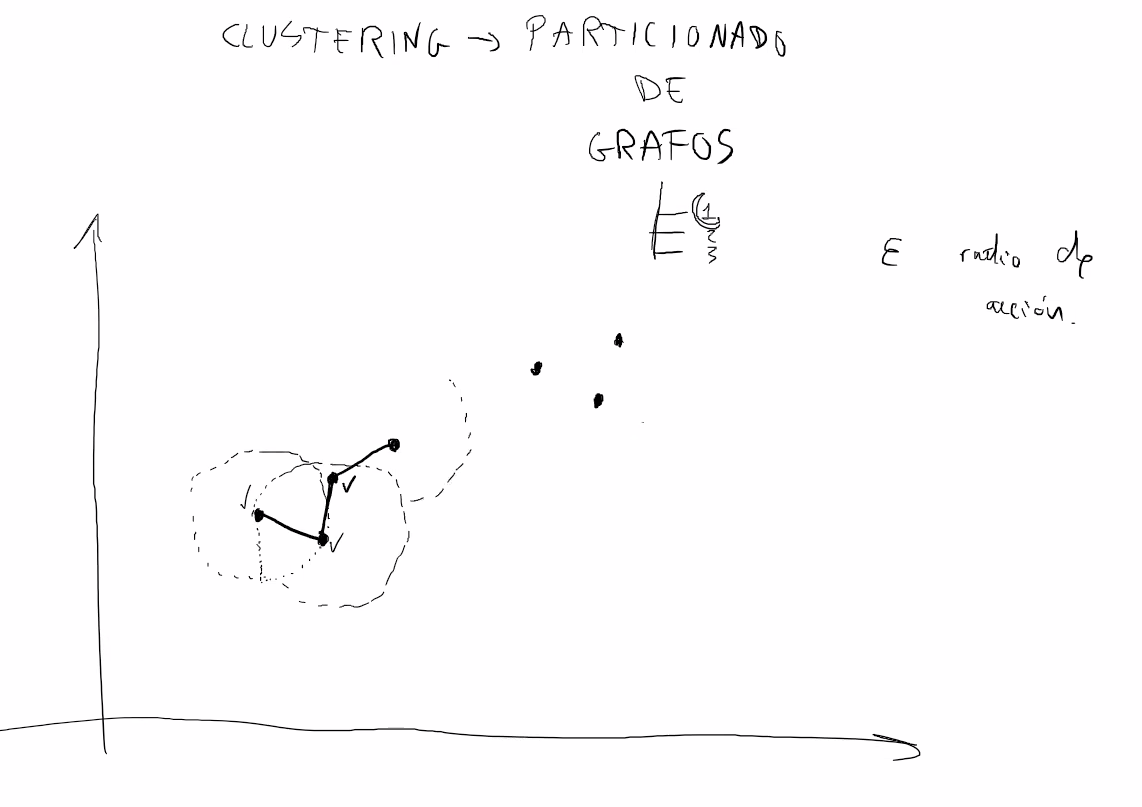
## Aglomerativo y divisivo



Aglomerativo también se llama Bottom-up, aglutina aquellos puntos que estén más cercanos en clusters.

## Grafos





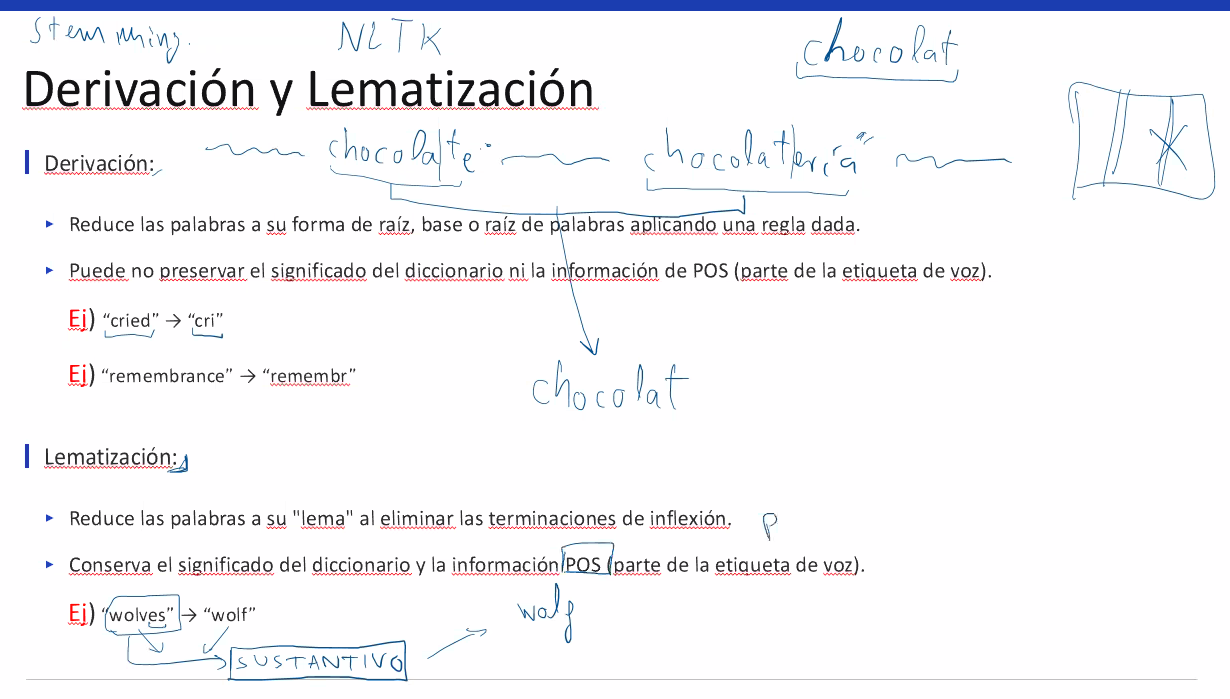
# PNL (Procesamiento de lenguaje natural)

**Corpus,** dentro del PNL es el conjunto de datos. En texto. En esta última parte se diferencia del dataset, ya que el dataset es númerico.

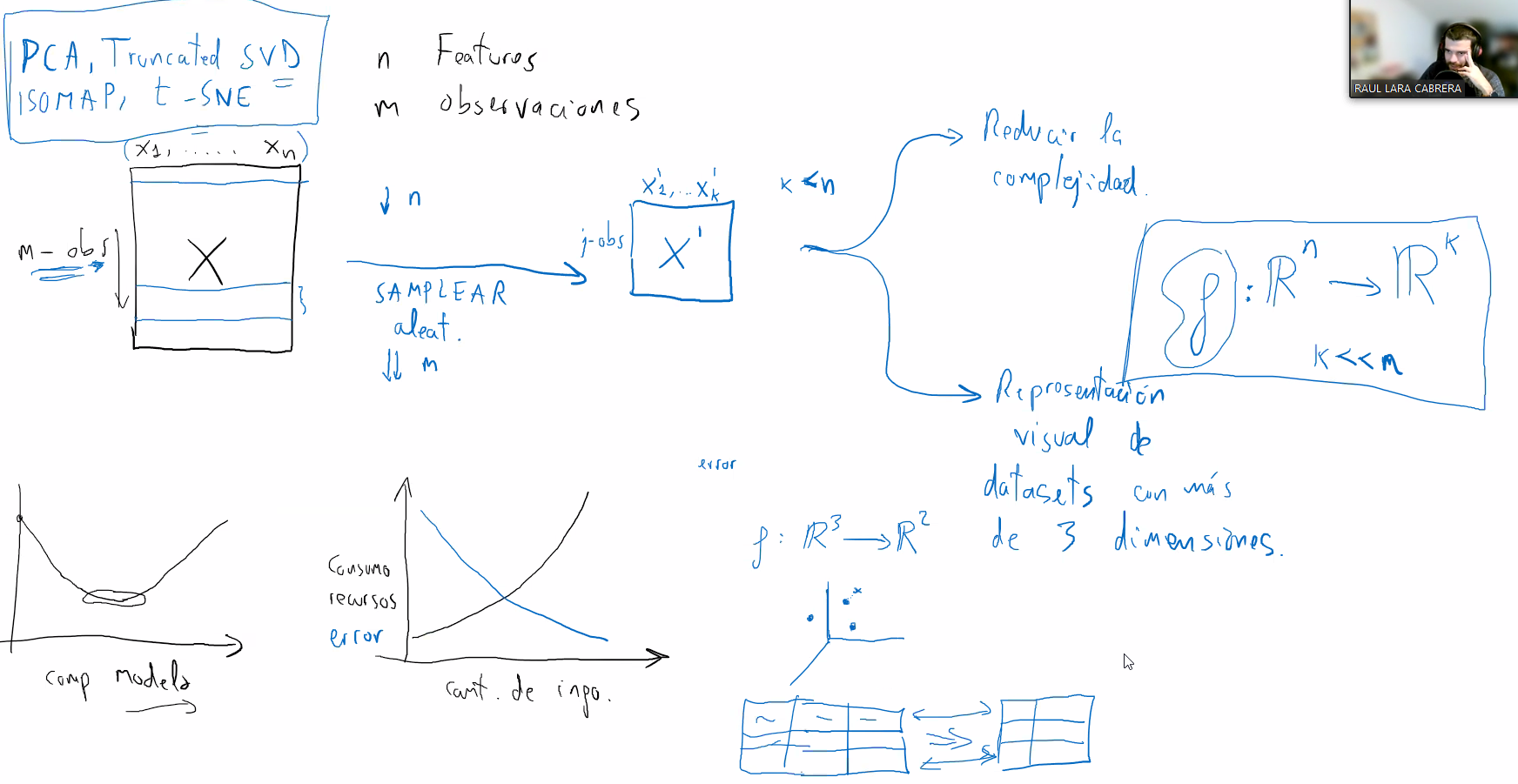


**Stopwords**, son aquellas palabras vacías que no contienen información semántica de uso general.

**Stemming**, reduce las palabras a la raíz.



## Reducción de dimensiones

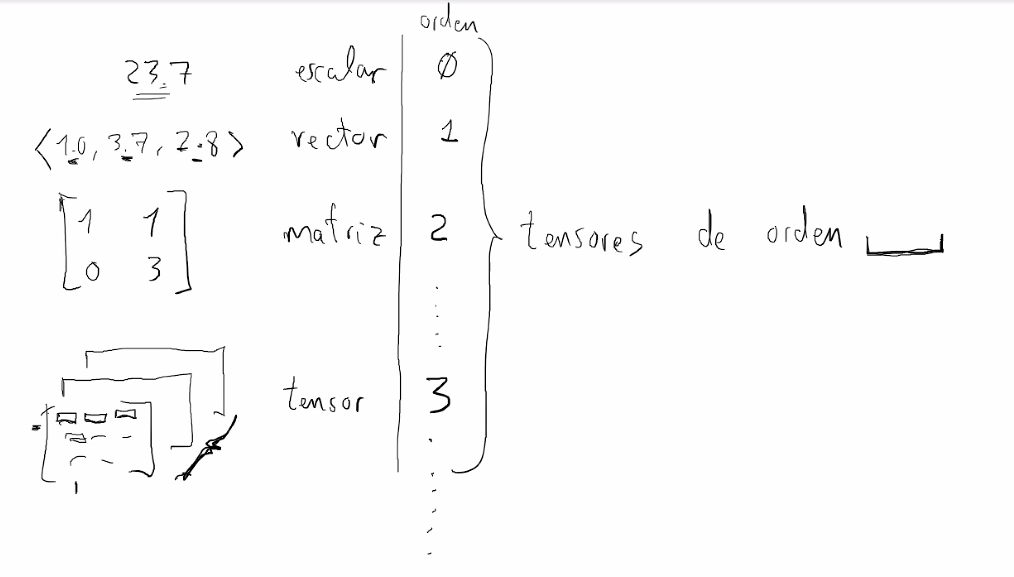


Se consideran aprendizaje no supervisado ya que con el dataset que vamos a trabajar no está etiquetado, aunque lo este, esa etiqueta no se podrá tocar, ya que nunca querremos reducir la salida.

# Deep learning

## Tensores

Es una estructura de datos donde se pueden tener n dimesiones, igual que los arrays.

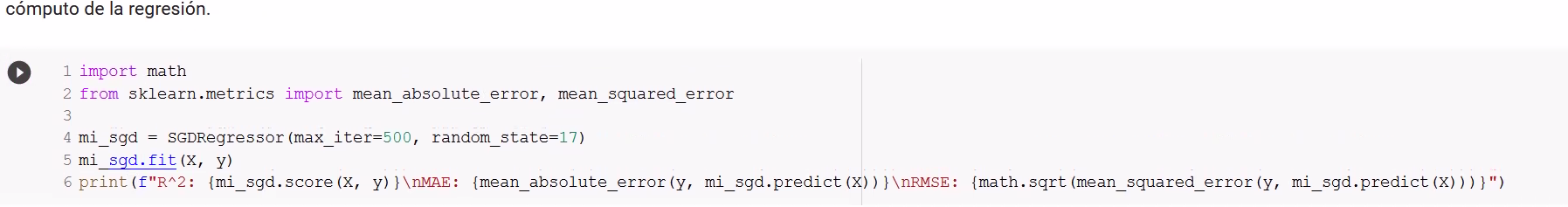


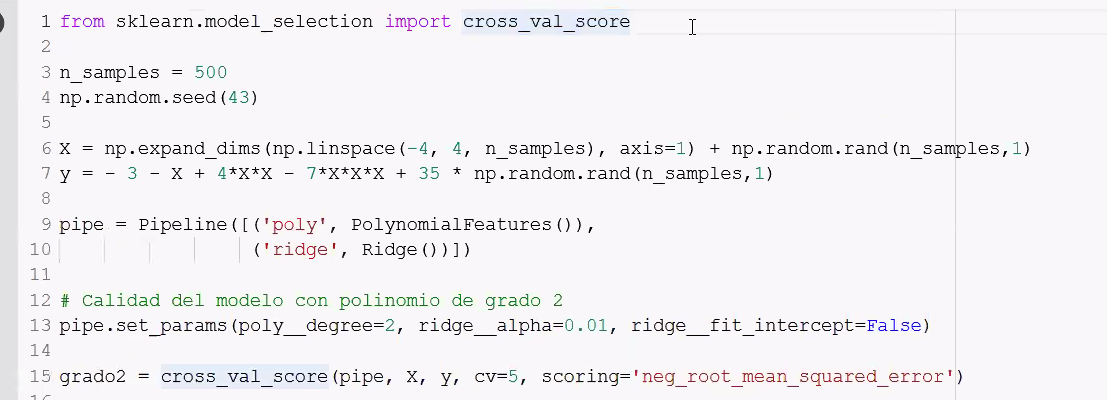
Los tensores tienen un orden y unas dimensiones.

## Variables de Tensorflow

Se corresponden a los parámetros que se introducen cuando entrenas un modelo. Estas variables se ajustan automáticamente con el entrenamiento.

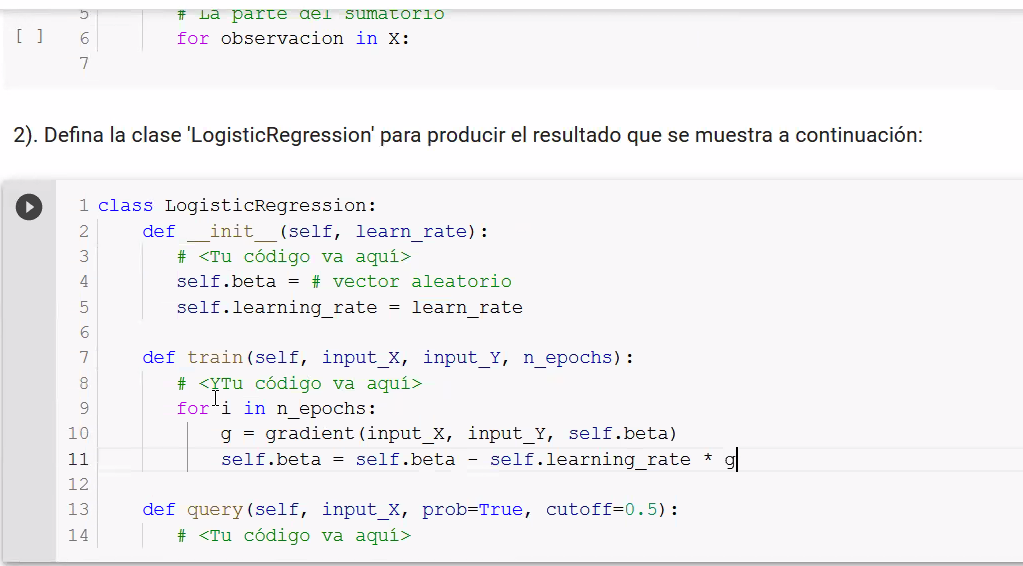
# Ejemplos de entramiento







La mejor opción seria de orden 3



# Bibliography

[Cognitive neural networks deep dive](https://developer.ibm.com/articles/cc-cognitive-neural-networks-deep-dive/)

[Models machine learning](https://developer.ibm.com/articles/cc-models-machine-learning/)

# References

1. Insight: profound perception. [↑](#endnote-ref-1)
2. Subset: a part of a larger group of related things. [↑](#endnote-ref-2)