



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

# APRENDIZAJE DE MÁQUINAS

**JOHN W. BRANCH**

Profesor Titular

Departamento de Ciencias de la Computación y de la Decisión

Director del Grupo de I+D en Inteligencia Artificial – GIDIA

[jwbranch@unal.edu.co](mailto:jwbranch@unal.edu.co)

<https://github.com/jwbranch/AprendizajeDeMaquinas>

<https://www.coursera.org/programs/unal-iuukt>

# METODOLOGÍA ENSEÑANZA – APRENDIZAJE

El aprendizaje sincrónico involucra estudios online a través de una plataforma. Este tipo de aprendizaje sólo ocurre en línea. Al estar en línea, el estudiante se mantiene en contacto con el docente y con sus compañeros. Se llama aprendizaje sincrónico porque la plataforma permite que los estudiantes pregunten al docente o compañeros de manera instantánea a través de herramientas como el chat o el video chat.

## Sesiones Remotas vía Google.Meet Sincrónicas y Asincrónicas

El aprendizaje asincrónico puede ser llevado a cabo online u offline. El aprendizaje asincrónico implica un trabajo de curso proporcionado a través de la plataforma o el correo electrónico para que el estudiante desarrolle, de acuerdo a las orientaciones del docente, de forma independiente. Un beneficio que tiene el aprendizaje asincrónico es que el estudiante puede ir a su propio ritmo.

# Descripción del Curso

El curso introduce los conceptos fundamentales y los métodos más utilizados en el campo del aprendizaje de máquinas enfocados desde las perspectivas de la naturaleza del problema que se requiere resolver, esto es, aprendizaje supervisado orientado a los problemas de clasificación y regresión para aplicaciones de predicción o pronóstico. Aprendizaje no supervisado orientado a tareas de agrupar o etiquetar un conjunto de datos, También se incluyen la aproximación general de técnicas modernas de aprendizaje tales como el aprendizaje por refuerzo y aprendizaje profundo.

# Contenido

1. Introducción.
2. Los datos en Aprendizaje de Máquinas.
3. **Aprendizaje Supervisado.**
4. Aprendizaje NO Supervisado.
5. Aprendizaje por Refuerzo.
6. Aprendizaje con Clases Desbalanceadas y Combinación de Modelos.
7. Aplicaciones y Casos de Éxito.

# Bibliografía Recomendada

**Osvaldo Simeone (2018)**, “A Brief Introduction to Machine Learning for Engineers”, Foundations and TrendsR in Signal Processing: Vol. 12, No. 3-4, pp 200–431. DOI: 10.1561/20000000102.

**Goodfellow, I., Bengio, Y. y Courville, A. (2016)** Deep Learning, MIT Press.

**Murphy, K. (2012)**. Machine Learning: A Probabilistic Perspective, MIT Press .

**Hastie, T., Tibshirani, R. y Friedman, J. (2011)**. The Elements of Statistical Learning. Springer.  
(Available for download on the authors' web-page.)

**Szepesvári, C. (2010)**. Algorithms for Reinforcement Learning. Morgan and Claypool.

**Haykin, S. (2008)**. Neural Networks and Learning Machines. Pearson.

**Sutton, R. y Barto, A. (1998)**. Reinforcement Learning: An Introduction. MIT Press.

**Mitchell, T. M. (1997)**. Machine Learning. 1st. McGraw-Hill Higher Education. (Chapter 1)

# EVALUACIÓN



## Certificado Coursera

*Sesenta años de inteligencia artificial – UNAM (Obligatorio)*

*IA para todos – Andrew Ng (Obligatorio)*

*Structuring Machine Learning Projects – Andrew Ng (Obligatorio)*

Machine Learning - University of Washington → **Curso #1: Machine Learning Foundations: A Case Study Approach** (Obligatorio)

## Informe de Lectura (Individual)

Machine Learning Algorithms: A Review

Machine Learning aspects and its Applications Towards Different Research Areas

## Trabajo Final (Debe ser en Grupo 3 ó 5 personas)

*Obtener el conjunto de datos (texto o audio o video o imagen) de los siguientes repositorios o cualquier otro disponible:*

<http://www.ics.uci.edu/~mllearn/databases/>

<https://www.kaggle.com/datasets>

*Origen, atributos, clases*

*"Scatter plot" de los datos*

*Visualización del conjunto en 2D (PCA o MDS)*

*Seleccionar un método de Entrenamiento y Evaluar el Desempeño.*

**20%**

**(Máx. 31 de Dic/2020)**

**20%**

**(Máx. 21 de Nov/2020)**

**60%**

**(Máx. 21 de Dic/2020)**

# APRENDIZAJE DE MÁQUINAS

## “Aprendizaje Supervisado”

**JOHN W. BRANCH**

Profesor Titular

Departamento de Ciencias de la Computación y de la Decisión

Director del Grupo de I+D en Inteligencia Artificial – GIDIA

[jwbranch@unal.edu.co](mailto:jwbranch@unal.edu.co)

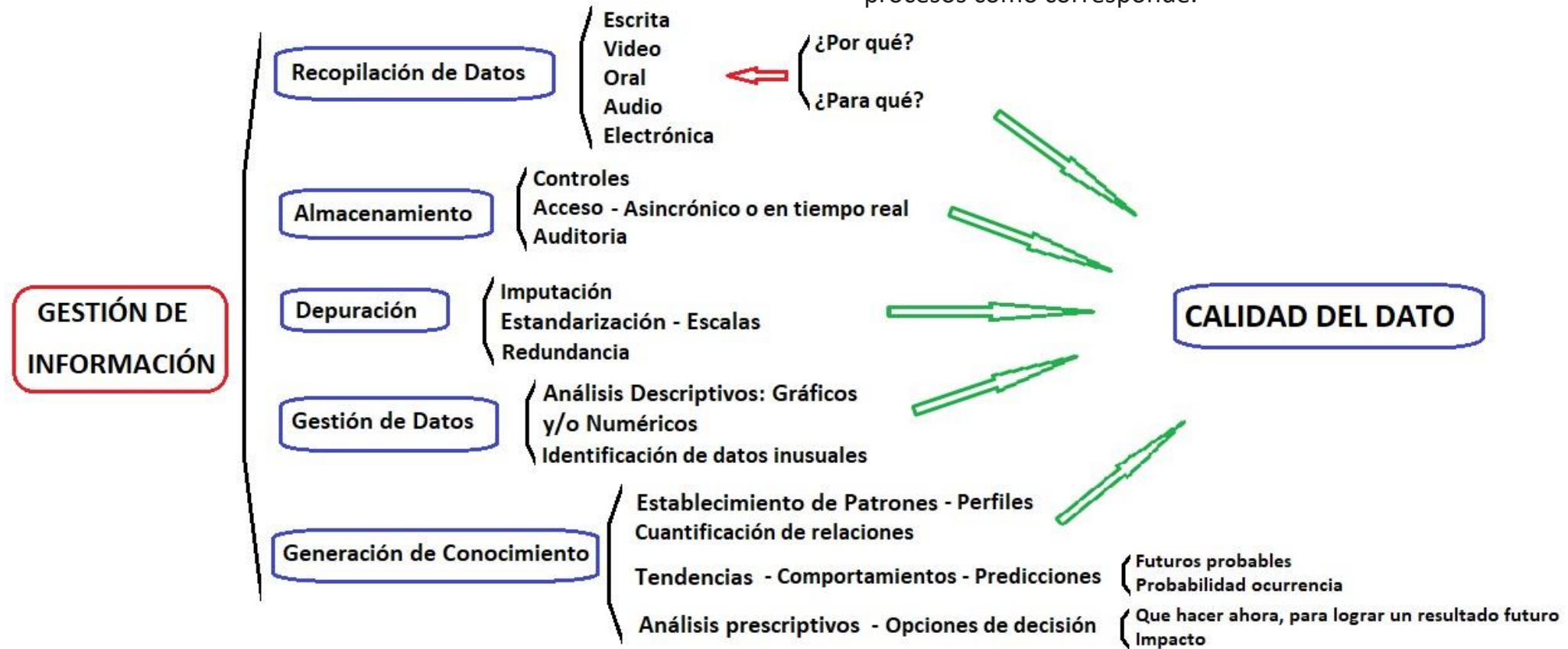
<https://github.com/jwbranch/AprendizajeDeMaquinas>

<https://www.coursera.org/programs/unal-iuukt>



# Cultura y Calidad del Dato

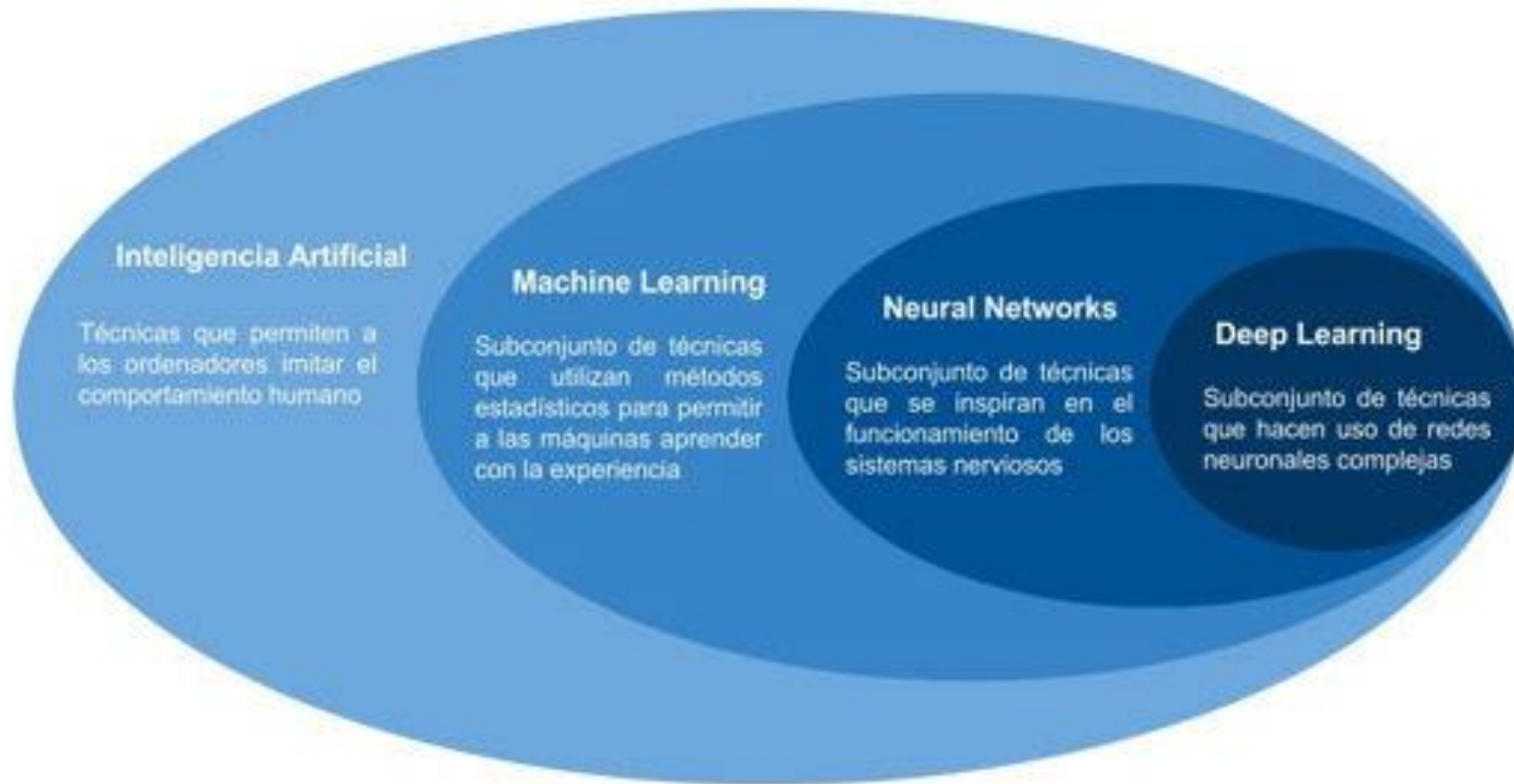
Un estudio realizado por Harvard Business Review, menos del 50% de las decisiones estructuradas de las empresas se basan en datos. Este resultado refleja la importancia de implementar una nueva cultura de datos y, sobre todo, llevar los procesos como corresponde.



*"...las organizaciones basadas en datos tienen 23 veces más probabilidades de adquirir clientes, seis veces más probabilidades de retener a esos clientes y 19 veces más probabilidades de ser rentables."*

McKinsey Global Institute

# Aprendizaje vs Inteligencia



**Las máquinas pueden:**

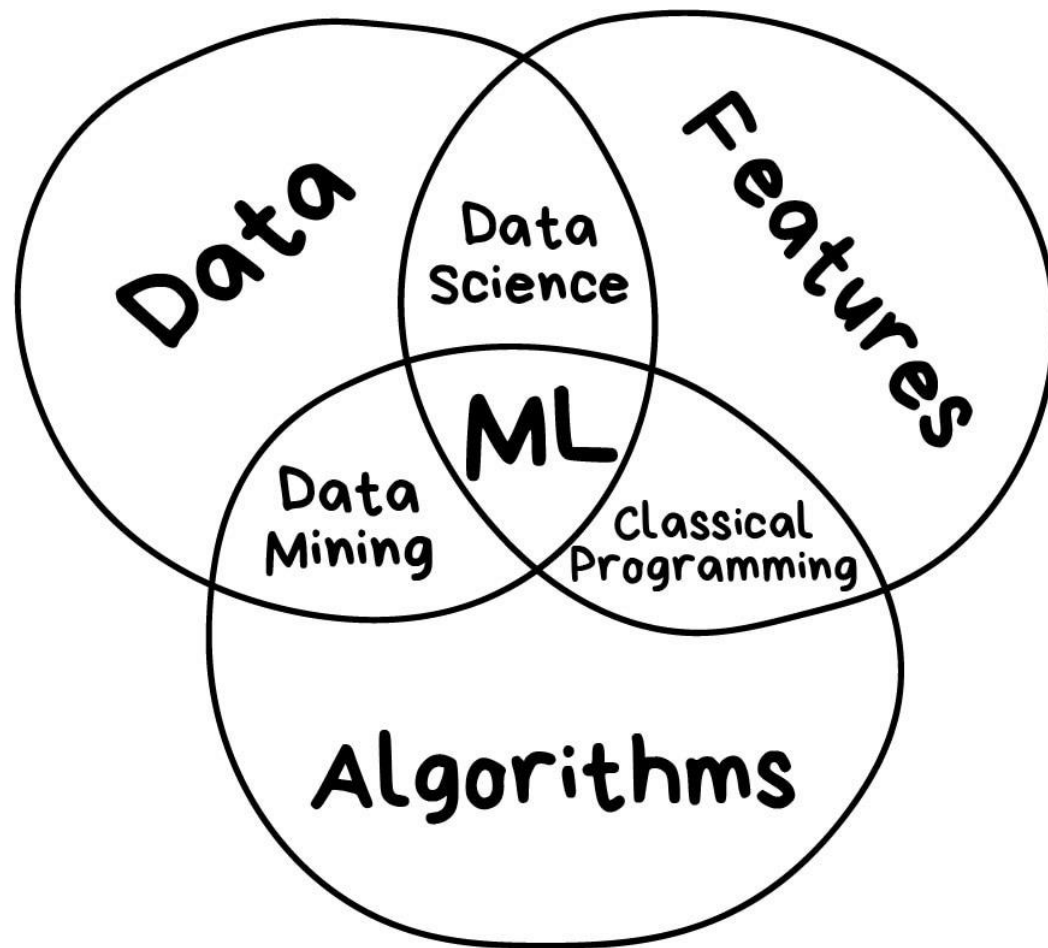
**Hacer Pronósticos**

**Memorizar**

**Reproducir**

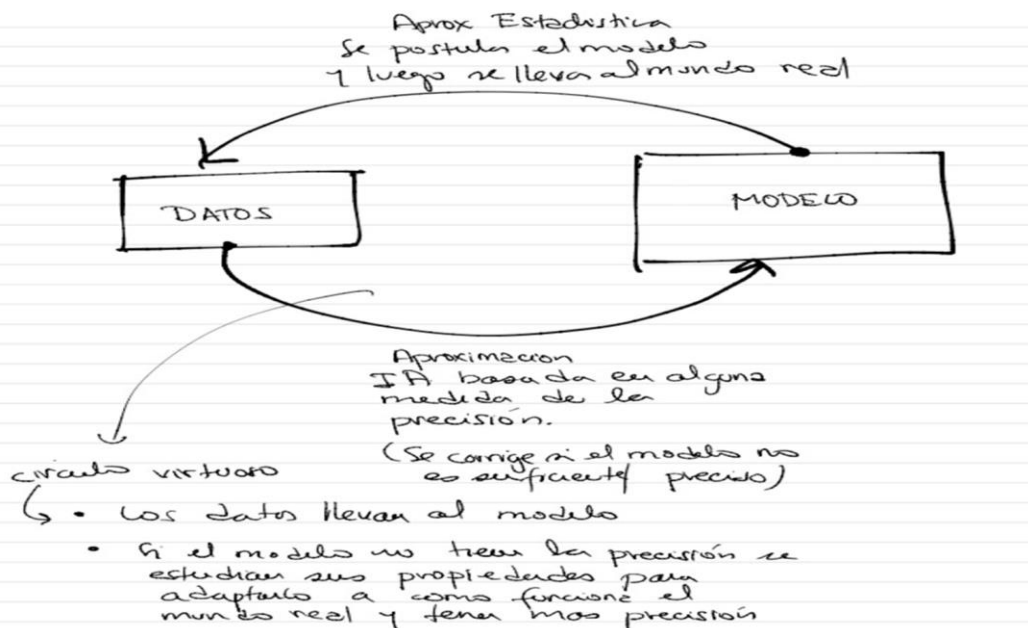
**Elegir el mejor artículo**

## Lo tres componentes del aprendizaje de máquinas

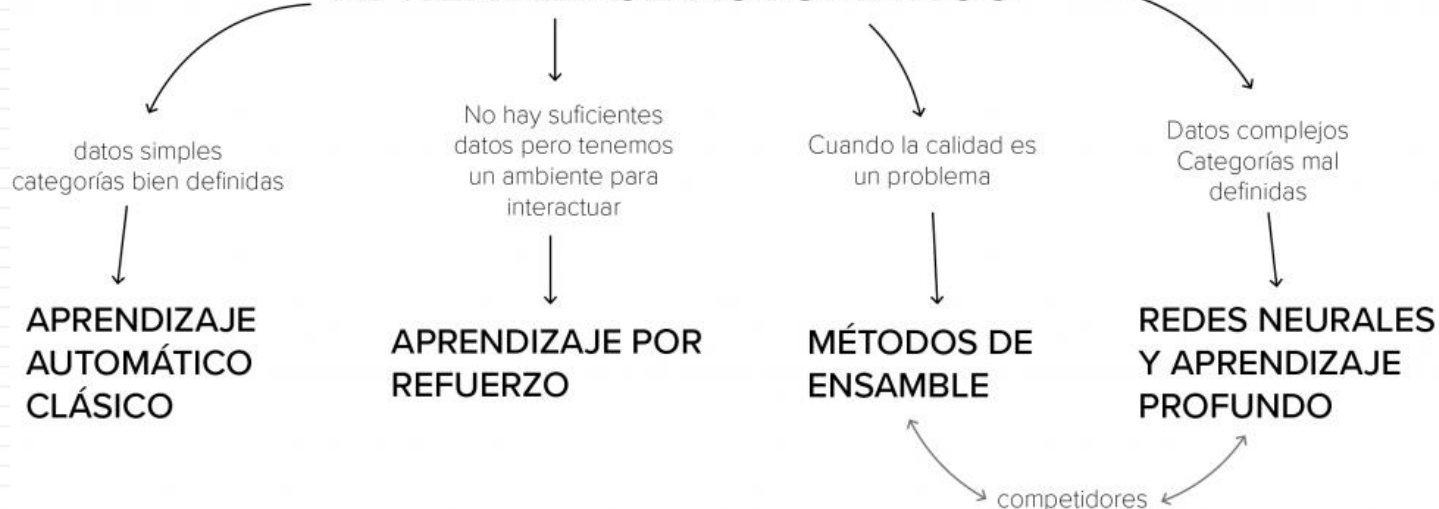


# Tipos de Aprendizaje de Máquinas

No existe una única forma de resolver un problema en el mundo del aprendizaje de máquinas. Siempre hay varios algoritmos que se ajustan, y la habilidad del científico de datos está en elegir cuál se adapta mejor.



## TIPOS PRINCIPALES DE APRENDIZAJE AUTOMÁTICO



# APRENDIZAJE AUTOMÁTICO CLÁSICO





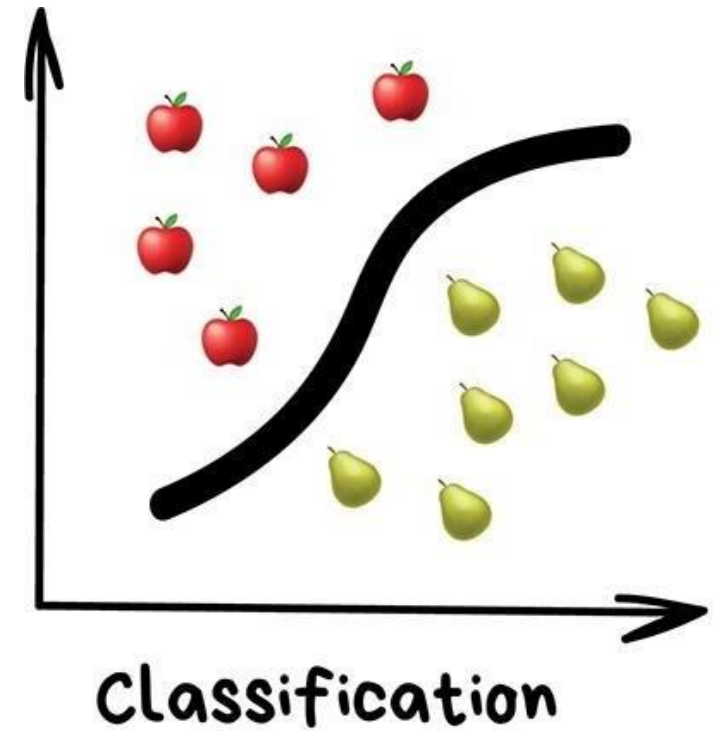
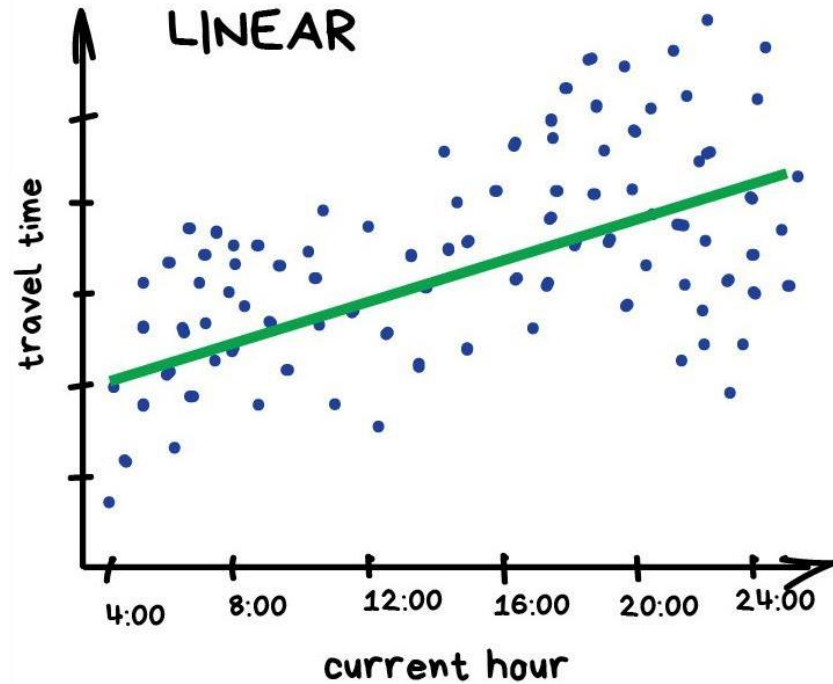


# Aprendizaje Supervisado

**Aprendizaje Supervisado:** requiere de un conjunto de datos conocidos a partir del cual se crea un **modelo** para predecir el valor de una variable de salida. El aprendizaje supervisado se puede usar en dos tareas:

- **Regresión:** en este caso los algoritmos de aprendizaje buscan predecir el valor de una **variable continua** a partir de los datos de entrada. Un ejemplo de una tarea de regresión es el de estimar la longitud de un salmón en función de su edad y su peso.
- **Clasificación:** en este caso la variable de salida es una etiqueta que determina la clase a la que pertenecen los datos de entrada, es decir, la variable de salida es una **variable discreta**.

# Tipos de Aprendizaje





# Contenido

## 1. Regresión

- a. Regresión Lineal
- b. Regresión Polinomial
- c. Regresión Lasso, Ridge y Elastic-net
- d. Red Neuronal para Regresión

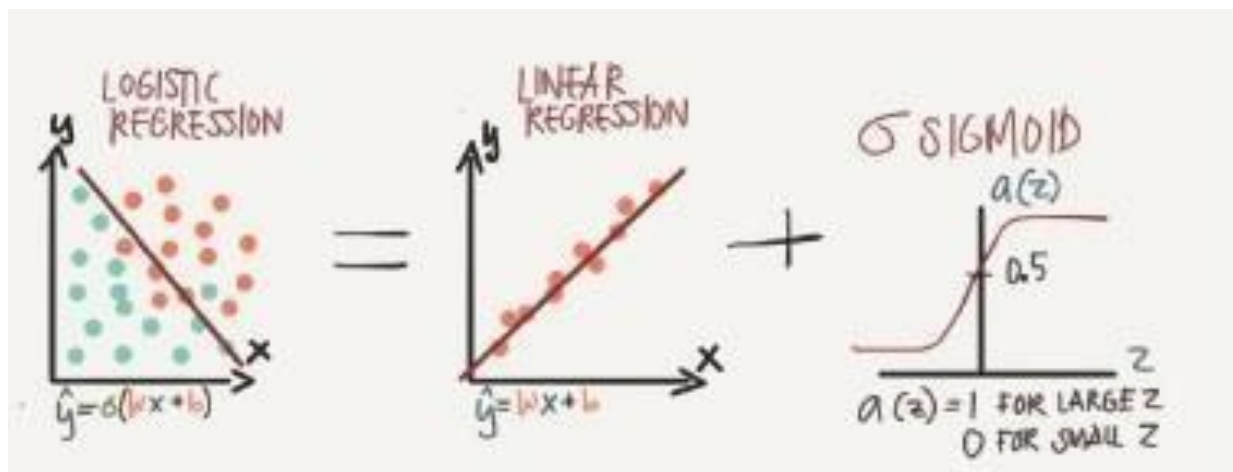
## 2. Clasificación

- a. Regresión Logística
- b. Árbol de Decisión
- c. K-nn
- d. Máquina de Vectores de Soporte
- e. Algoritmo XGBoost para Clasificación
- f. Red Neuronal para Clasificación

# Regresión Logística

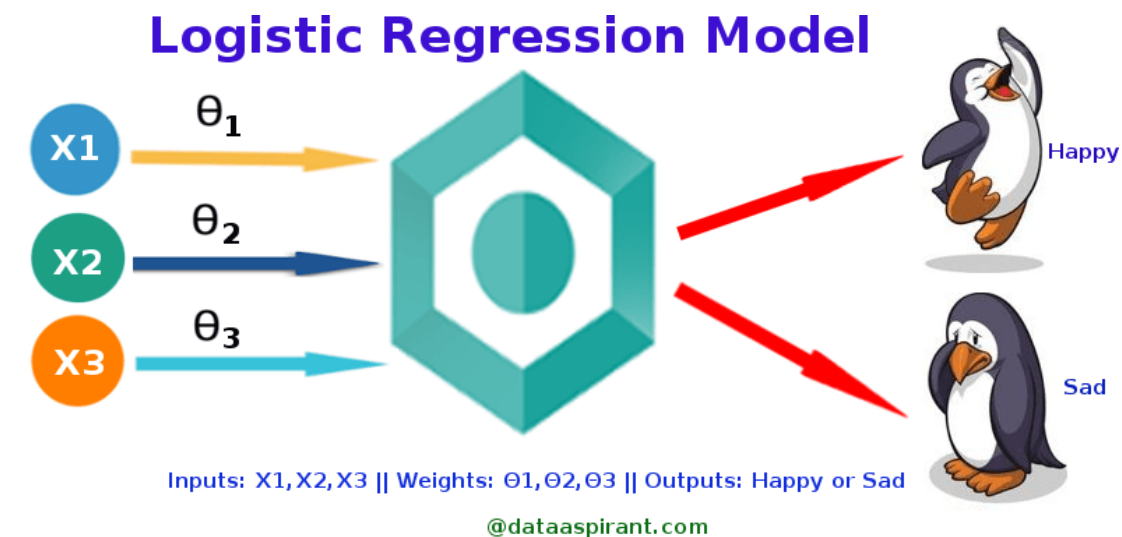
## DEFINICIÓN:

La regresión logística es un modelo de clasificación que se utiliza para **predecir la probabilidad**  $P(y = 1)$  **de una variable dependiente categórica** en función de  $x$ . Así, la variable  $y$  es una variable binaria codificada como 1 (positivo, éxito, etc.) o 0 (negativo, falla, etc.).



## Algunos ejemplos de aplicación:

- E-mail: spam/no spam
- Transacciones en línea: fraude/no fraude
- Tumores: maligno/no maligno



# Regresión Logística

## VENTAJAS:

- Es un modelo de clasificación eficiente y simple.
- No es necesario disponer de grandes recursos computacionales.
- Los resultados son altamente interpretables.



## DESVENTAJAS:

- Imposibilidad de resolver directamente problemas no lineales.
- La variable objetivo debe ser linealmente separable.
- La regresión logística no es uno de los algoritmos más potentes que existen.

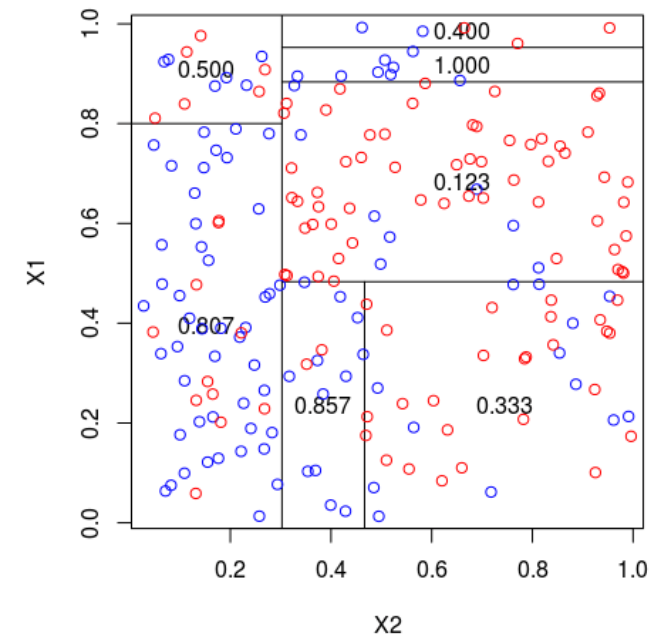
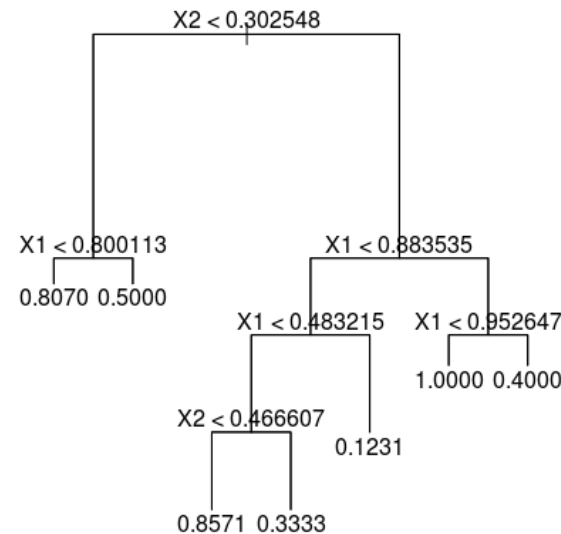


# Árbol de Decisión

Los Árboles de Decisión pueden resumirse con los siguientes puntos:

- Los árboles de decisión son modelos predictivos que utilizan un conjunto de reglas binarias para calcular un valor objetivo.
- Cada árbol individual es un modelo bastante simple que tiene ramas, nodos y hojas.

Un árbol de decisión **consisten en estimar un resultado haciendo una serie de preguntas a los datos**, cada pregunta estrechando nuestros posibles valores hasta que el modelo tenga la suficiente confianza como para hacer una sola predicción. El orden de la pregunta así como su contenido son determinados por el modelo. Además, las preguntas formuladas están todas en forma de Verdadero/Falso.



# Árbol de Decisión

Durante el entrenamiento, el modelo se ajusta con cualquier dato histórico que sea relevante para el dominio del problema y el verdadero valor que queremos que el modelo aprenda a predecir. El modelo aprende cualquier relación entre los datos y la variable objetivo.

Después de la fase de entrenamiento, el árbol de decisión produce un árbol calculando las mejores preguntas, así como su orden para hacer las estimaciones más precisas posibles. Cuando se quiere hacer una predicción se debe proporcionar al modelo el mismo formato de datos para hacer una predicción. La predicción será una estimación basada en los datos del entrenamiento en el que se ha entrenado.

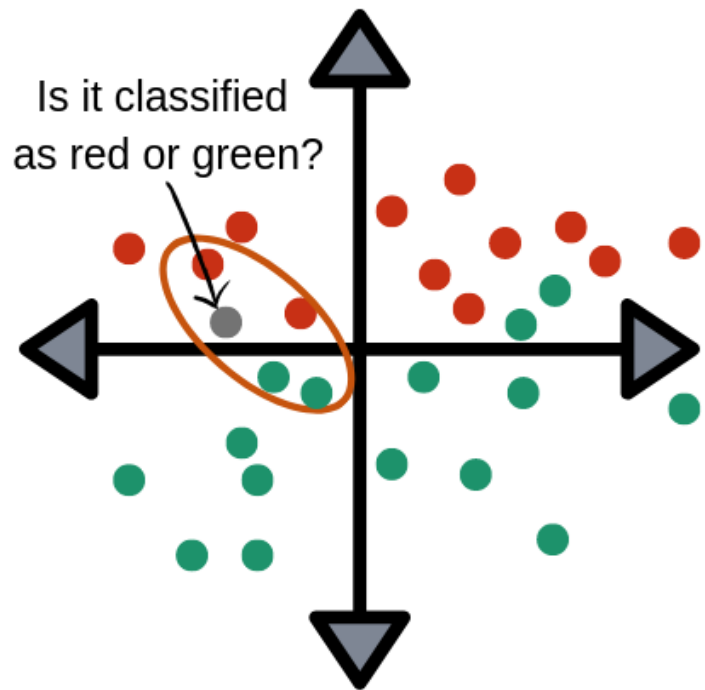
## DECISION TREE REGRESSION

Similar to decision tree classification, however uses Mean Squared Error or similar metrics instead of cross-entropy or Gini impurity to determine splits.



# Clasificador KNN

La idea básica sobre la que se fundamenta este paradigma es que un nuevo caso se va a clasificar en la clase más frecuente a la que pertenecen sus  $K$  vecinos más cercanos



## COMIENZO

Entrada:  $D = \{(\mathbf{x}_1, c_1), \dots, (\mathbf{x}_N, c_N)\}$

$\mathbf{x} = (x_1, \dots, x_n)$  nuevo caso a clasificar

PARA todo objeto ya clasificado  $(x_i, c_i)$

calcular  $d_i = d(\mathbf{x}_i, \mathbf{x})$

Ordenar  $d_i (i = 1, \dots, N)$  en orden ascendente

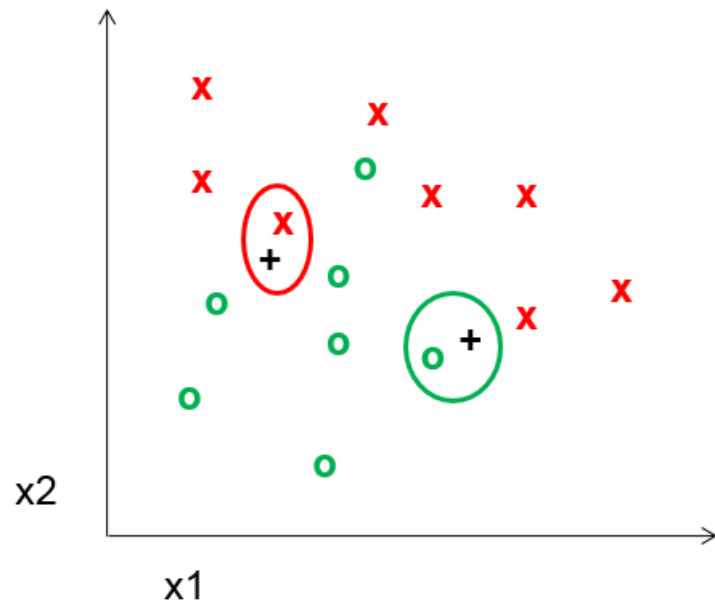
Quedarnos con los  $K$  casos  $D_{\mathbf{x}}^K$  ya clasificados más cercanos a  $\mathbf{x}$

Asignar a  $\mathbf{x}$  la clase más frecuente en  $D_{\mathbf{x}}^K$

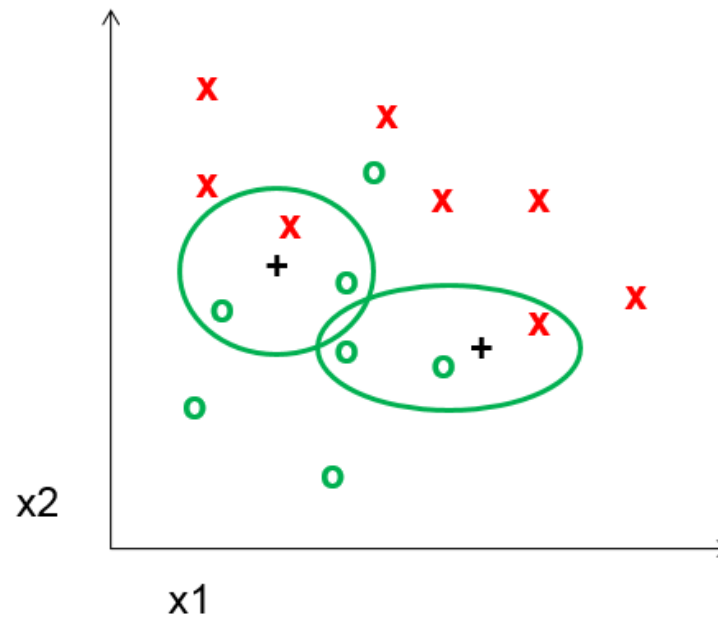
FIN

# Clasificador KNN

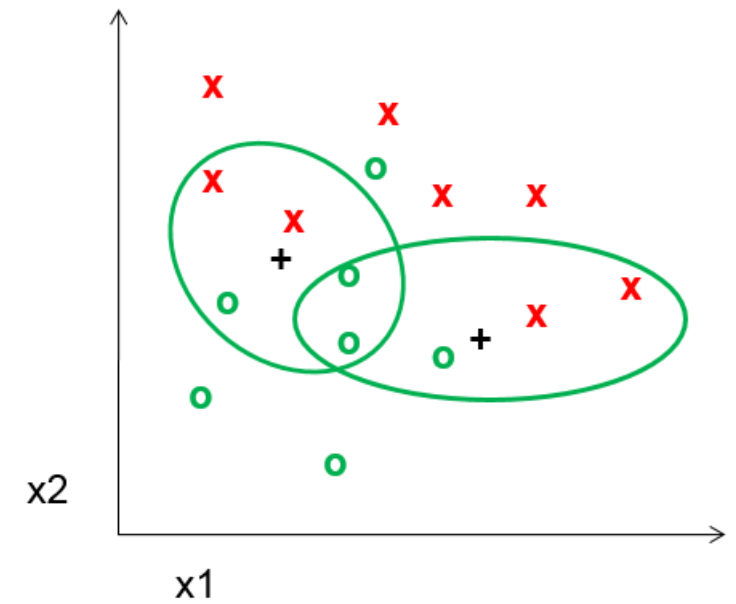
KNN-1 (1 Vecino más cercano)



KNN-3 (3 Vecinos más cercanos)



KNN-5 (5 Vecinos más cercanos)

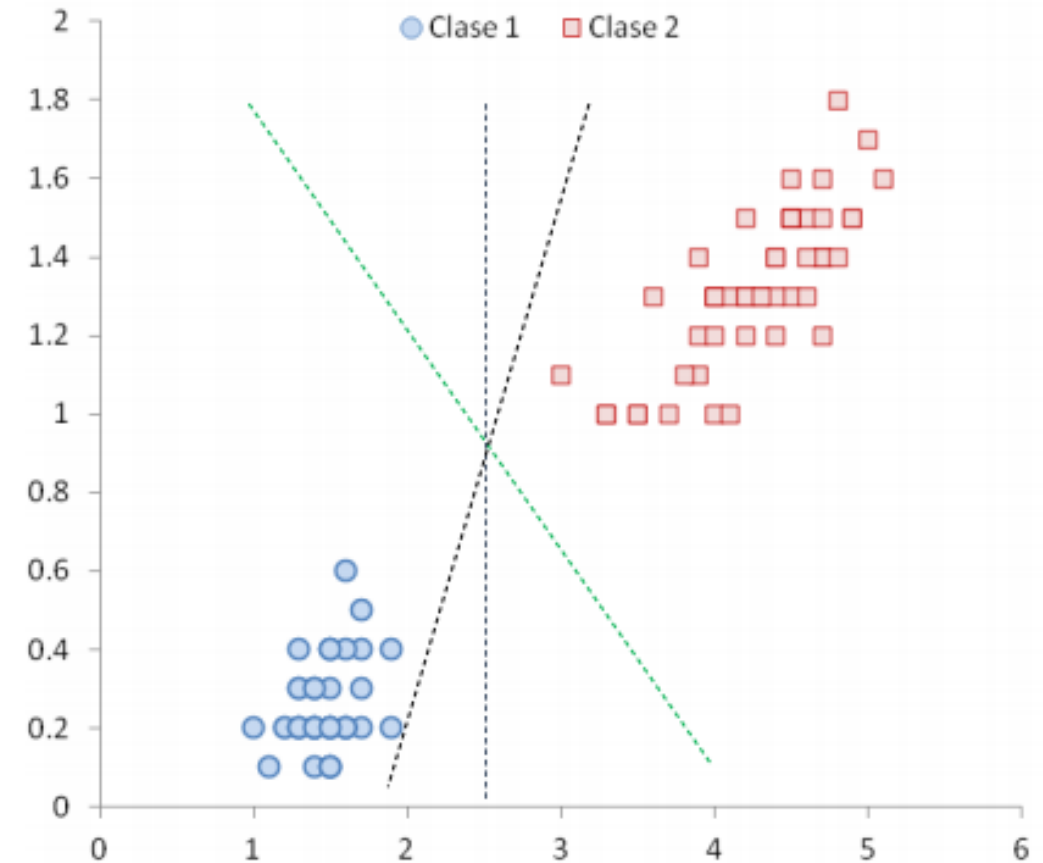


# Máquinas de Vectores de Soporte

Las Máquinas de Vectores de Soporte (SVM) buscan el hiperplano que separe de forma óptima los puntos de una clase de otra.

En un problema de dos clases, linealmente separables, pueden existir muchas fronteras de decisión (o hiperplanos) que pueden separar las clases.

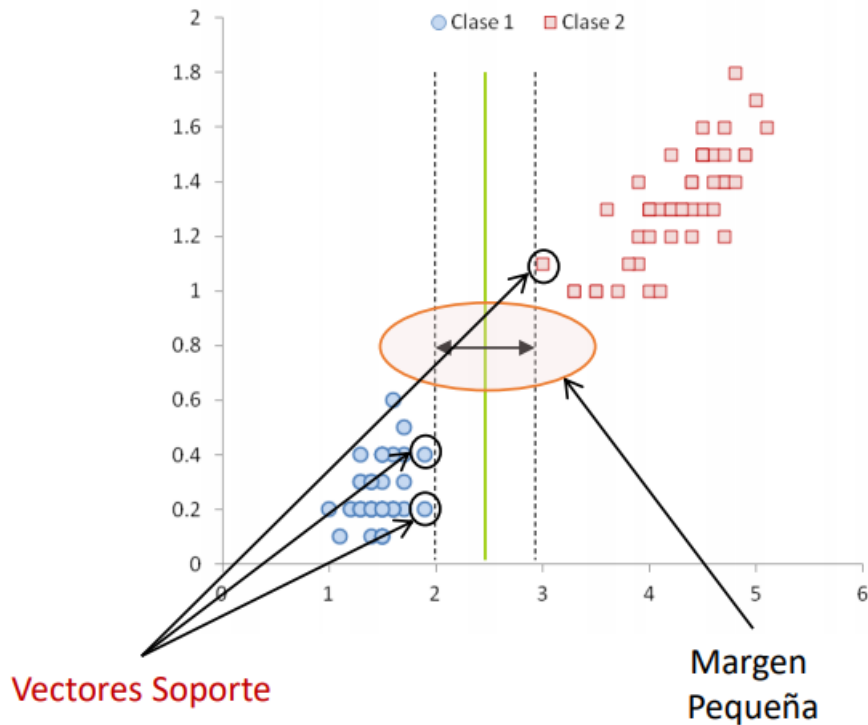
Sin embargo, ¿Son todas esas fronteras igual de buenas?



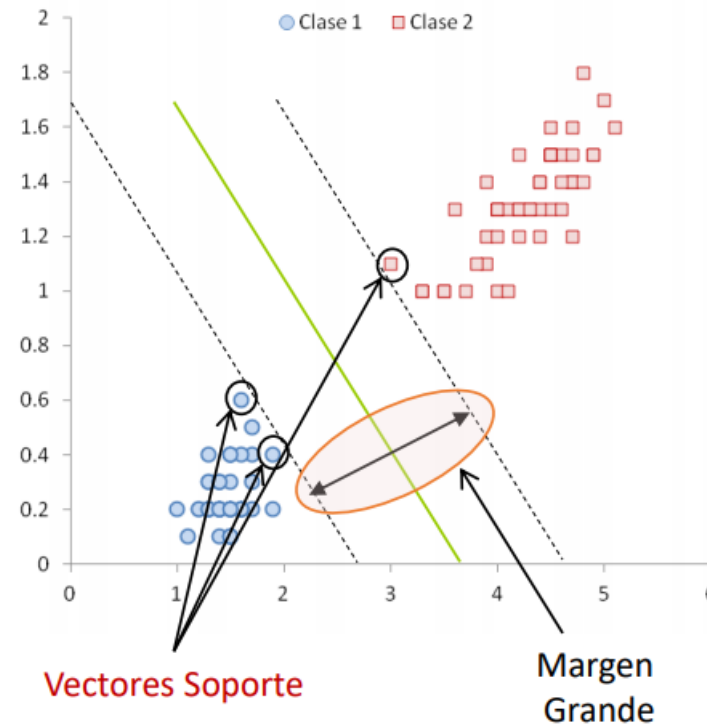


# Máquinas de Vectores de Soporte

La SVM busca el hiperplano que maximiza la distancia (o margen) con los puntos que estén más cerca de él, razón por la cual también se les conoce a las SVM como clasificadores de margen máximo.



Como el hiperplano separa las muestras positivas (+1) de las negativas (-1), los puntos que están en el hiperplano deben satisfacer la ecuación:  $\mathbf{w}^T \mathbf{x} + b = 0$ .



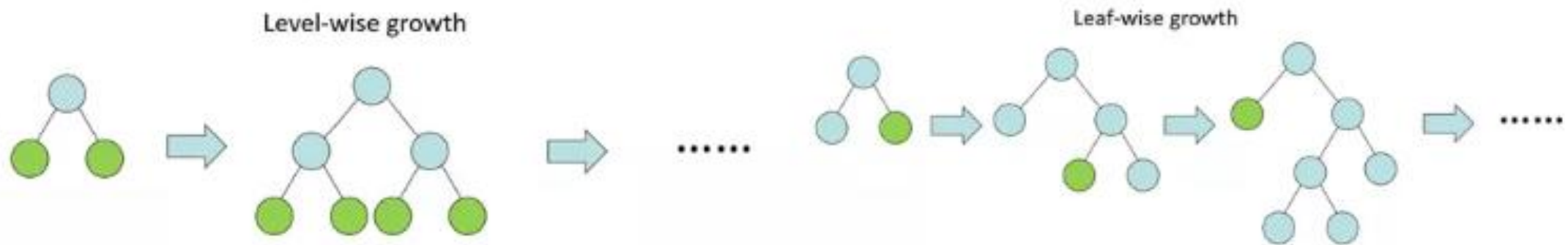
# XGBoost: eXtreme Gradient Boosting

Links de interés:

- <https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/>

Xgboost es un algoritmo de ensamble, tipo boosting, de árboles de decisión. **En un ensamble boosting los árboles son construidos de manera secuencial**, por lo que cada árbol siguiente reduce los errores de los árboles previos. Cada árbol aprende de sus predecesores y actualiza los errores.

Xgboost es uno de los métodos más populares de modelado de bases de datos tabulares de cualquier tamaño, es muy rápido y escalable.



*An illustration demonstrating the difference between level-wise and leaf-wise growth*

# Red Neuronal para Clasificación

*“El objetivo del entrenamiento de una RNA es conseguir que una aplicación determinada, para un conjunto de entradas produzca el conjunto de salidas deseadas o mínimamente consistentes”*

***El conjunto de Aprendizaje debe ser:***

*Significativo: Número suficiente de ejemplos.*

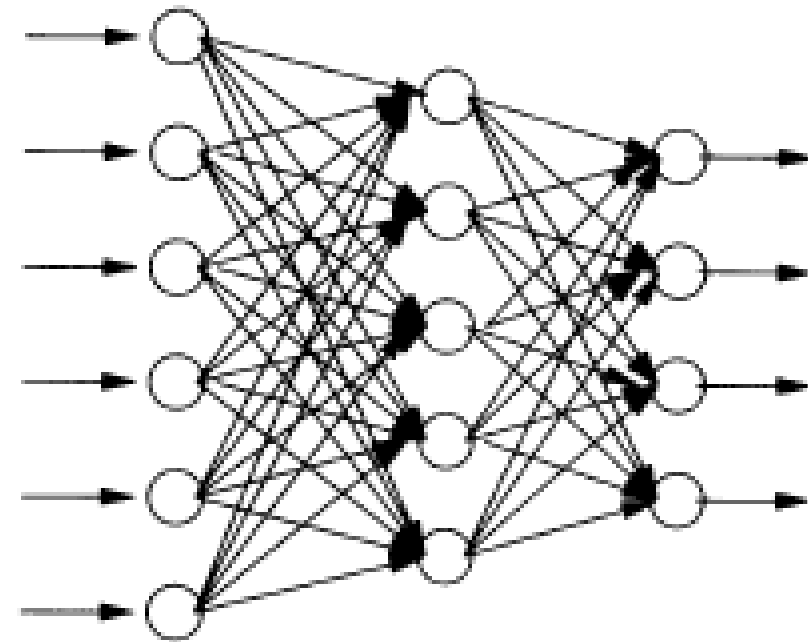
*Representativo: Conjunto de aprendizaje diverso.*

***La convergencia en el entrenamiento se da por:***

*Número fijo de ciclos.*

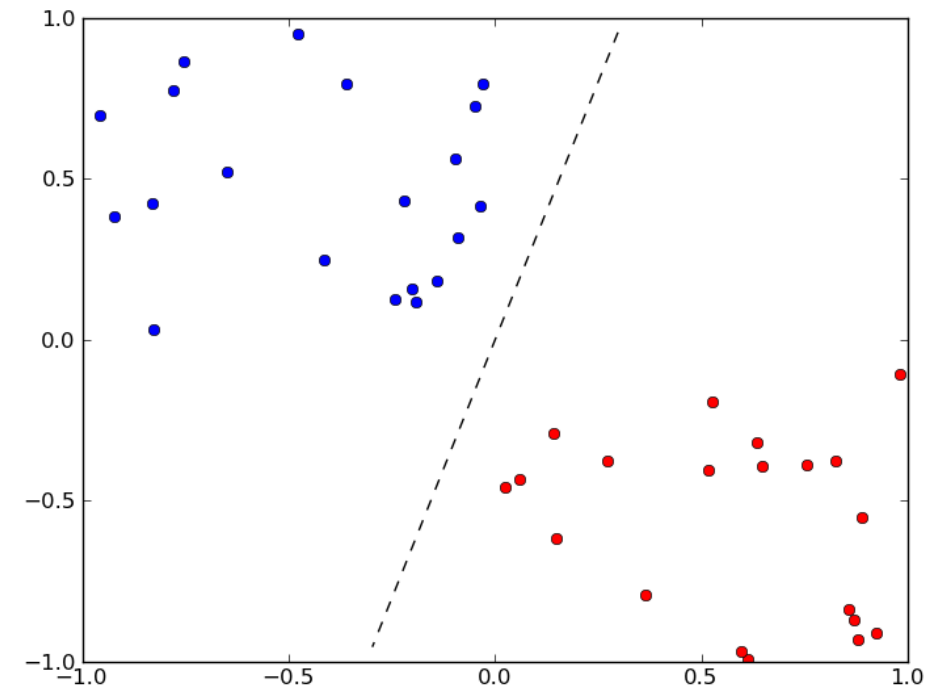
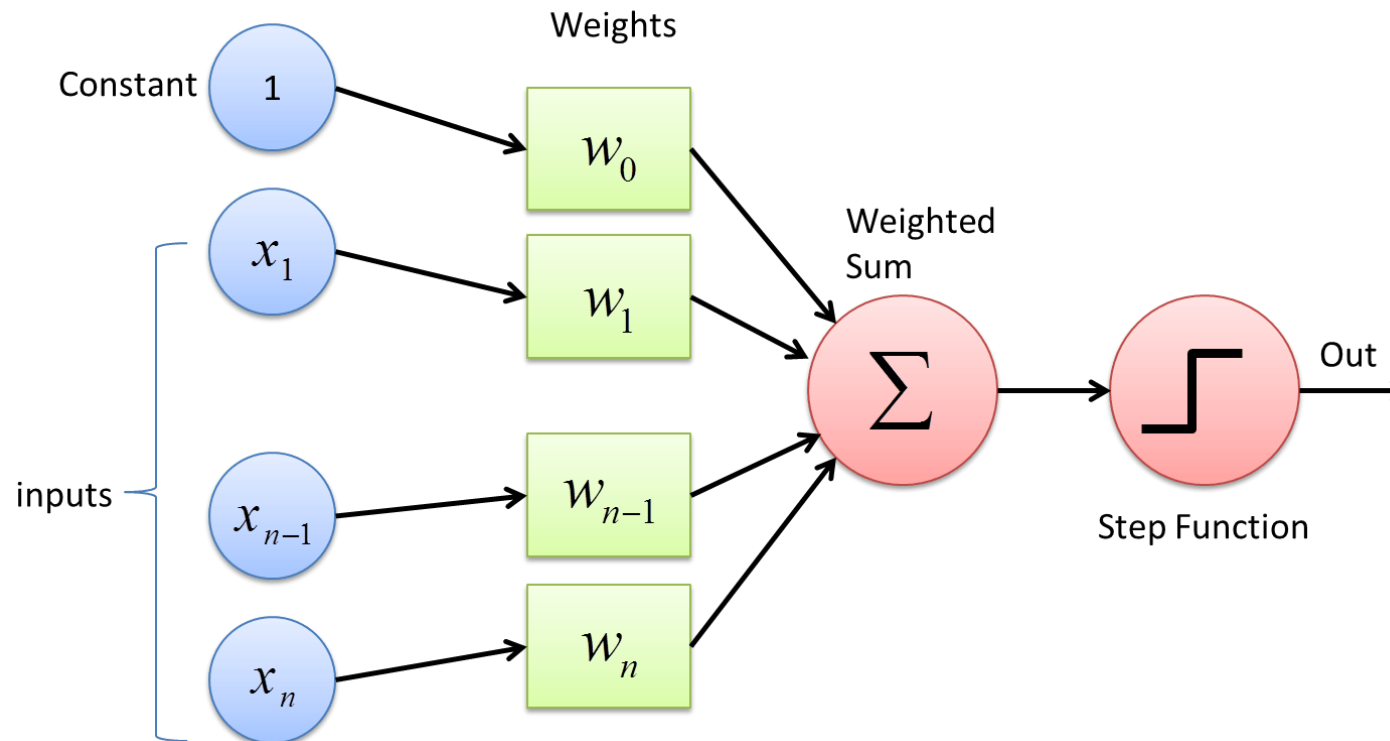
*Error descienda de cierta cantidad.*

*Modificación de los pesos sea irrelevante.*



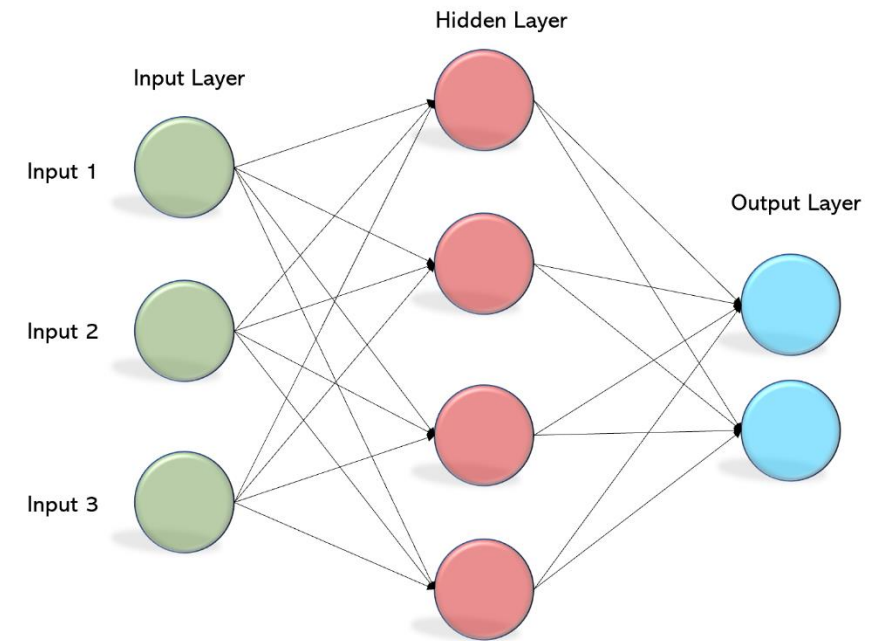
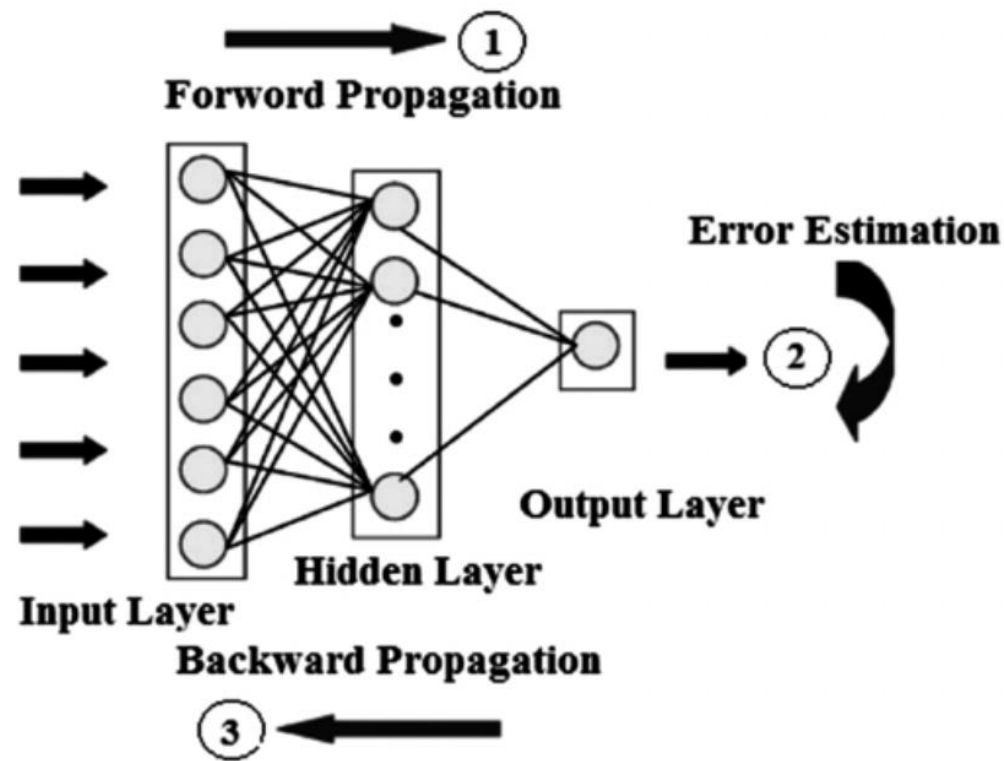
# Red Neuronal para Clasificación

Un perceptron (la red neuronal artificial más simple) permite resolver problemas linealmente separables. Este funciona multiplicando cada valor de entrada por un peso (que aprende la red) y sumado para una sola neurona. Por último, el valor obtenido es pasado por una función de activación para poder dividir el valor resultante en dos conjuntos.



# Red Neuronal para Clasificación

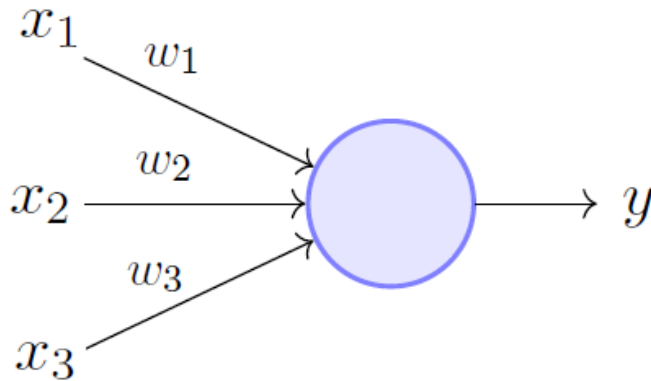
Un multi-perceptron (una red densa o totalmente conectada) permite resolver problemas linealmente NO separables. Esta es la ventaja que tiene contra el perceptrón simple. Como contiene más neuronas y capas ocultas, es capaz de mapear funciones más complejas a través de los pesos aprendidos con las épocas de entrenamiento. Lo que vuelve a una red neuronal compleja poder resolver problemas linealmente NO separables, son las funciones de activación no lineales.



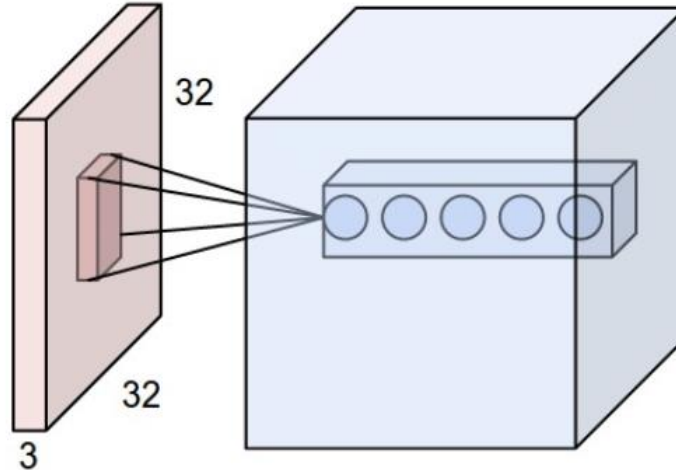
# Tipos de neuronas más utilizadas

Existen un alto número de tipos de neuronas diseñados para las redes neuronales. Entre las más comunes tenemos:

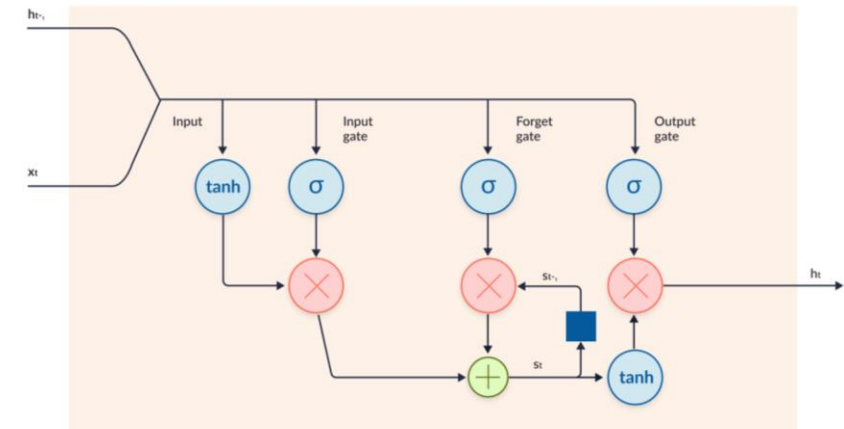
Perceptron



Convolutacional

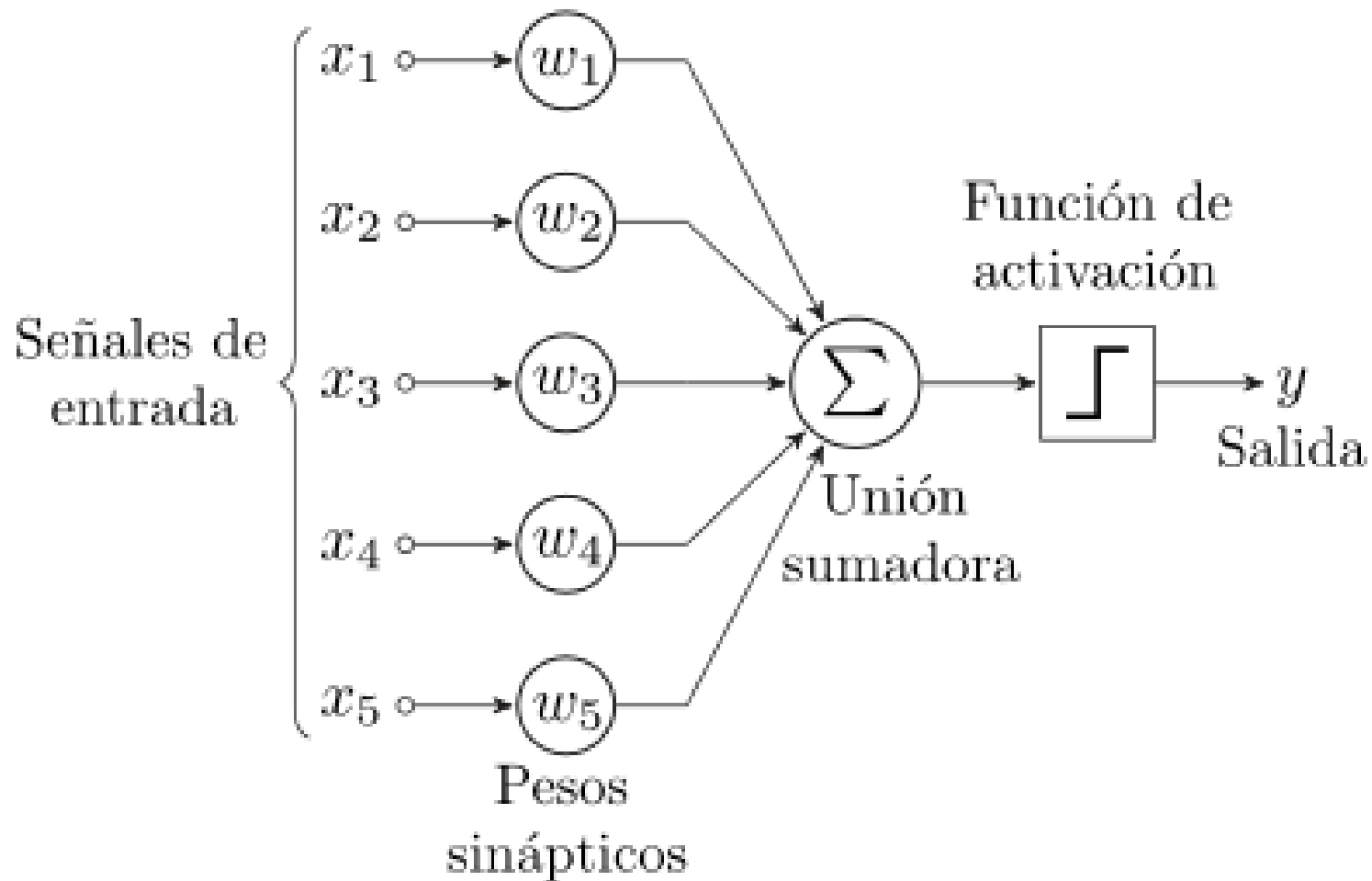


Recurrente



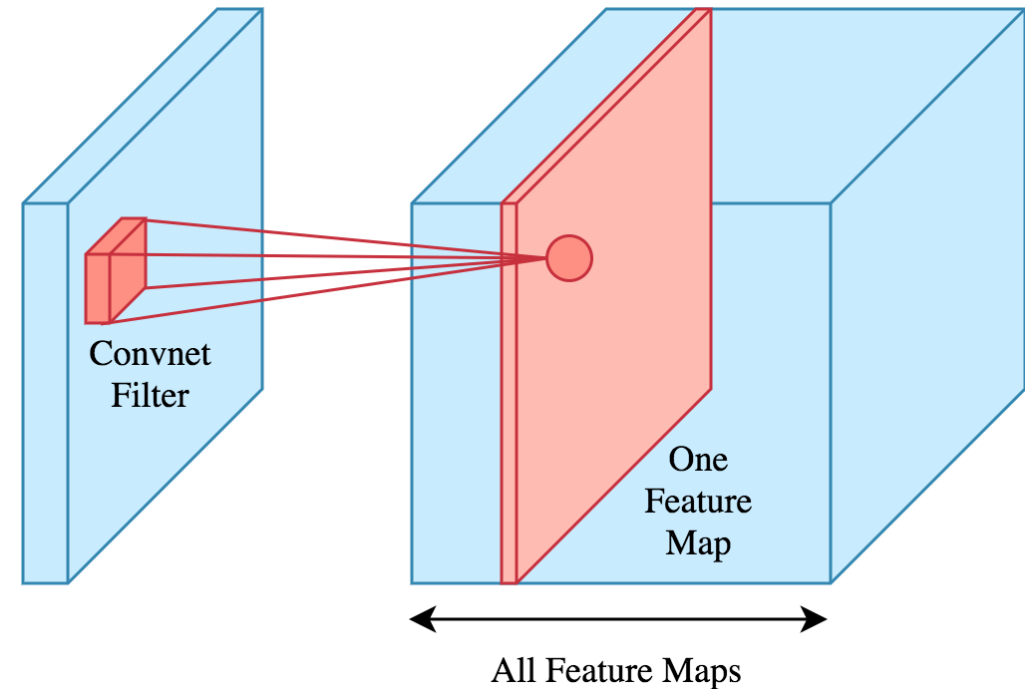
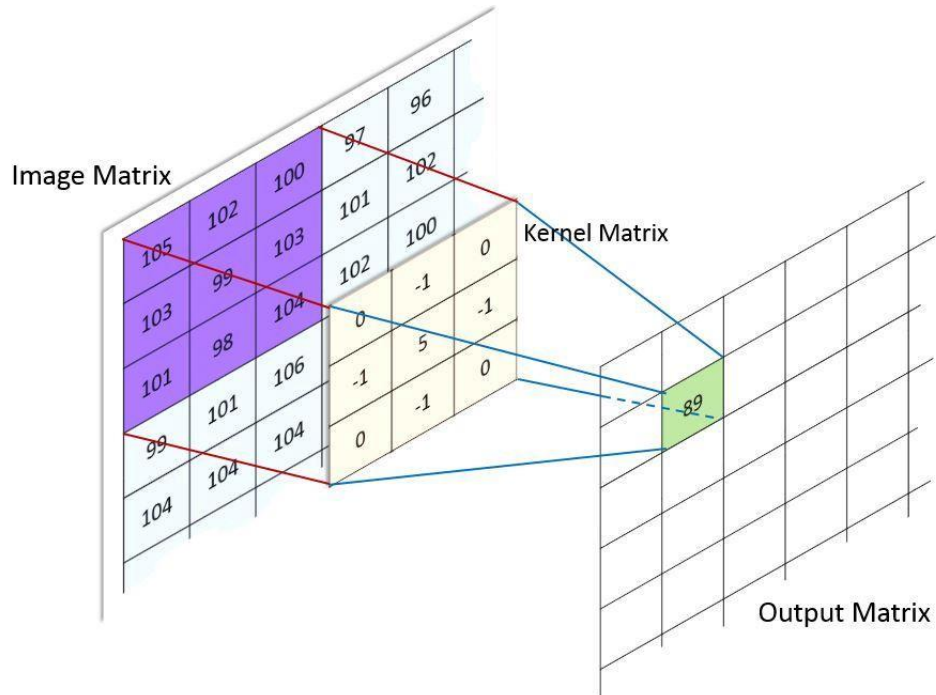
# Perceptron

Es la neurona tradicional y se calcula su valor multiplicando todas las entradas por sus pesos de conexión y sumados todos en un solo valor. Sus entradas son de 1 dimensión.



# Convolutacional

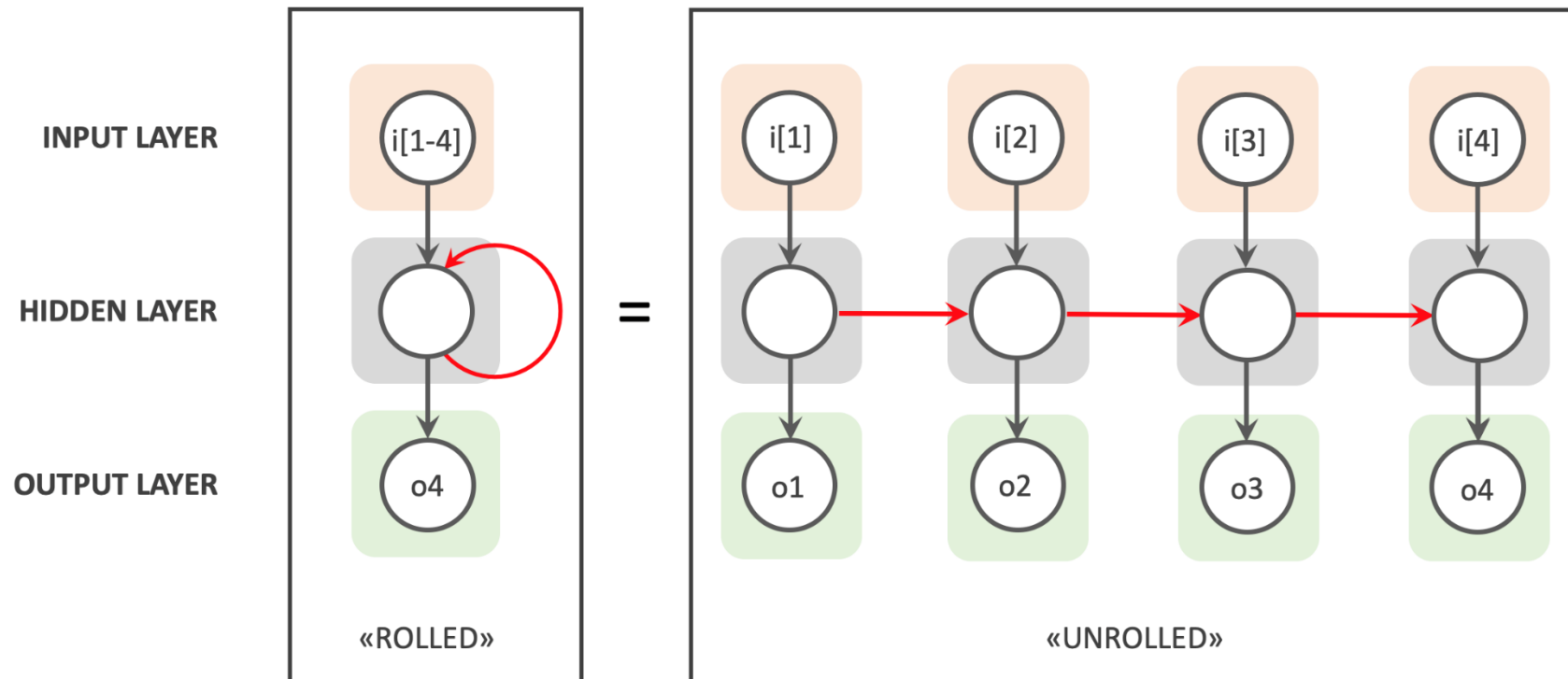
Es la neurona especial que recibe como entrada un valor de dos o tres dimensiones (una imagen). Esta es capaz de explotar la relación espacial que los datos de entrada contienen. Dependiendo de sus parámetros, el resultado de la capa es otra imagen (datos de dos o tres dimensiones) en otra distribución de datos. De una manera más directa, esta neurona es capaz de aprender “filtros” para las entradas.














# Recurrentes

Es la neurona especial que recibe como entrada un valor de dos dimensiones (una señal en el tiempo). Esta es capaz de explotar la relación temporal que los datos de entrada contienen. La salida de esta neurona, dependiendo de los parámetros, puede ser una única salida al final o puede ser una salida en cada tiempo (dato) de entrada.

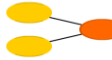


# A mostly complete chart of Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org

-  Backfed Input Cell
-  Input Cell
-  Noisy Input Cell
-  Hidden Cell
-  Probabilistic Hidden Cell
-  Spiking Hidden Cell
-  Output Cell
-  Match Input Output Cell
-  Recurrent Cell
-  Memory Cell
-  Different Memory Cell
-  Kernel
-  Convolution or Pool

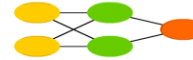
Perceptron (P)



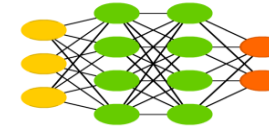
Feed Forward (FF)



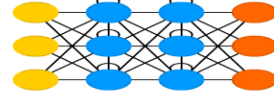
Radial Basis Network (RBF)



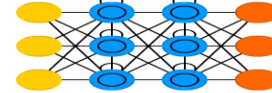
Deep Feed Forward (DFF)



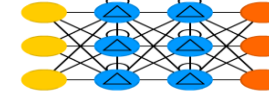
Recurrent Neural Network (RNN)



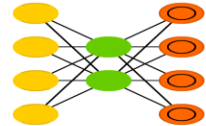
Long / Short Term Memory (LSTM)



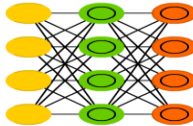
Gated Recurrent Unit (GRU)



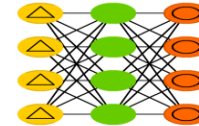
Auto Encoder (AE)



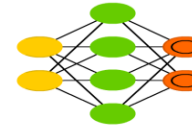
Variational AE (VAE)



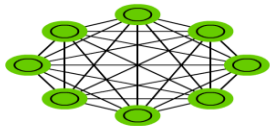
Denoising AE (DAE)



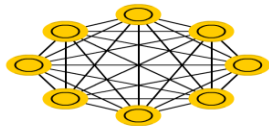
Sparse AE (SAE)



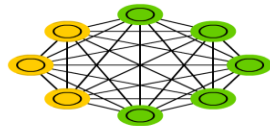
Markov Chain (MC)



Hopfield Network (HN)



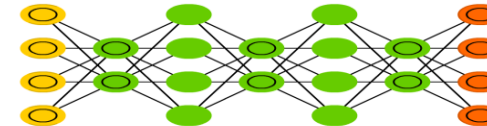
Boltzmann Machine (BM)



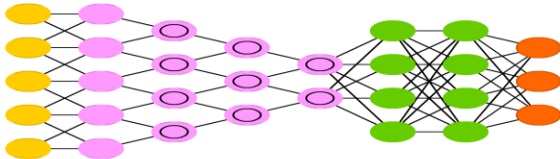
Restricted BM (RBM)



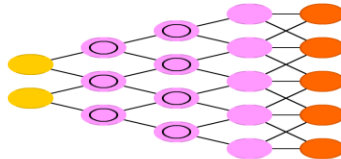
Deep Belief Network (DBN)



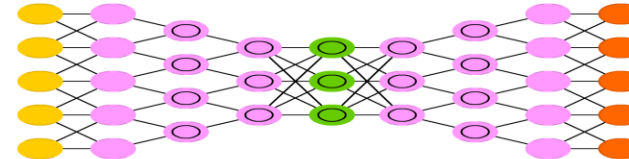
Deep Convolutional Network (DCN)



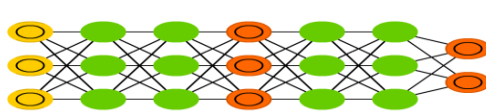
Deconvolutional Network (DN)



Deep Convolutional Inverse Graphics Network (DCIGN)



Generative Adversarial Network (GAN)



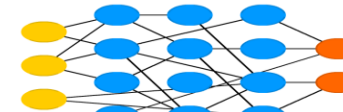
Liquid State Machine (LSM)



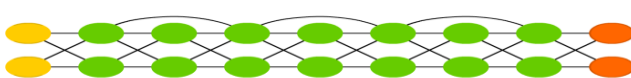
Extreme Learning Machine (ELM)



Echo State Network (ESN)



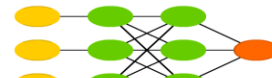
Deep Residual Network (DRN)



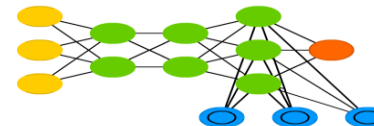
Kohonen Network (KN)



Support Vector Machine (SVM)



Neural Turing Machine (NTM)



## EJEMPLO

Predicción en la recuperación del oro con técnicas de aprendizaje de máquina

## Pórfidos Cupríferos (Veta de oro)



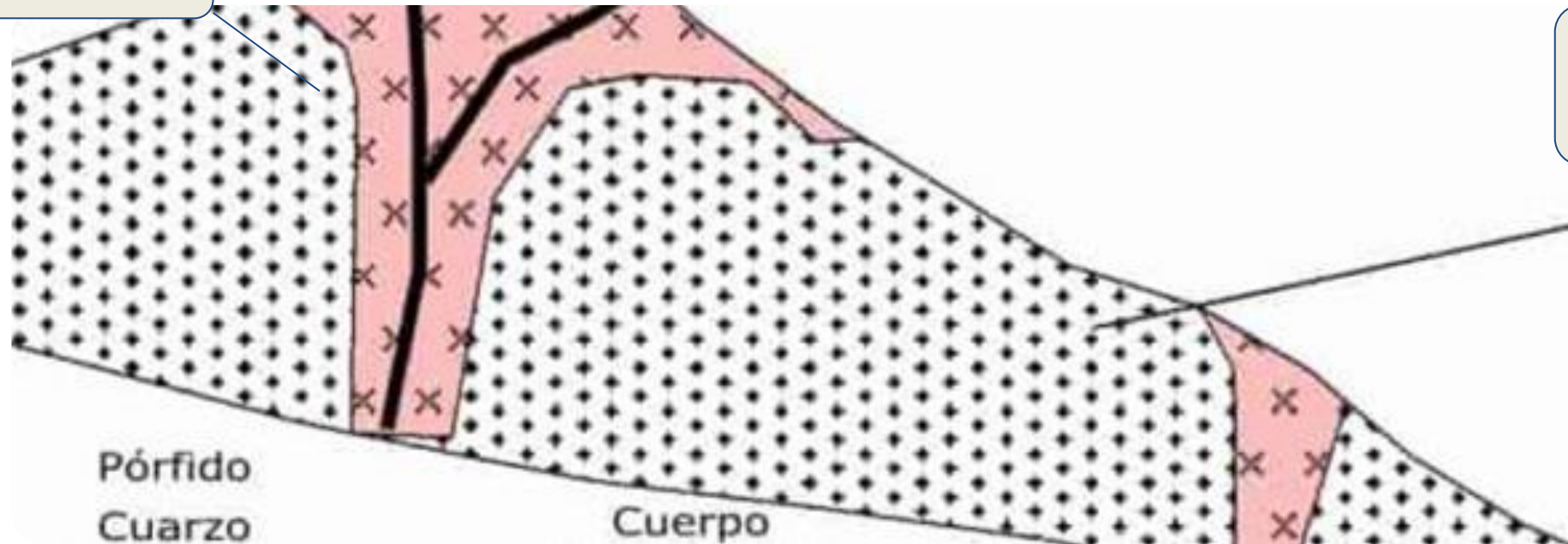
Veta Sur

Yuguana

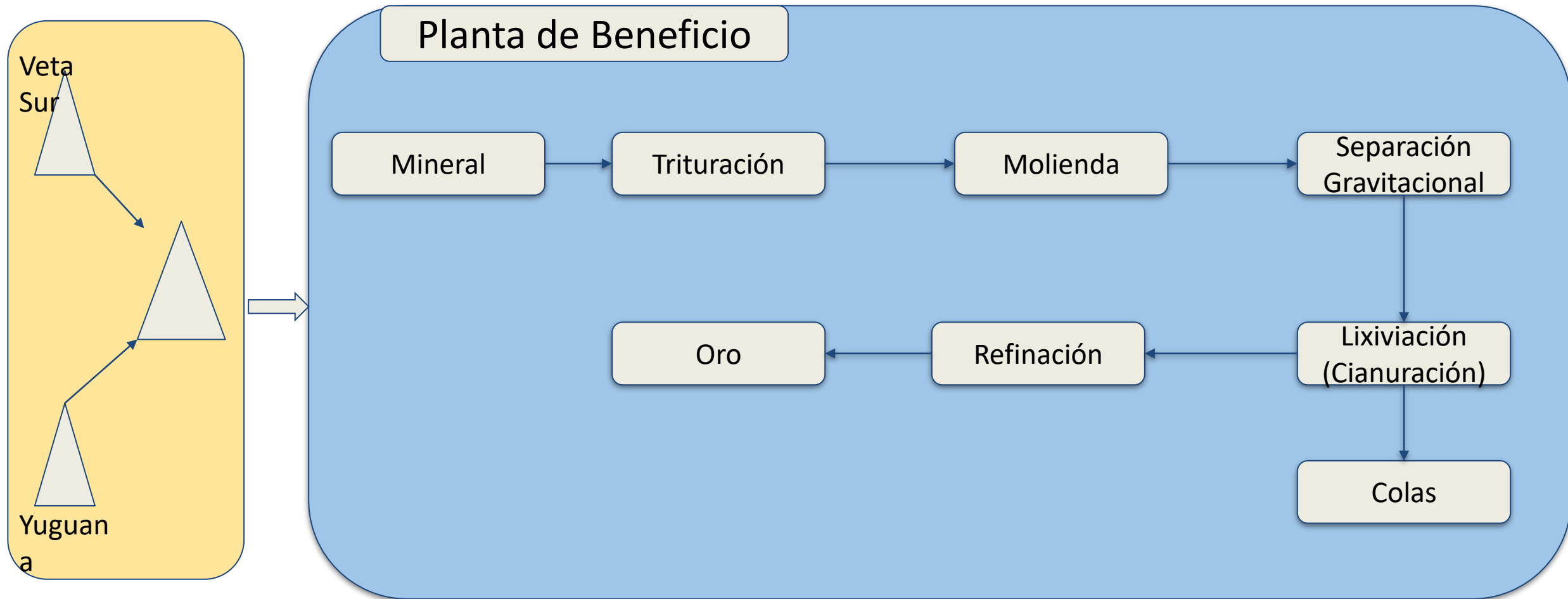
Volcánico  
Erosionado

Pórfido  
Cuarzo

Cuerpo



# Proceso de producción del Oro





# Tabla de Datos

1															
2	Composite	# test	Au	Ag	Cu	Pb	Zn	Fe	As	% Rec. Gravity	Au % Rec. Leach	Total	% Rec. Gravity	Ag % Rec. Leach	Total
3	<u>VSU-0</u>	17	19,987	44,534	779,020	1264,985	5458,144	8,424	265,511	73,815	22,088	95,903	11,767	49,936	61,703
4	<u>VSU-0</u>	22	20,151	49,273	632,720	1264,985	5458,144	8,424	265,511	73,211	22,428	95,639	10,635	68,045	78,680
5	<u>VSU-0</u>	25	19,968	45,890	674,844	1264,985	5458,144	8,424	265,511	73,885	22,015	95,899	11,419	48,500	59,919
6	<u>VSU-0</u>	39	19,458	41,018	670,635	1264,985	5458,144	8,424	265,511	75,822	21,407	97,229	12,776	44,281	57,057
7	<u>VSU-0</u>	55	20,561	43,457	675,558	1264,985	5458,144	8,424	265,511	71,751	24,752	96,503	12,059	56,232	68,291
8	<u>VSU-0</u>	61	19,509	46,377	705,536	1435,872	4441,843	6,469	265,511	75,621	21,922	97,543	11,300	60,193	71,493
9	<u>VSU-0</u>	67	20,618	47,777	740,652	1625,530	4414,298	6,308	265,511	71,554	24,862	96,416	10,968	63,701	74,669
10	<u>VSU-0</u>	73	20,216	47,312	663,898	1254,999	4932,907	5,741	265,511	72,979	24,057	97,036	11,076	60,474	71,550
11	<u>VSU-0</u>	80	21,252	51,169	1007,623	1102,351	2244,770	8,590	265,500	68,880	28,677	97,557	10,162	68,138	78,299
12	<u>VSU-0</u>	81	20,839	45,506	697,557	1313,914	4822,066	6,972	265,511	70,794	25,972	96,766	11,516	55,405	66,921
13	<u>VSU-0</u>	91	20,293	42,891	728,418	1303,626	6610,936	7,907	265,511	72,699	24,053	96,752	12,218	53,279	65,497
14	<u>VSU-20</u>	15	20,753	44,773	836,413	1141,886	2460,446	9,965	474,271	68,982	25,819	94,801	8,737	54,846	63,583
15	<u>VSU-20</u>	16	21,732	47,582	818,915	1141,886	2460,446	9,965	474,271	65,876	29,159	95,035	8,221	52,263	60,484
16	<u>VSU-20</u>	23	21,716	51,942	800,256	1141,886	2460,446	9,965	474,271	65,924	29,016	94,939	7,531	70,041	77,572

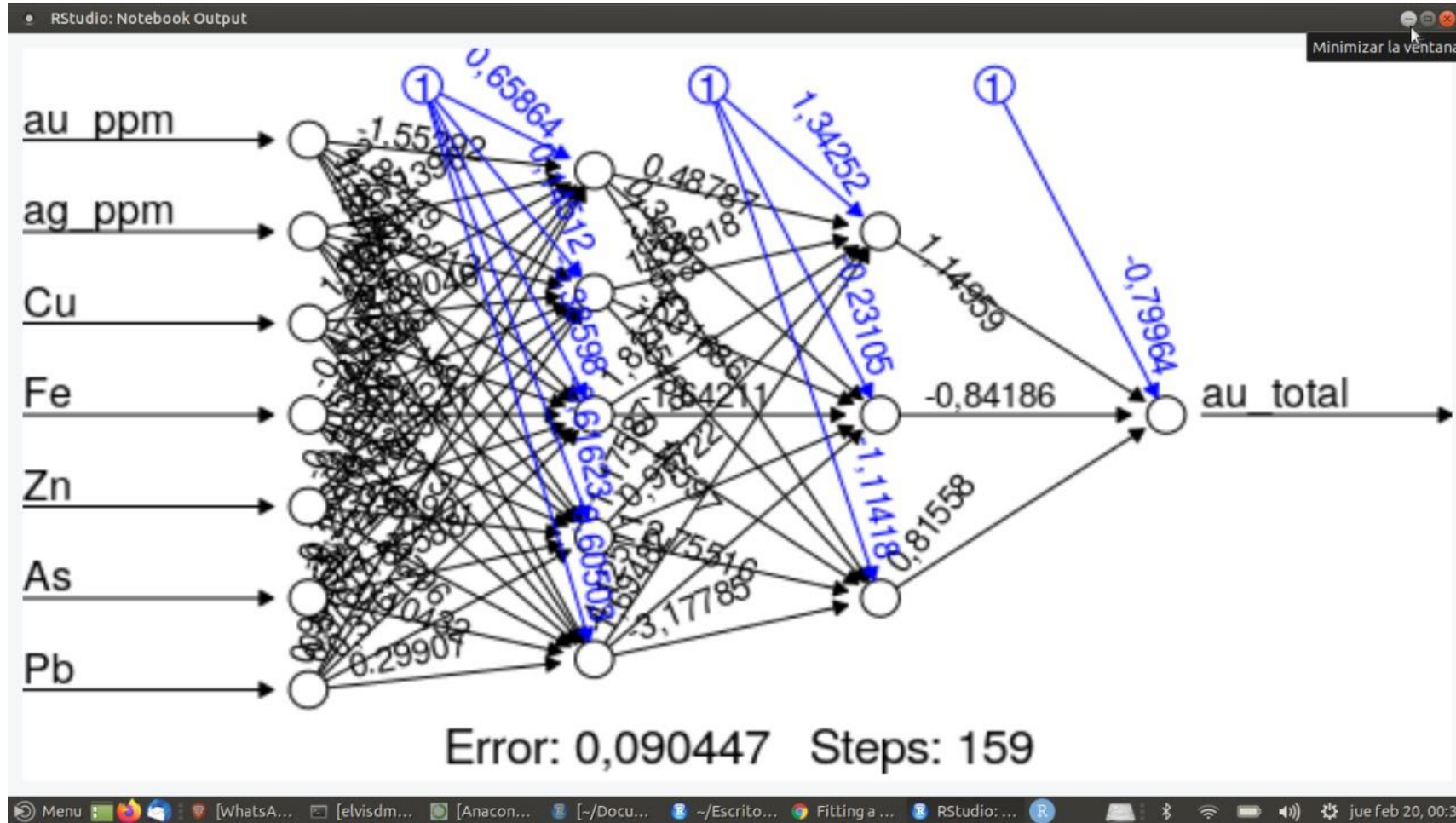
¿Se puede predecir cuánto se va a recuperar de oro al final del proceso minero?

# Datos preparados

	test	au_ppm	ag_ppm	Cu	Pb	Zn	Fe	As	au_gravity	au_leach	au_total	ag_gravity	ag_leach	ag_total	composite
1	22	20	49	632	1264	5458	8	265	73.21110	22.42830	95.63940	10.635488	68.04496	78.68045	VSU-0
2	25	19	45	674	1264	5458	8	265	73.88466	22.01467	95.89933	11.419418	48.49958	59.91900	VSU-0
3	39	19	41	670	1264	5458	8	265	75.82190	21.40685	97.22875	12.775838	44.28111	57.05695	VSU-0
4	55	20	43	675	1264	5458	8	265	71.75123	24.75215	96.50338	12.058712	56.23211	68.29082	VSU-0
5	61	19	46	705	1435	4441	6	265	75.62149	21.92169	97.54318	11.299658	60.19290	71.49256	VSU-0
6	67	20	47	740	1625	4414	6	265	71.55423	24.86189	96.41612	10.968357	63.70083	74.66919	VSU-0
7	73	20	47	663	1254	4932	5	265	72.97881	24.05749	97.03630	11.076197	60.47359	71.54979	VSU-0
8	80	21	51	1007	1102	2244	8	265	68.88001	28.67667	97.55668	10.161628	68.13784	78.29947	VSU-0
9	81	20	45	697	1313	4822	6	265	70.79412	25.97152	96.76564	11.515846	55.40471	66.92056	VSU-0
10	91	20	42	728	1303	6610	7	265	72.69893	24.05350	96.75243	12.218081	53.27897	65.49706	VSU-0
11	15	20	44	836	1141	2460	9	474	68.98168	25.81912	94.80079	8.736721	54.84646	63.58318	VSU-20
12	16	21	47	818	1141	2460	9	474	65.87594	29.15894	95.03488	8.220964	52.26279	60.48375	VSU-20
13	23	21	51	800	1141	2460	9	474	65.92357	29.01570	94.93927	7.530982	70.04138	77.57236	VSU-20
14	40	21	44	752	1141	2460	9	474	68.03405	26.93313	94.96718	8.788163	48.78963	57.57779	VSU-20
15	71	18	45	764	1170	4093	6	474	78.16934	17.57556	95.74489	8.593709	66.77955	75.37326	VSU-20
16	72	21	48	778	1238	4141	5	474	66.29339	31.85603	98.14941	8.097376	66.17509	74.27247	VSU-20



# Modelo de Predicción



## Predichos vs Observados

Error = 2,59

<b>prediccion</b> <dbi>	<b>real</b> <dbi>
95.78129	97.54318
95.64481	94.96718
94.51549	95.74489
96.42540	98.14941
95.95849	95.72399
93.77977	94.49732
95.12479	92.86778
93.87093	95.19341
92.19314	92.64212
89.73256	94.58457