ELEMENTOS DE MACHINE LEARNING

Impartido por: MSc. Ronaldo Canizales

Diplomado de Análisis Computacional Estadístico de Datos con Python

About me:

MSc. Ronaldo Canizales
Trustworthy AI, Graph ML, HPC, CS Edu
www.cs.colostate.edu/~rcanizal

Studied
Worked
Presented

Agenda del Módulo

01

Intro. teórica y práctica

02

Aprendizaje supervisado

03

Deep Learning

04

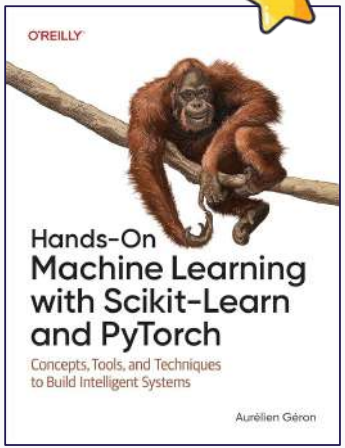
Otras aplicaciones

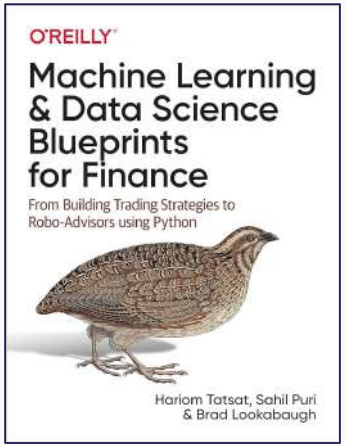
05

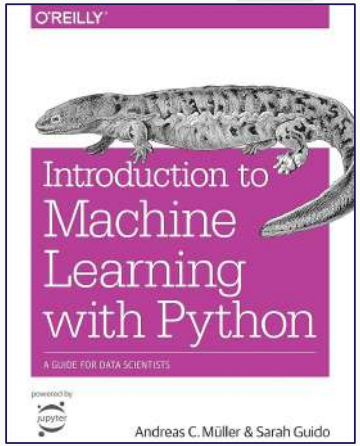
Siguientes pasos

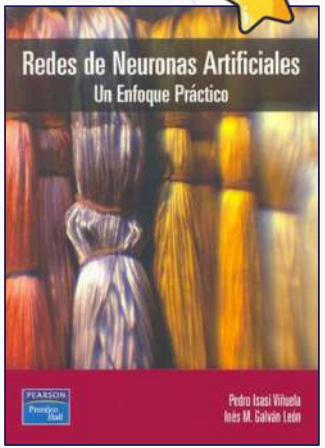
Repo

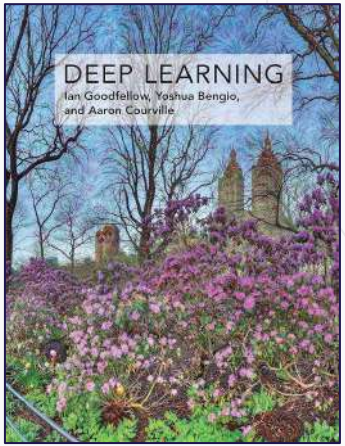
github.com/armandocodigos/UCA-Winter25-ElementosML

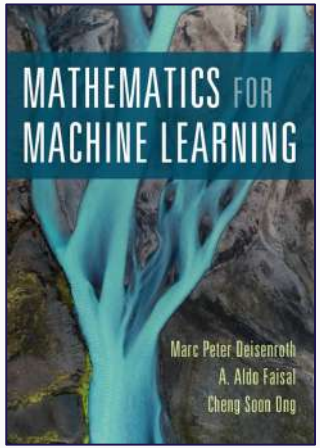

O'REILLY
Hands-On
Machine Learning
with Scikit-Learn
and PyTorch
Concepts, Tools, and Techniques
to Build Intelligent Systems
Aurélien Géron

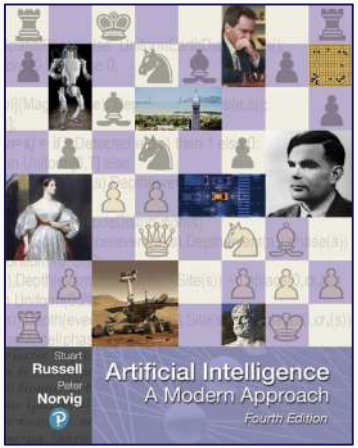

O'REILLY
Machine Learning
& Data Science
Blueprints
for Finance
From Building Trading Strategies to
Robo-Advisors using Python
Hariom Tatsat, Sahil Puri
& Brad Lookabaugh

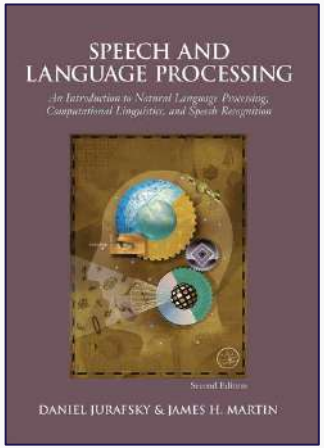

O'REILLY
Introduction to
Machine
Learning
with Python
A GUIDE FOR DATA SCIENTISTS
Andreas C. Müller & Sarah Guido

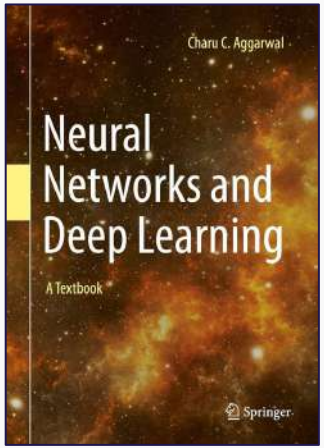

Redes de Neuronas Artificiales
Un Enfoque Práctico
Pedro Isasi Viñuela
Inés M. Galván León

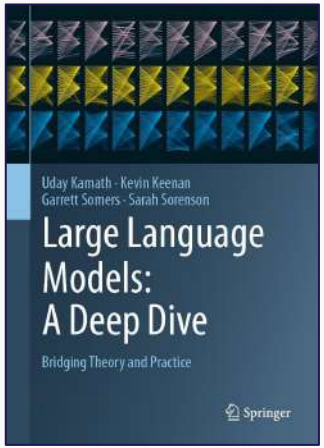

DEEP LEARNING
Ian Goodfellow, Yoshua Bengio,
and Aaron Courville


MATHEMATICS FOR
MACHINE LEARNING
Marc Peter Deisenroth
A. Aldo Faisal
Cheng Soon Ong


Stuart
Russell
Peter
Norvig
Artificial Intelligence
A Modern Approach
Fourth Edition


SPEECH AND
LANGUAGE PROCESSING
An Introduction to Natural Language Processing,
Computational Linguistics, and Speech Recognition
Second Edition
DANIEL JURAFSKY & JAMES H. MARTIN


Charu C. Aggarwal
Neural
Networks and
Deep Learning
A Textbook
Springer


Uday Kamath · Kevin Keenan
Garrett Somers · Sarah Sorenson
Large Language
Models:
A Deep Dive
Bridging Theory and Practice
Springer

1.1

Introducción

Desmitificando el Machine Learning: conceptos relacionados

¿Qué es el Machine Learning?

- "Machine learning is the science (and art) of programming computers so they can learn from data." Aurelien Geron, 2023.
- "[Machine learning is the] field of study that gives computers the ability to learn without being explicitly programmed." Arthur Samuel, 1959.
- "A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E." Tom Mitchell, 1997.

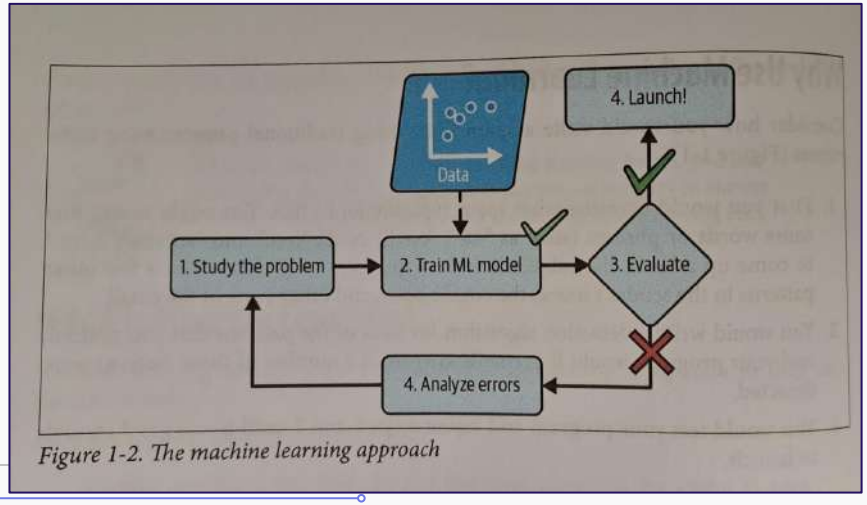
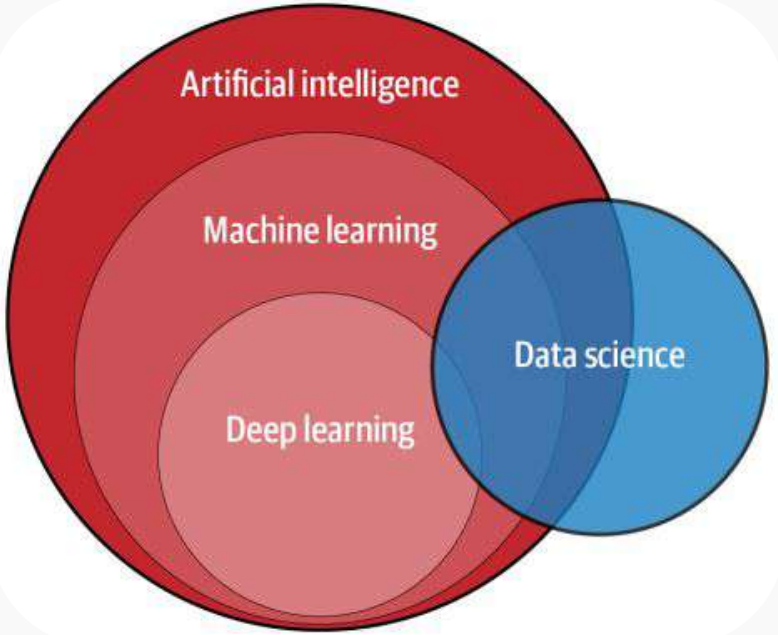


Figure 1-2. The machine learning approach

DS vs AI vs ML vs DL

La inteligencia artificial es la ciencia que estudia la forma de hacer que una computadora desarrolle la capacidad de realizar con éxito tareas complejas que generalmente requieren inteligencia humana.

- Percepción visual
- Reconocimiento de voz
- Toma de decisiones
- Traducción entre idiomas
- ¡Muchas más!



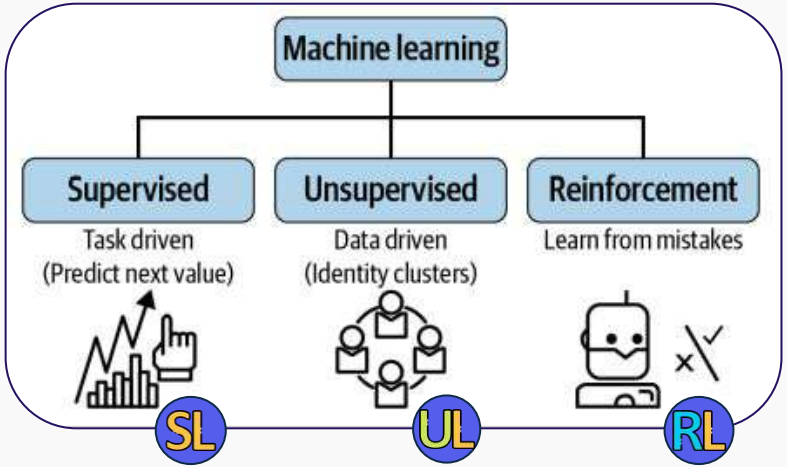
DS vs AI vs ML vs DL

El aprendizaje de máquina (ML) es una aplicación de la IA. Es un conjunto de algoritmos que poseen la capacidad de aprender automáticamente del entorno.

El aprendizaje profundo (DL) es un subconjunto del ML que implica el estudio de algoritmos relacionados con redes neuronales artificiales. Permite resolver problemas más complejos.

La ciencia de datos (DS) es un campo interdisciplinario similar a la minería de datos. Se encarga de extraer información de los datos en diversas formas, ya sea estructuradas o no.

Tipos de Aprendizaje de Máquina



El objetivo principal del aprendizaje supervisado es entrenar un modelo a partir de datos etiquetados que nos permita hacer predicciones sobre datos futuros o no vistos.

Aquí, el término supervisado se refiere a un conjunto de muestras donde ya se conocen las señales de salida deseadas (etiquetas).

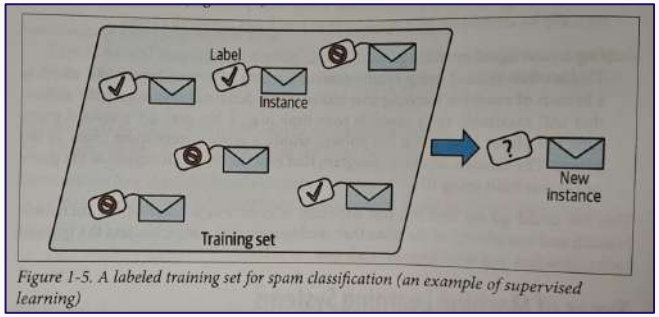


Figure 1-5. A labeled training set for spam classification (an example of supervised learning)

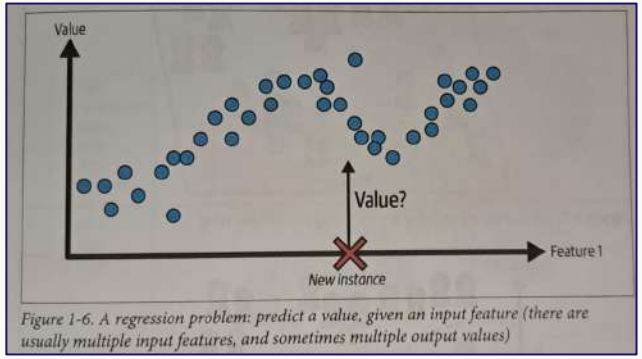
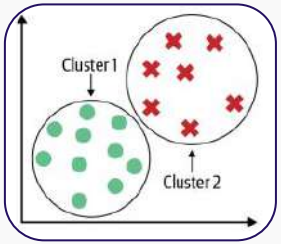


Figure 1-6. A regression problem: predict a value, given an input feature (there are usually multiple input features, and sometimes multiple output values)

Tipos de Aprendizaje de Máquina

El aprendizaje no supervisado es un tipo de aprendizaje automático que se utiliza para extraer inferencias de conjuntos de datos que consisten en datos de entrada sin respuestas etiquetadas.



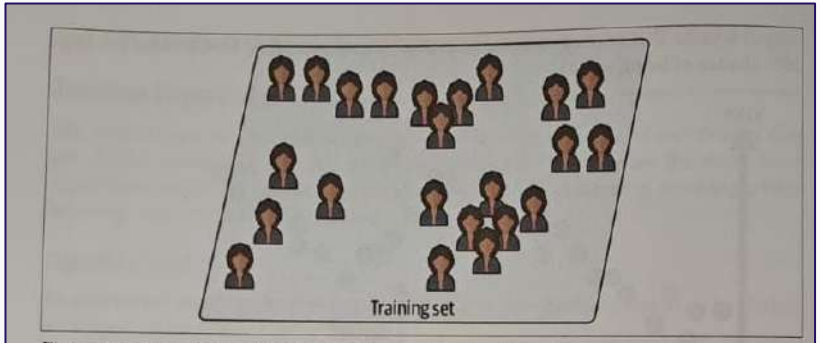


Figure 1-7. An unlabeled training set for unsupervised learning

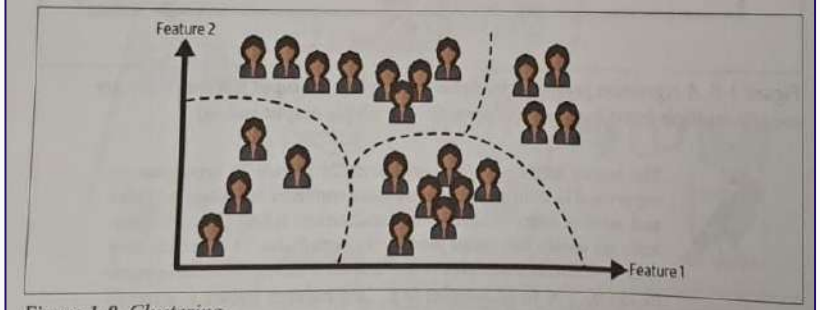
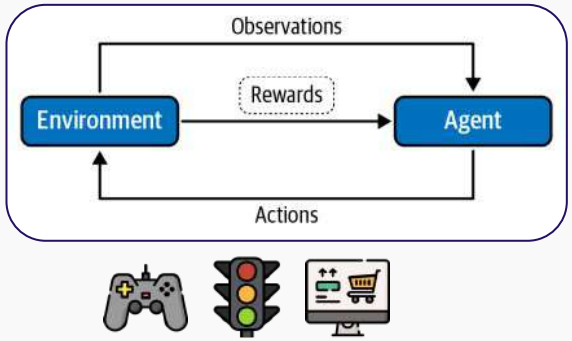


Figure 1-8. Clustering

Tipos de Aprendizaje de Máquina

Aprender de la experiencia, las recompensas o castigos es el concepto detrás del aprendizaje por refuerzo. Se trata de tomar las acciones adecuadas para maximizar la recompensa en situaciones particulares donde no hay una respuesta explícita.



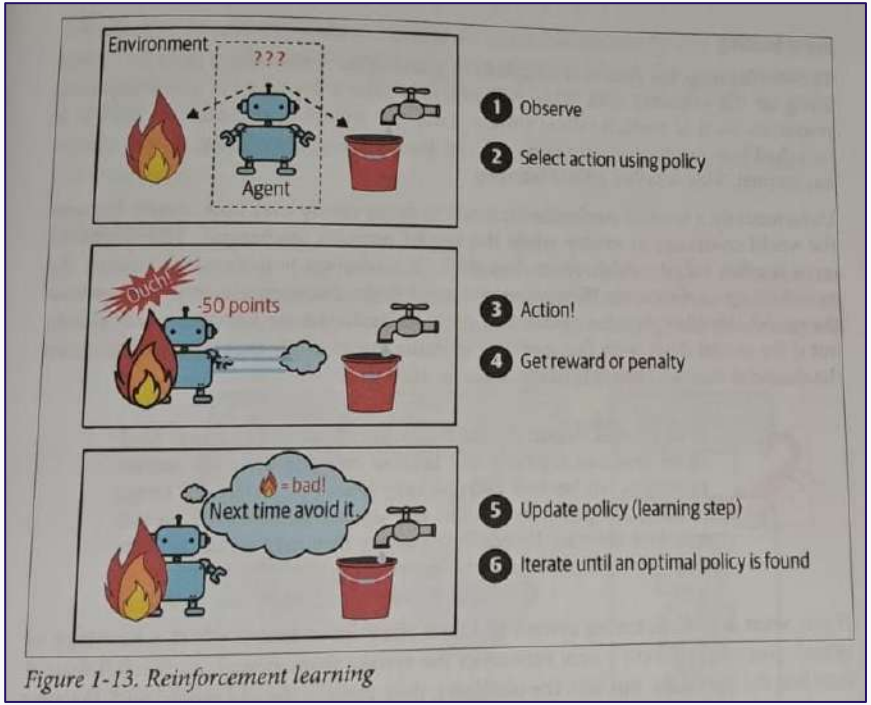


Figure 1-13. Reinforcement learning

Tipos de Aprendizaje de Máquina

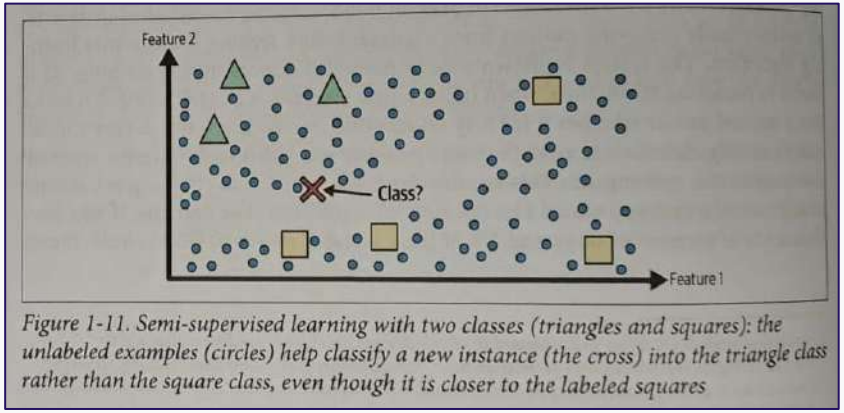
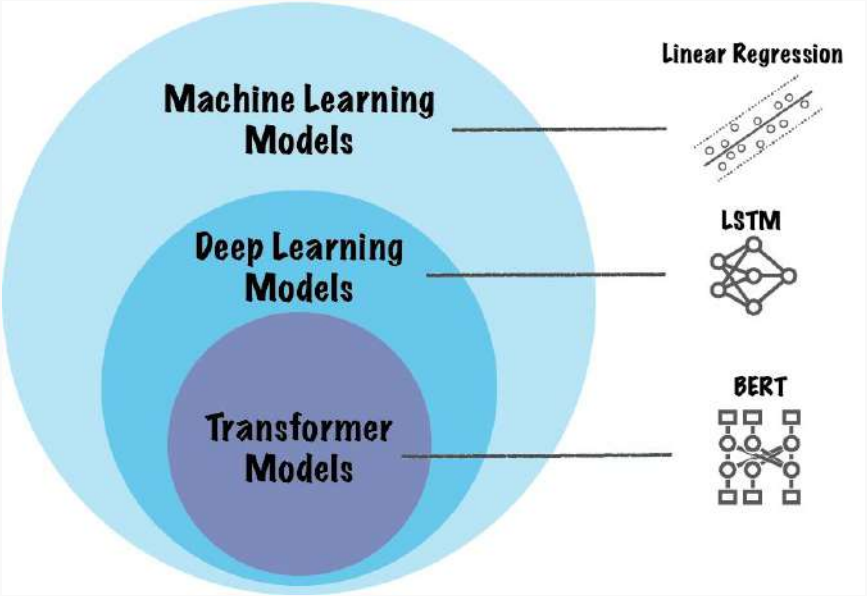


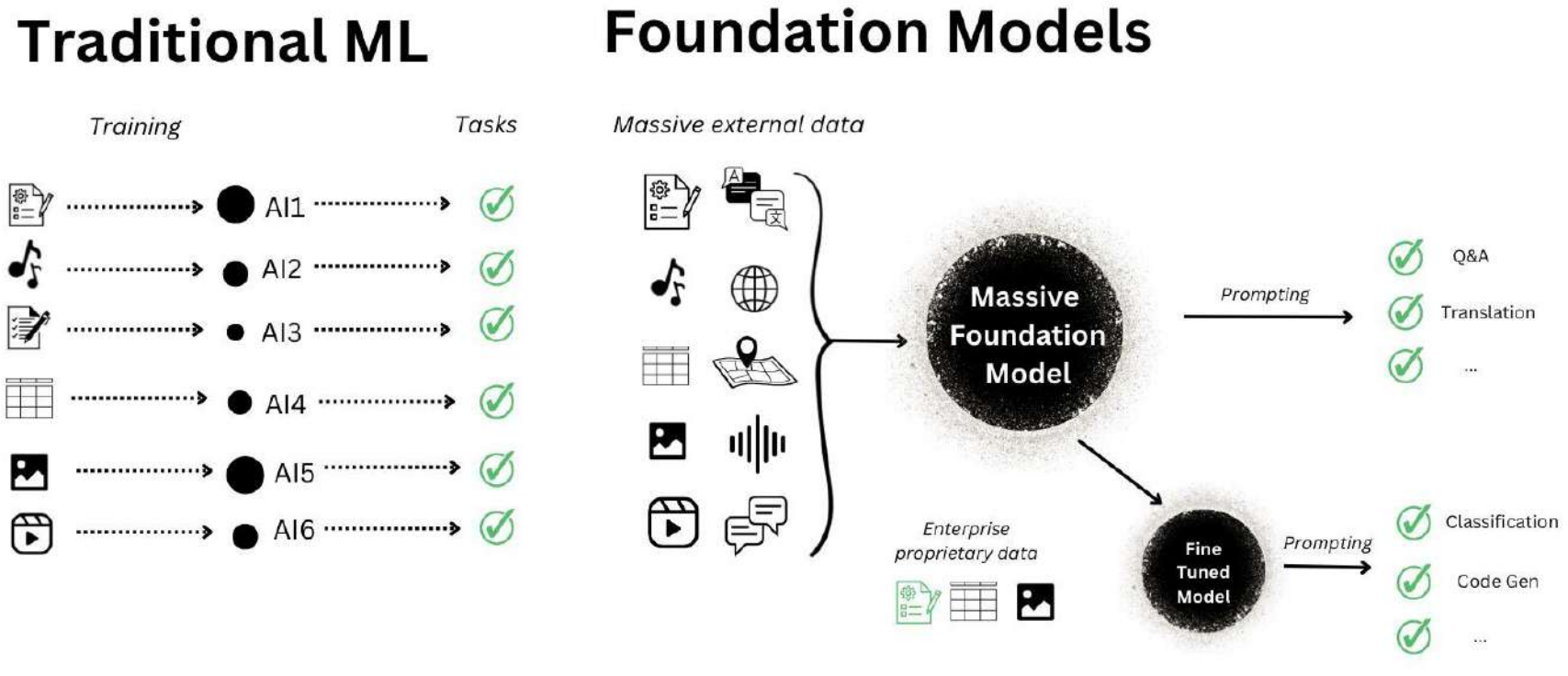
Figure 1-11. Semi-supervised learning with two classes (triangles and squares): the unlabeled examples (circles) help classify a new instance (the cross) into the triangle class rather than the square class, even though it is closer to the labeled squares

Aprendizaje semi-supervisado

"Attention is all you need" (2017)



Tipos de Aprendizaje de Máquina



Resolución de problemas con IA

Automatización: reportes de tránsito (LLMs)📝

Automatización: mapas geoespaciales (LLMs)🏆

Recuperación de datos acelerográficos históricos (VLMs)📝📊

Análisis: robot en laboratorio químico (VLMs y CNNs)📊

Software: estimación de recursos (GraphML)📝📚

Automatización: actas violaciones a DDHH (web scraping+LLMs)📝

Realidad Virtual: estrés cognitivo (MLPs)🏆

Recomendación: mercado laboral (GraphML)

Sismología: mapa de respuesta homogénea (Ensemble)

Sismología: ambientes tectónicos y fuentes sísmicas (MLPs)

Realidad Aumentada: turismo y volcanes (CNNs)

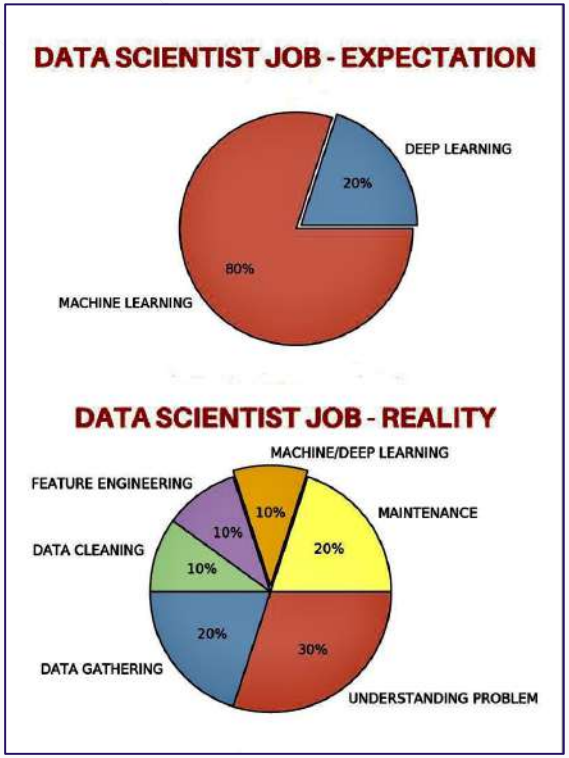
Redes sociales: análisis de discurso (NLP)

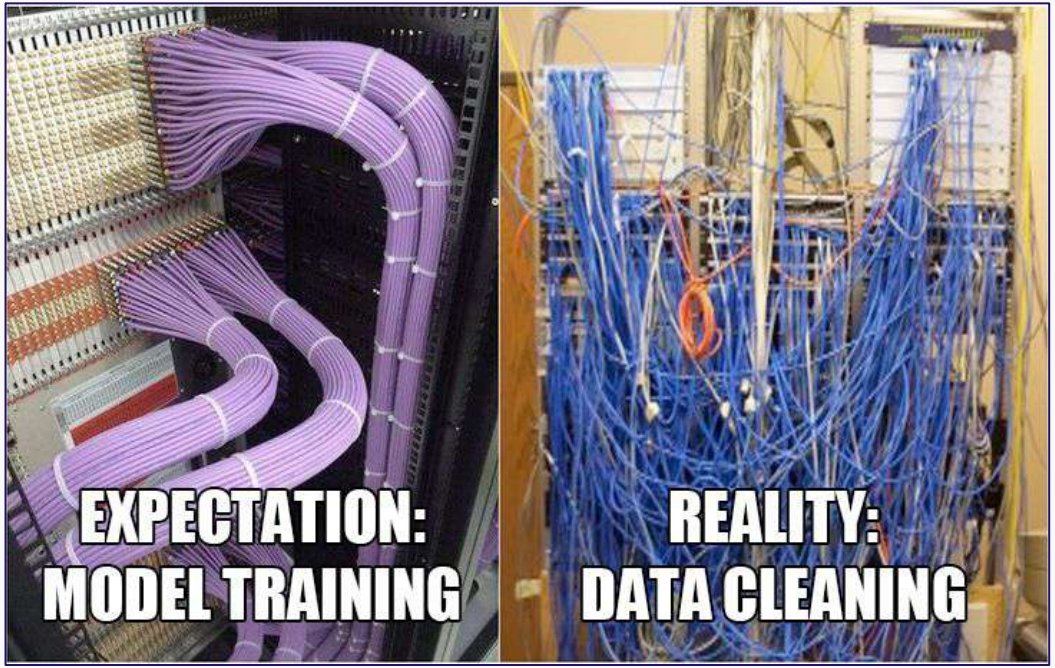
Educación: evaluaciones cualitativas (NLP+DTs)

Software: detección temprana errores (MLPs)

Control de brazo robot interactivo (DTs) 📚

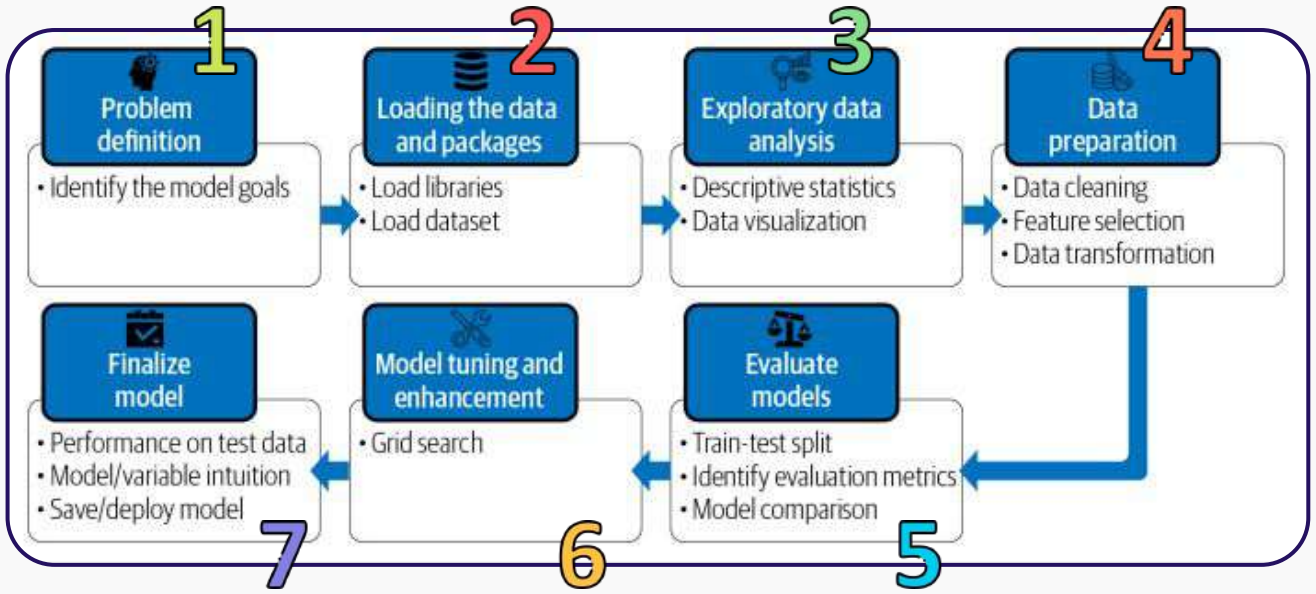
Etapas de un proyecto de DS y ML





Etapas de un proyecto de DS y ML

Ciencia de datos



Aprendizaje de Máquina

1.2

Manos a la obra

Ejemplo sencillo “end-to-end”

Librerías a utilizar

NumPy

pip install numpy

Pandas

pip install pandas

Matplotlib

pip install matplotlib

Python 3.12+

Anaconda
JupyterLab
Jupyter Notebook
Google Colab

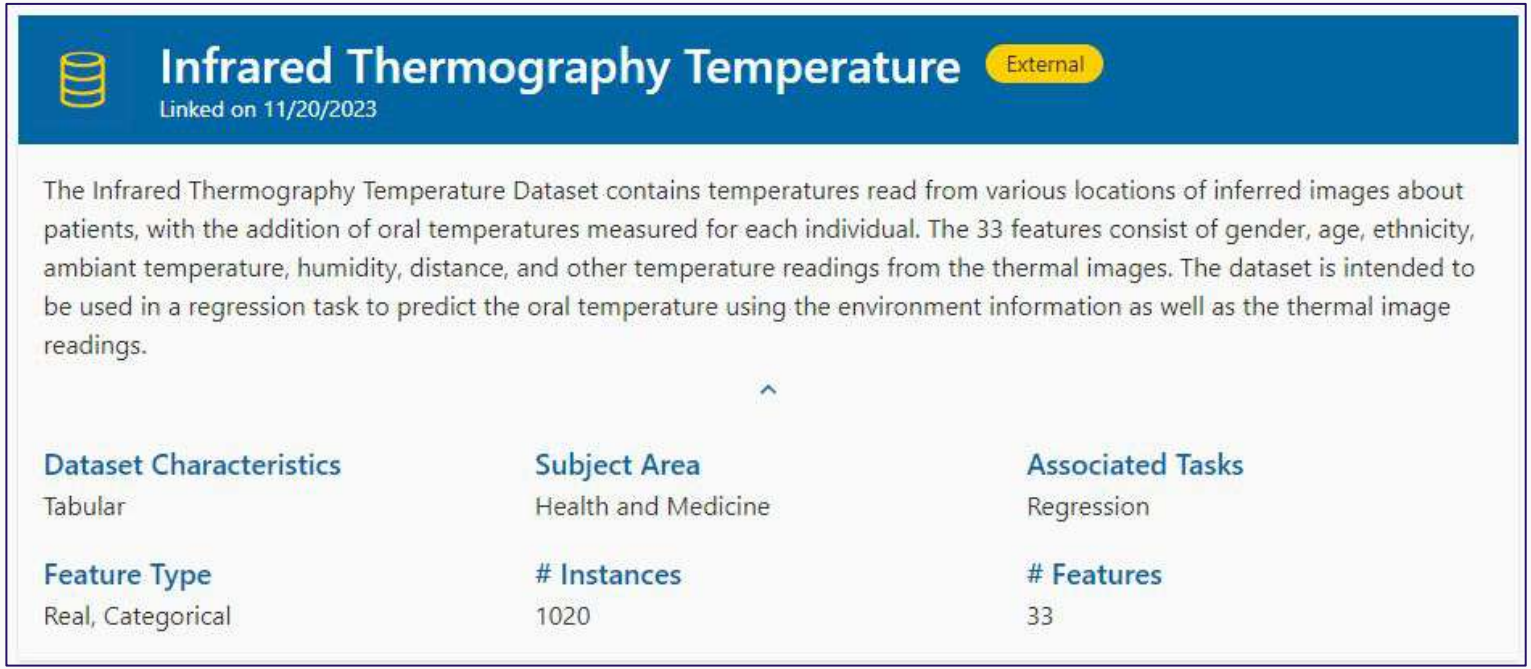
Scikit-Learn

pip install -U scikit-learn

ML Repository

pip install ucimlrepo

Sencillo ejemplo “end-to-end”



Infrared Thermography Temperature

External

Linked on 11/20/2023

The Infrared Thermography Temperature Dataset contains temperatures read from various locations of inferred images about patients, with the addition of oral temperatures measured for each individual. The 33 features consist of gender, age, ethnicity, ambiant temperature, humidity, distance, and other temperature readings from the thermal images. The dataset is intended to be used in a regression task to predict the oral temperature using the environment information as well as the thermal image readings.

Dataset Characteristics
Tabular

Subject Area
Health and Medicine

Associated Tasks
Regression

Feature Type
Real, Categorical

 # Instances
1020

 # Features
33

Enlace

Training y Test subsets

Etapas de una solución

Actividad recomendada

Instalar todas las librerías en un ambiente de desarrollo de su preferencia

Explorar manualmente los datos de entrada para obtener un modelo más ligero o con menor error

2.1

Aprendizaje supervisado

Principales desafíos

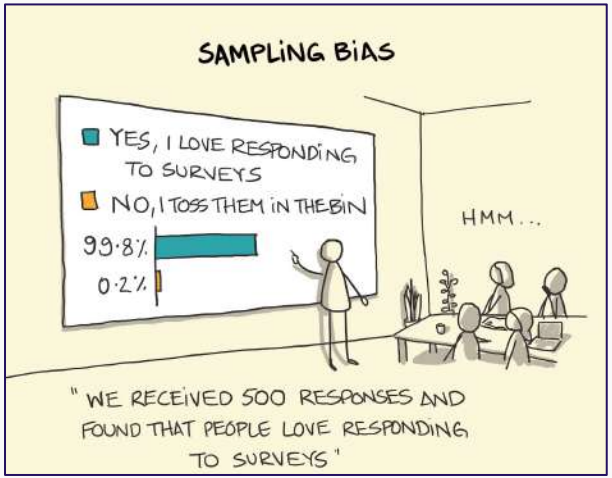
- Cantidad insuficiente de datos (entrenamiento + prueba).
  - Principal diferencia con el aprendizaje humano.
  - *"The unreasonable effectiveness of data"* by Peter Norvig (2009).

- El conjunto de aprendizaje debe poseer las siguientes características:
  - **Ser significativo:**
  - Debe haber un número suficiente de ejemplos.
  - No podrá generalizar adecuadamente.

  - **Ser representativo:**
  - El conjunto debe ser diverso.
  - Se debe evitar tener muchos más ejemplos de un tipo que del resto.

Principales desafíos

- Utilizar data insuficiente:
- Sesgo muestral:



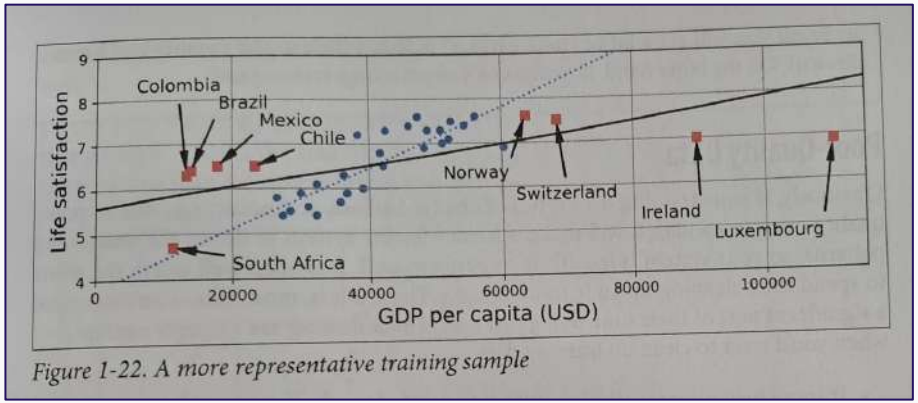
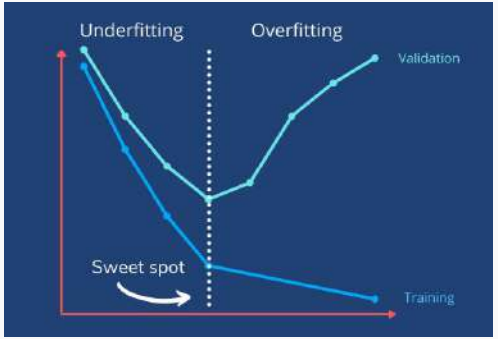


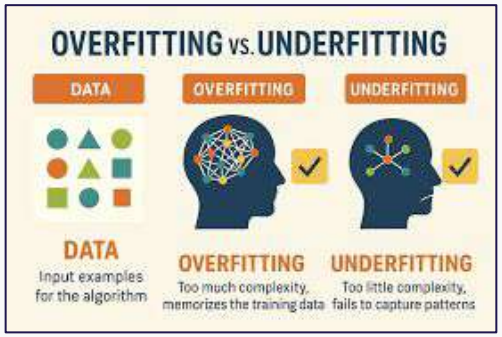
Figure 1-22. A more representative training sample

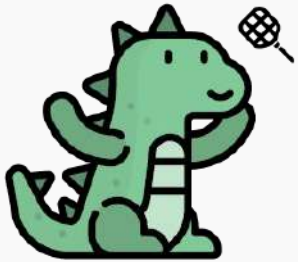
- Elección presidencial EEUU 1936: La revista “Literary Digest” predijo victoria demócrata con 57%, pero en realidad los republicanos ganaron con 62%.
  - A quienes enviaron encuestas?
  - Menos del 25% de los encuestados respondieron.

Principales desafíos



- Datos de entrenamiento de baja calidad:
  - Valores nulos, datos atípicos, y ruido (errores al obtener la data).
- Datos irrelevantes:
  - Es necesario invertir tiempo en “ingeniería de características”.
  - Se puede hacer uso inclusive de metodos de reduccion de la dimensionalidad.
- Overfitting and underfitting.

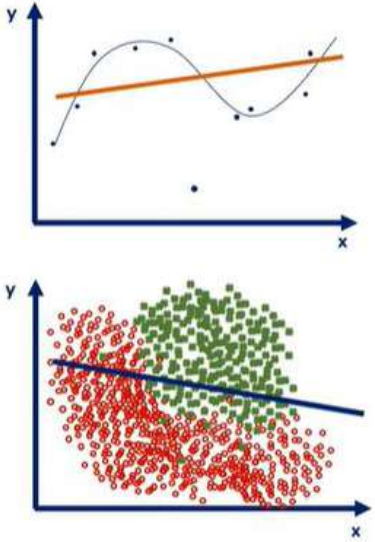




Underfitting

Overfitting

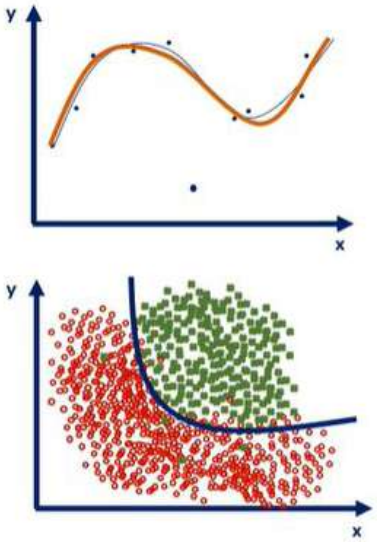
An **underfitted** model



Doesn't capture any logic

- High loss
- Low accuracy

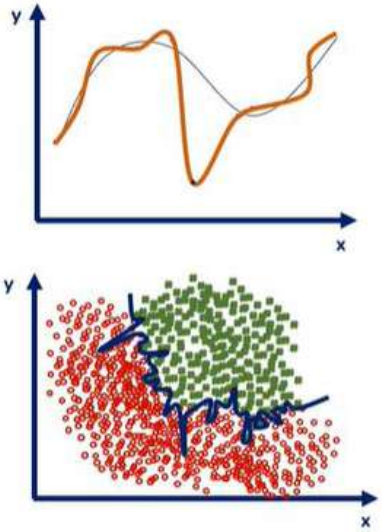
A **good** model



Captures the underlying logic of the dataset

- Low loss
- High accuracy

An **overfitted** model

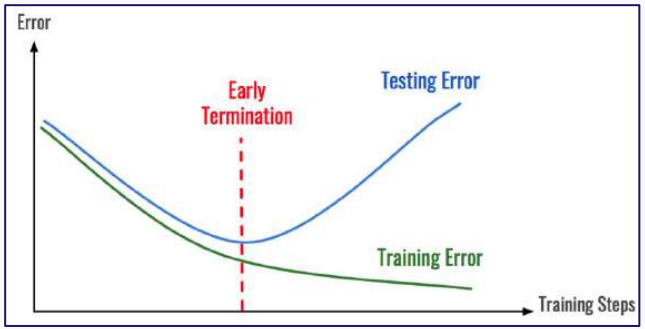


Captures all the noise, thus "missed the point"

- Low loss
- Low accuracy

Principales desafíos

- Soluciones al sobreajuste (overfitting):
  - Simplificar el modelo (disminuir la cantidad de parámetros).
  - Obtener más datos de entrenamiento.
  - Reducir el ruido en los datos de entrenamiento (remover datos atípicos o irrelevantes).
  - Early stopping!



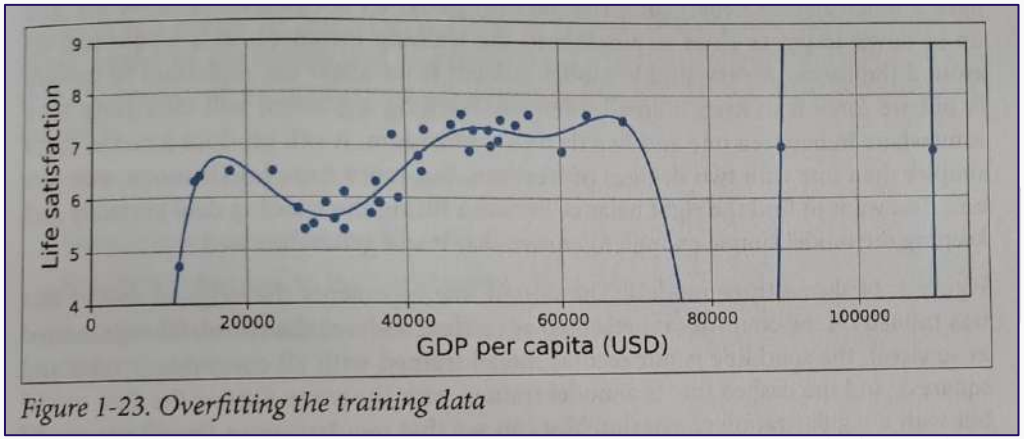
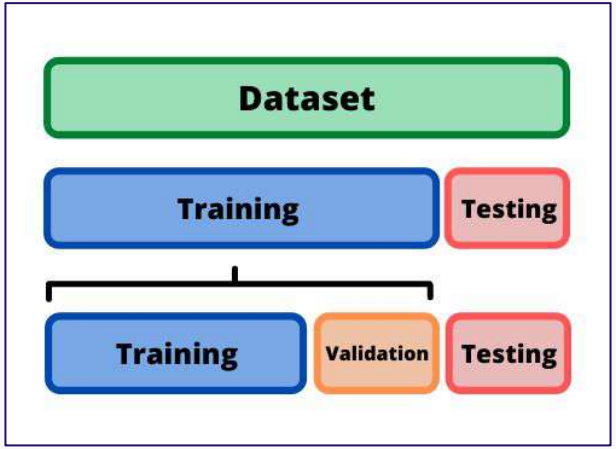


Figure 1-23. Overfitting the training data

Estrategias útiles

- Subconjuntos de datos: entrenamiento, validación y prueba.
- Selección de hyper-parámetros.



2.2

Regresión

Aprendizaje supervisado (salida continua)

Student Performance

Donated on 11/26/2014

Predict student performance in secondary education (high school).

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate	Social Science	Classification, Regression
Feature Type	**# Instances**	**# Features**
Integer	649	30

Dataset Information

Additional Information

This data approach student achievement in secondary education of two Portuguese schools. The data attributes include student grades, demographic, social and school related features) and it was collected by using school reports and questionnaires. Two datasets are provided regarding the performance in two distinct subjects: Mathematics (mat) and Portuguese language (por). In [Cortez and Silva, 2008], the two datasets were modeled under binary/five-level classification and regression tasks. Important note: the target attribute G3 has a strong correlation with attributes G2 and G1. This occurs because G3 is the final year grade (issued at the 3rd period), while G1 and G2 correspond to the 1st and 2nd period grades. It is more difficult to predict G3 without G2 and G1, but such prediction is much more useful (see paper source for more details).

Enlace

Máquinas de soporte vectorial

Su propósito es encontrar un hiperplano que maximice la distancia entre las instancias de entrenamiento.

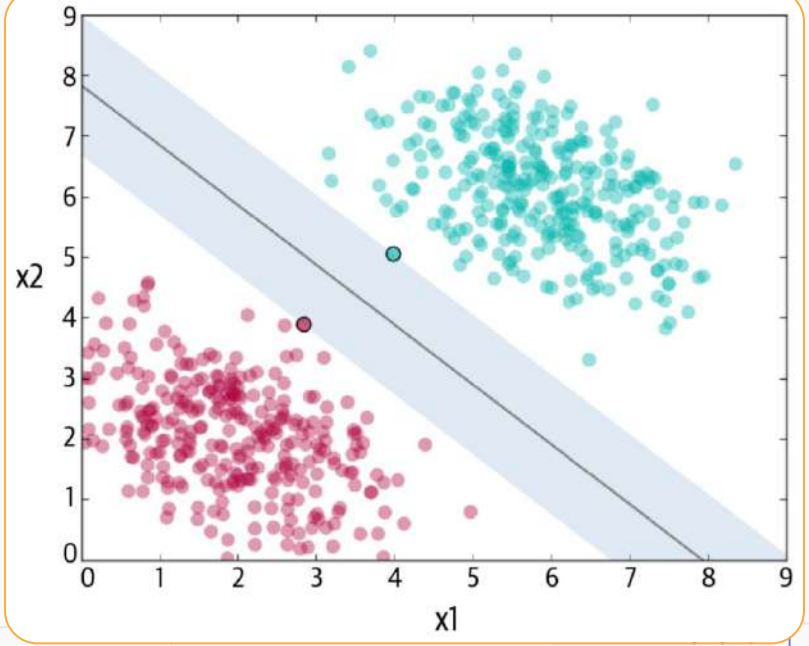
En la práctica, los datos casi nunca pueden ser separados perfectamente por un hiperplano (como en la figura), así que se hace necesario flexibilizar esa regla: permitir que algunos puntos violen la frontera.

Ventajas: es robusto ante el sobreajuste, especialmente en altas dimensiones. Es muy útil para casos no-lineales

Desventajas: no es muy intuitivo. Requiere gran cantidad de memoria cuando se utiliza en conjuntos de datos grandes.

```
from sklearn.svm import SVR
model = SVR()
model.fit(X, Y)
```

```
from sklearn.svm import SVC
model = SVC()
model.fit(X, Y)
```

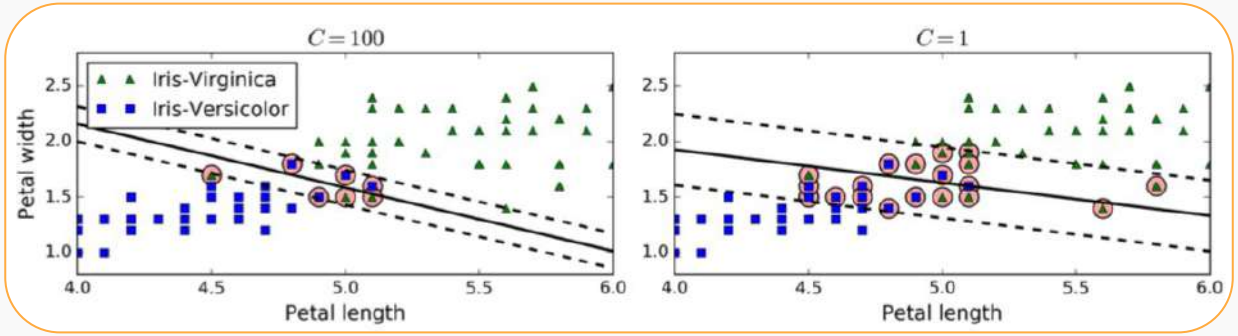


Hiper-parámetros de las MSV

Penalty (C en sklearn)

- Se le especifica al algoritmo qué tan estrictos debemos ser para evitar que hayan violaciones a la regla.
- Valores grandes crearán un hiperplano con un margen más pequeño.
- Valores pequeños conllevarán a márgenes más grandes, y de esa forma se reducirá el sobreajuste.

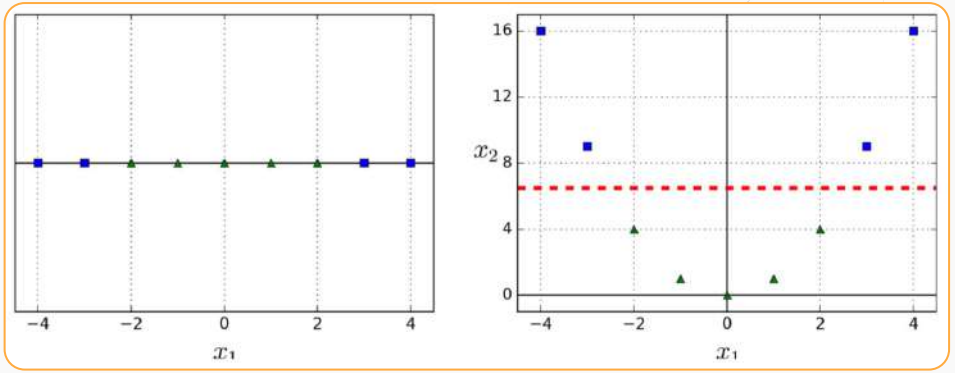
```
SVC(kernel="linear", C=1)
```



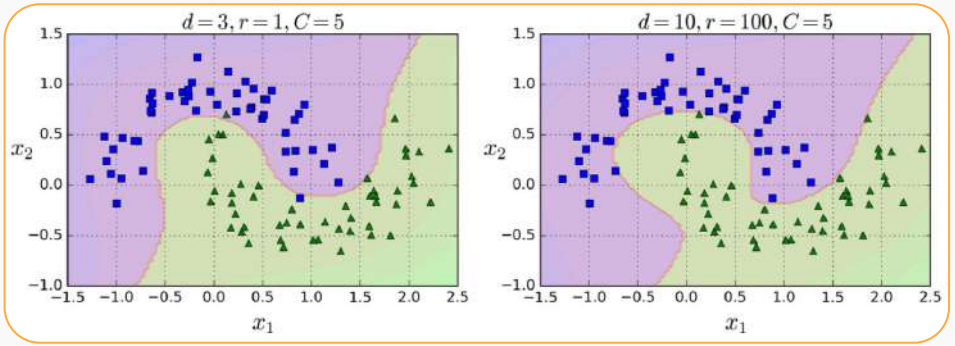
H-params en MSV

Kernels (kernel en sklearn)

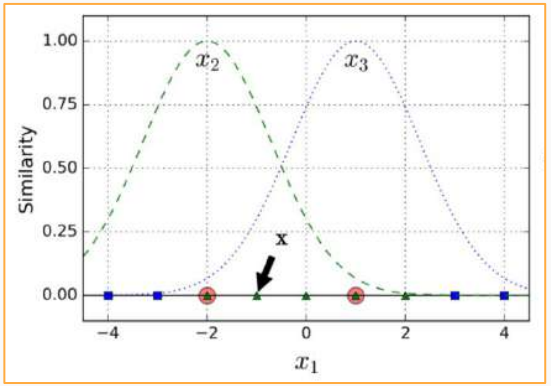
- Existen casos en que además de flexibilizar la regla, es necesario agregar una dimensión extra a los datos.
- Se controla la manera en que las variables de entrada serán proyectadas.
- Existen varios, pero: lineal, polinómico y RBF (Gaussian Radial Basis Function) son los más comunes.

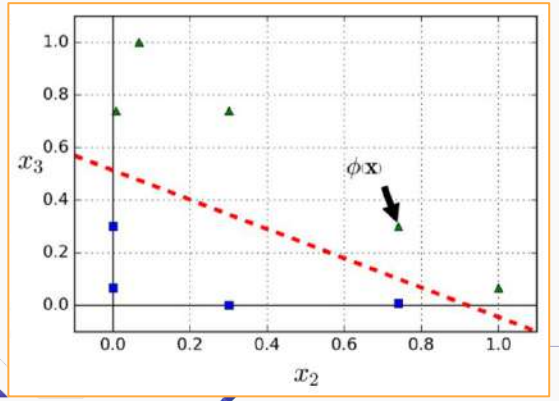



```
SVC(kernel="poly", degree=3, coef0=1, C=5)
```

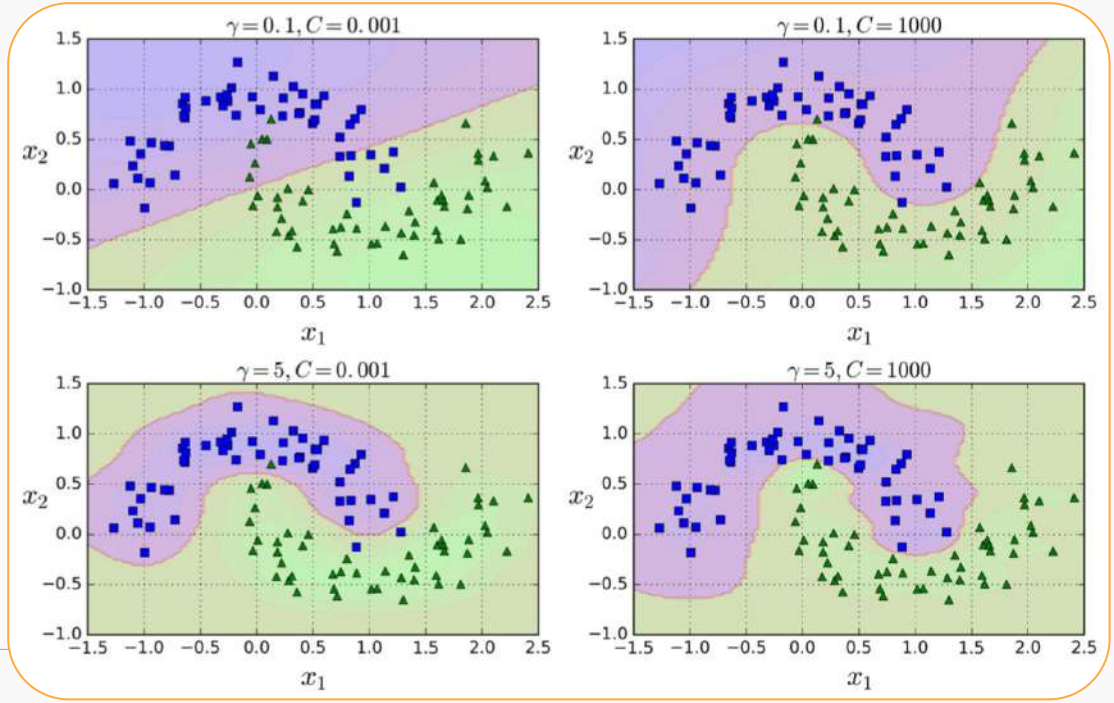


Hiper-parámetros de las MSV






```
SVC(kernel="rbf", gamma=5, C=0.001)
```



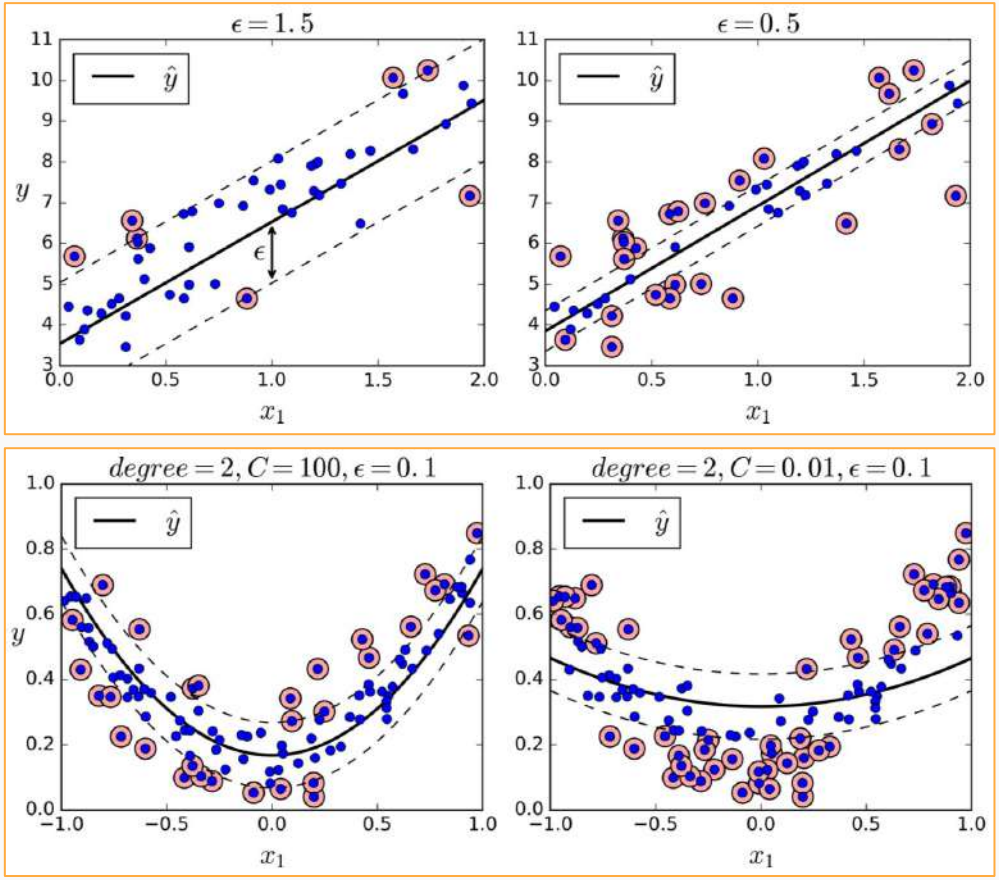
H-params en MSV

```
from sklearn.svm import LinearSVR

svm_reg = LinearSVR(epsilon=1.5)
svm_reg.fit(X, y)
```

Epsilon (epsilon en sklearn)

- Intenta colocar tantas instancias como sea posible dentro de los márgenes.

```
from sklearn.svm import SVR

svm_poly_reg = SVR(kernel="poly", degree=2, C=100, epsilon=0.1)
svm_poly_reg.fit(X, y)
```



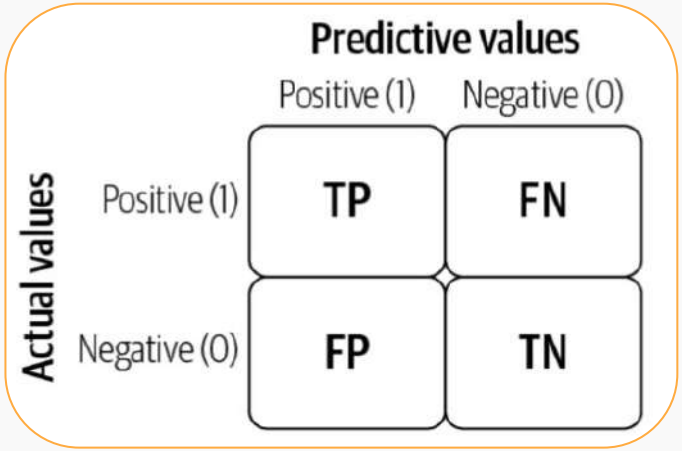
2.3

Clasificación

Aprendizaje supervisado (salida discreta)

Matriz de confusión

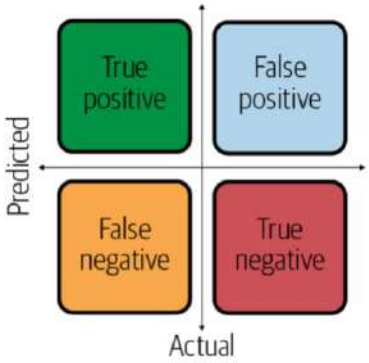
- En algunos escenarios será de mayor importancia un indicador que otro.
- Cuando las clases están balanceadas, entonces el Accuracy es muy útil, caso contrario no es de fiar.



$$\text{Precision} = \frac{\text{True positive}}{\text{Actual results}} \quad \text{or} \quad \frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

$$\text{Recall} = \frac{\text{True positive}}{\text{Predictive results}} \quad \text{or} \quad \frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$

$$\text{Accuracy} = \frac{\text{True positive} + \text{True negative}}{\text{Total}}$$



Cuando se trate de más de dos clases, las métricas deberán calcularse para cada una de ellas.

Bank Marketing

Donated on 2/13/2012

The data is related with direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe a term deposit (variable y).

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate	Business	Classification

Feature Type	# Instances	# Features
Categorical, Integer	45211	16

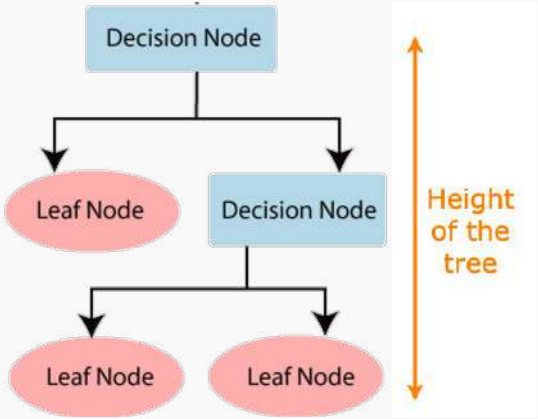
Dataset Information

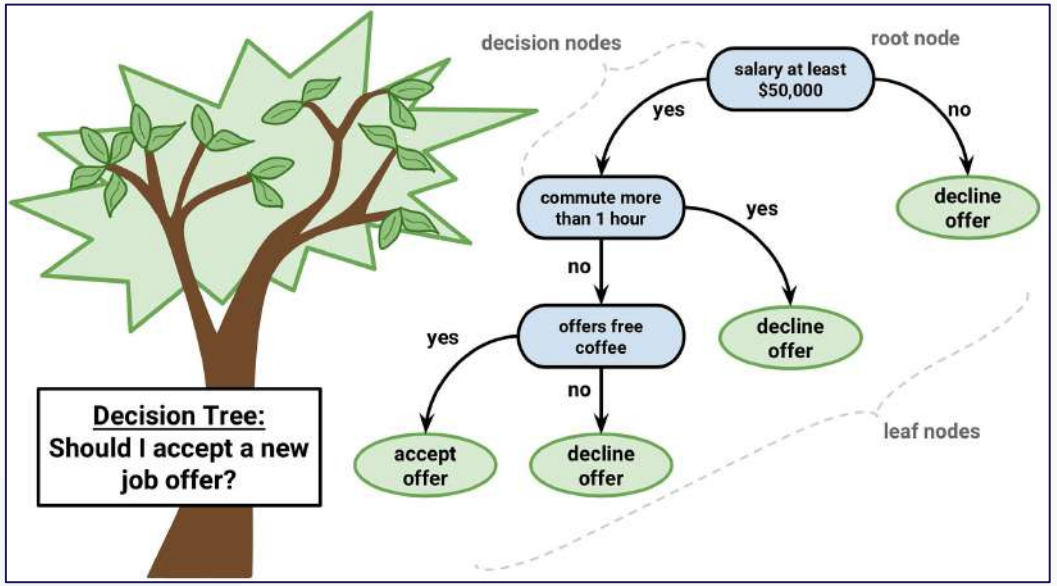
Additional Information

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

Enlace

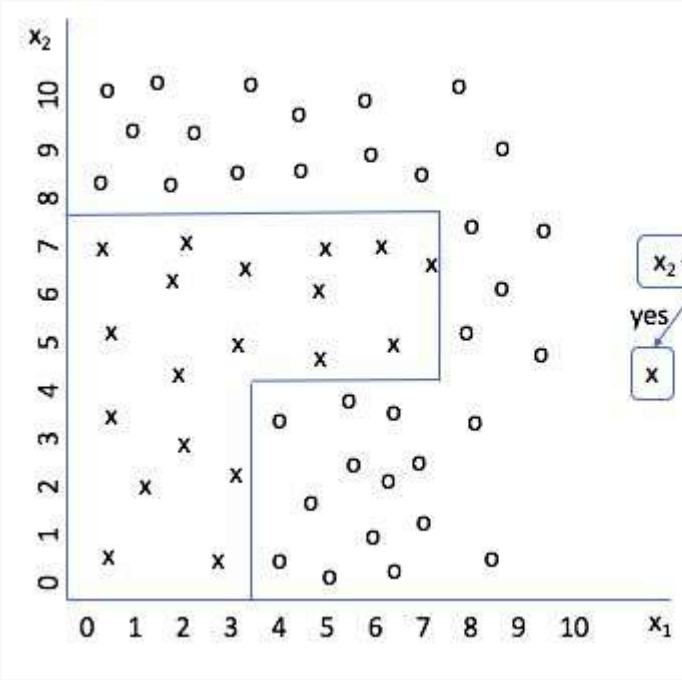
Árboles de Decisión

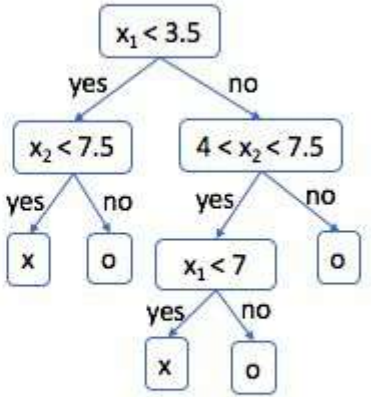


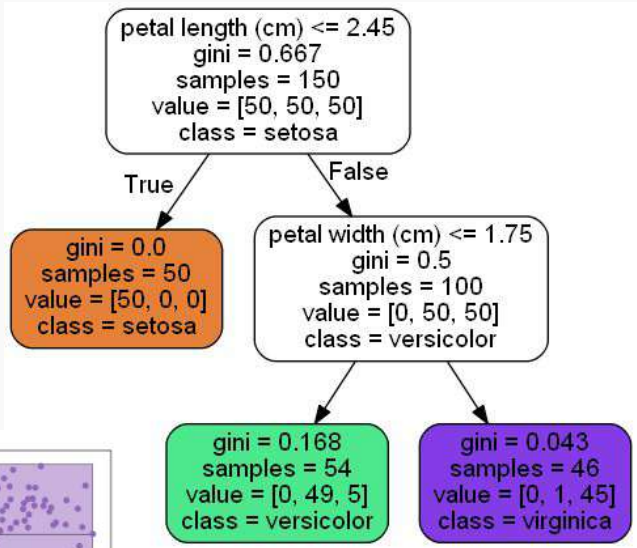


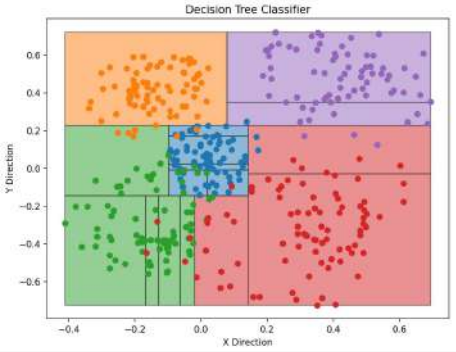

```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier

iris = load_iris(as_frame=True)
X_iris = iris.data[["petal length (cm)", "petal width (cm)"]].values
y_iris = iris.target

tree_clf = DecisionTreeClassifier(max_depth=2, random_state=42)
tree_clf.fit(X_iris, y_iris)
```

Árboles de Decisión









Hiper parámetros

Visualización

Árboles de Decisión

```
from sklearn.datasets import make_moons

X_moons, y_moons = make_moons(n_samples=150, noise=0.2, random_state=42)

tree_clf1 = DecisionTreeClassifier(random_state=42)
tree_clf2 = DecisionTreeClassifier(min_samples_leaf=5, random_state=42)
tree_clf1.fit(X_moons, y_moons)
tree_clf2.fit(X_moons, y_moons)
```

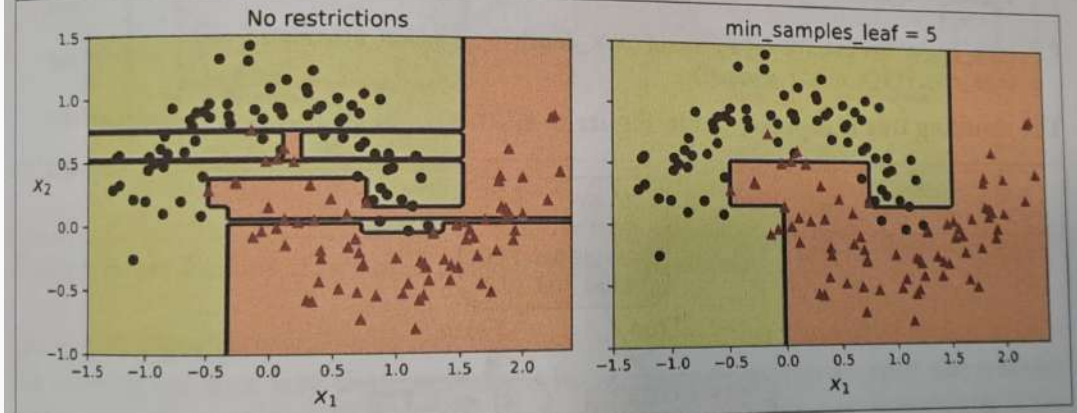
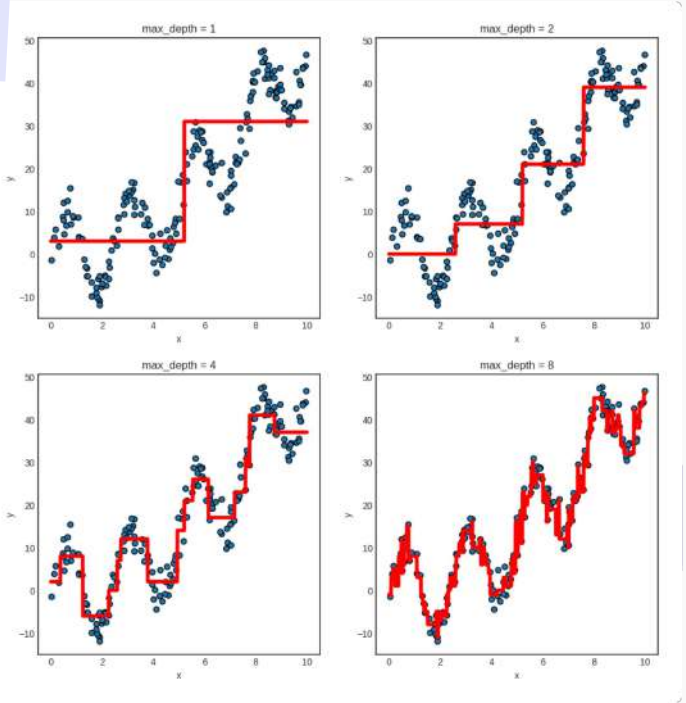


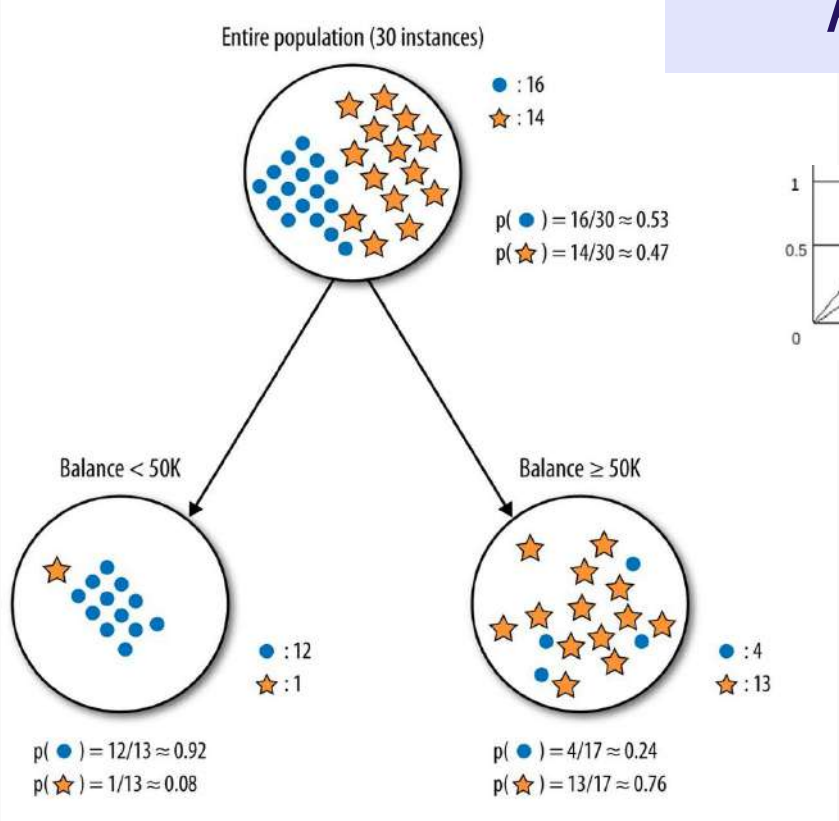
Figure 6-3. Decision boundaries of an unregularized tree (left) and a regularized tree (right)

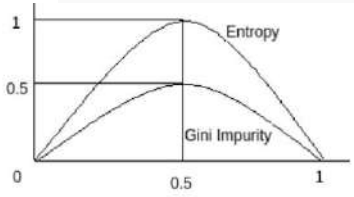
Lo mismo aplica para problemas de regresión.



¿Dónde se traza la frontera entre infra y sobre ajuste?

Árboles de Decisión





Impurity Criterion

Gini Index

$$I_G = 1 - \sum_{j=1}^{c} p_j^2$$

p_j: proportion of the samples that belongs to class c for a particular node

Entropy

$$I_H = -\sum_{j=1}^{c} p_j log_2(p_j)$$

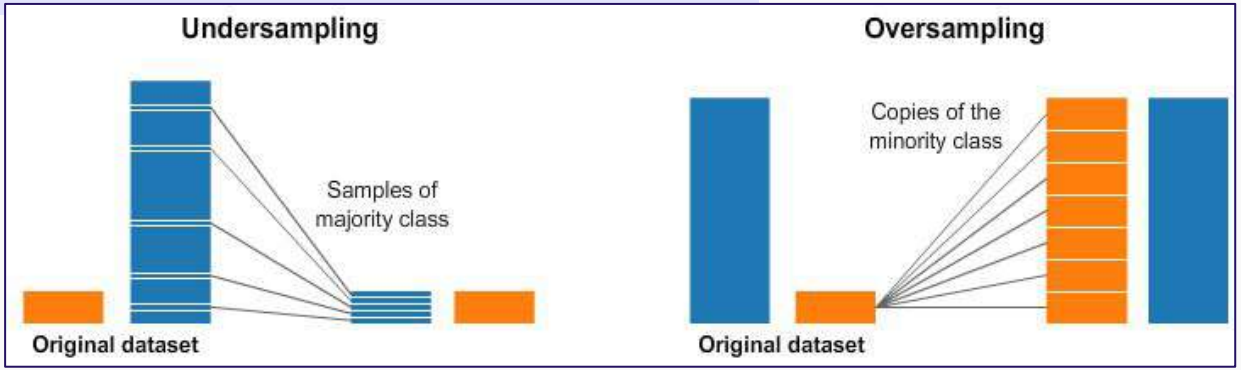
p_j: proportion of the samples that belongs to class c for a particular node.

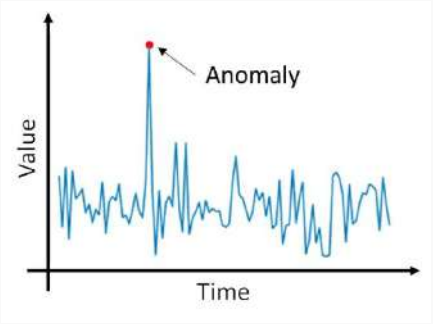
*This is the the definition of entropy for all non-empty classes ($p \neq 0$). The entropy is 0 if all samples at a node belong to the same class.

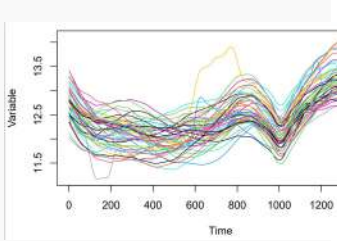
En la práctica, no hay mayor diferencia entre utilizar el índice de Gini o entropía: ambos producen árboles similares. Sin embargo, calcular Gini es más rápido.

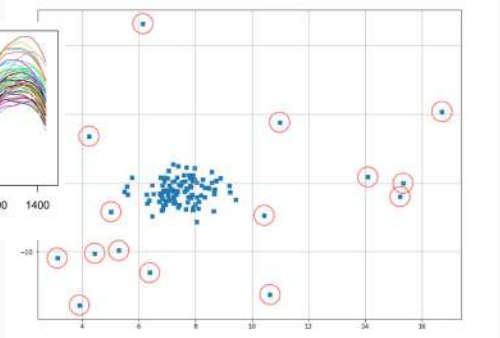
Cuando difieren, Gini tiende a apartar la clase mayoritaria en una sola rama, mientras que entropía tiende a generar árboles más balanceados.

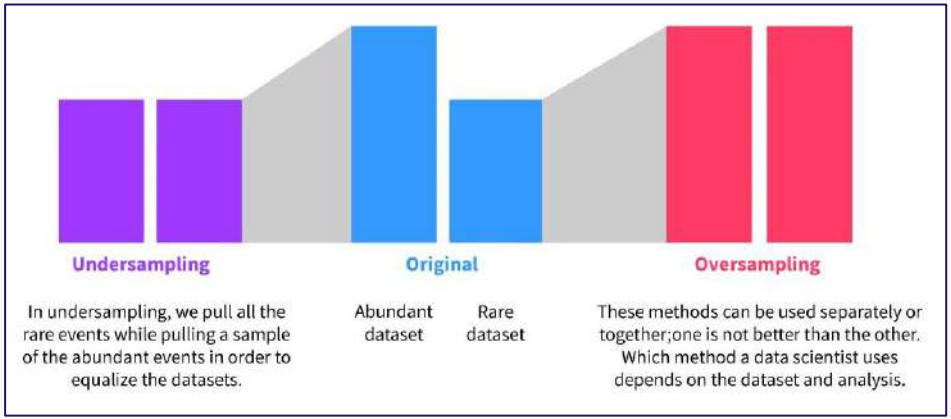
Datos no balanceados y detección de anomalías





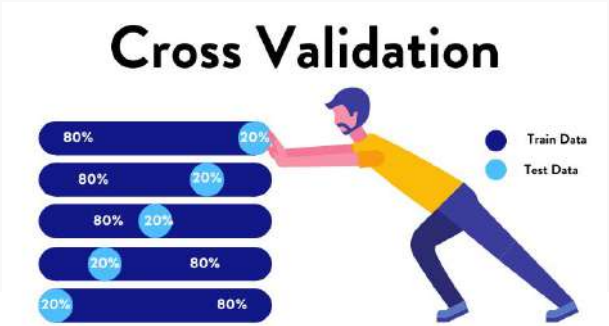






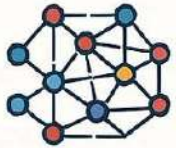
Otras estrategias útiles

- Selección de hyper-parámetros.
- Validación cruzada:



Cross-Validation

MACHINE LEARNING DEEP LEARNING MODEL EVALUATION



Generalization

K-Fold Cross-Validation

dataset split into K equal

↳ dataset split into K equal folds

Generalization

LOOCV

single-sample test sets iterated

Leave-P-Out CV

p-sample test sets with all possible

Stratified K-Fold CV

folds maintaining class proportions

p-sample test sets with all possible combinations

Time Series CV

Sliding Window

Expanding Window

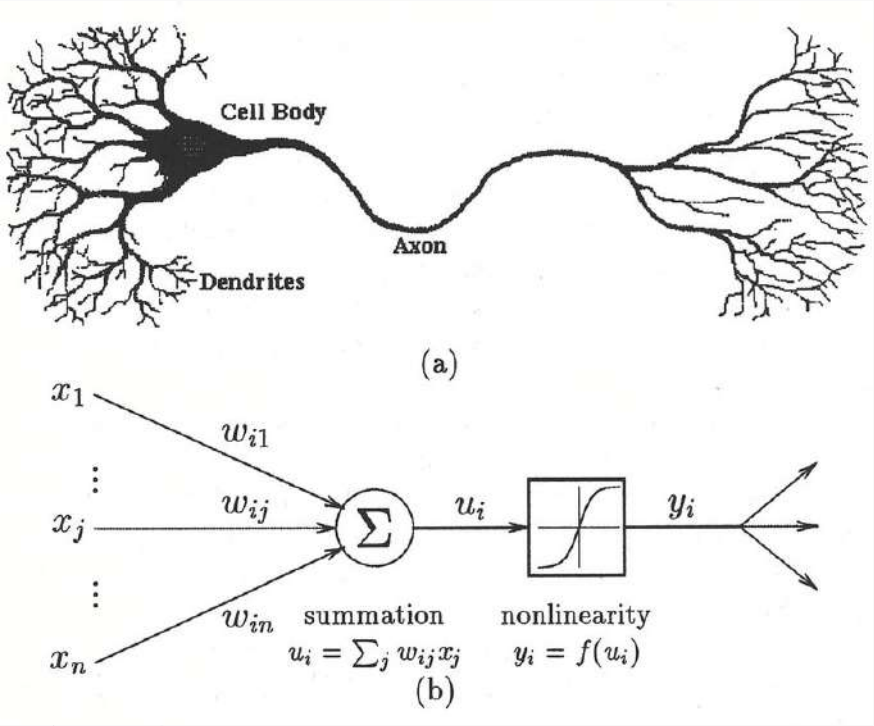
sequential train-test splits →

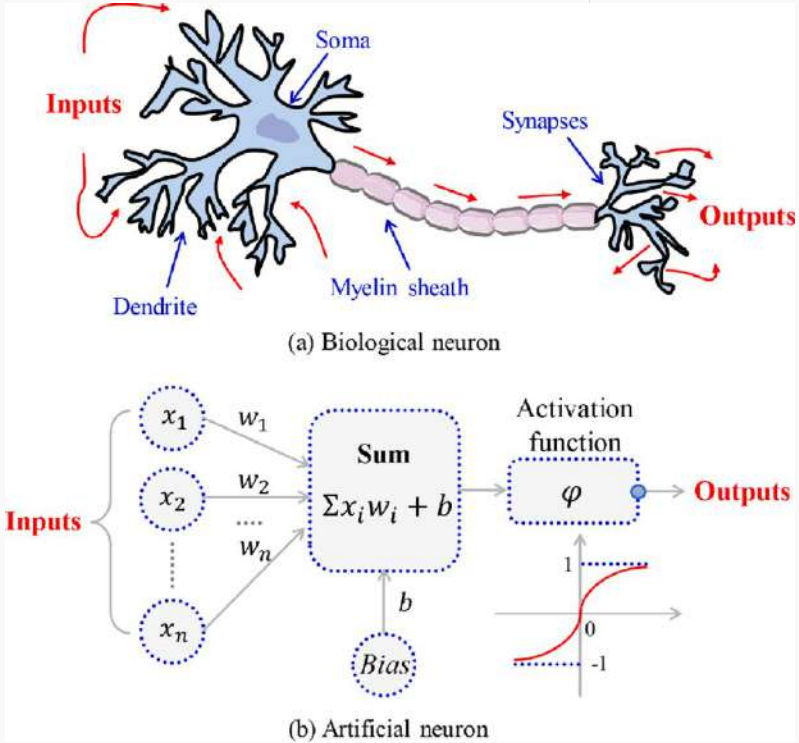
3.1

Deep Learning

¿Red de neuronas artificiales?

Neurona biológica vs neurona artificial

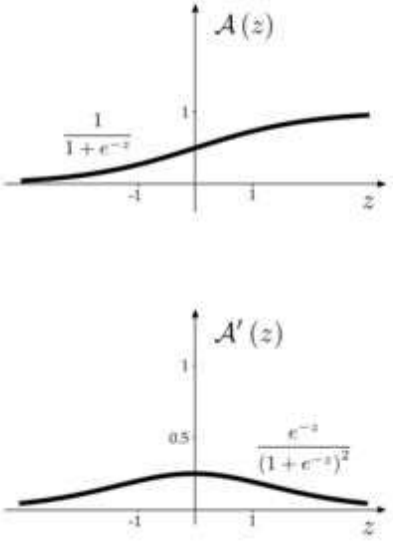




Funciones de activación comunes

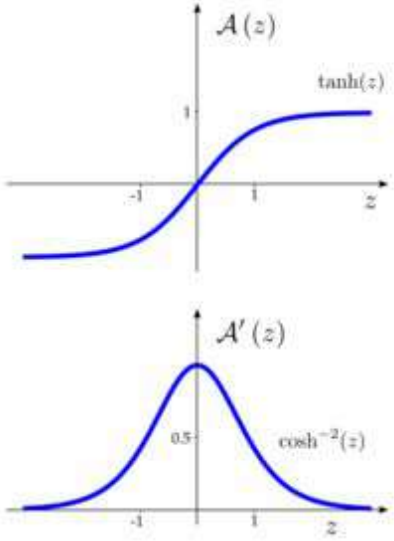
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



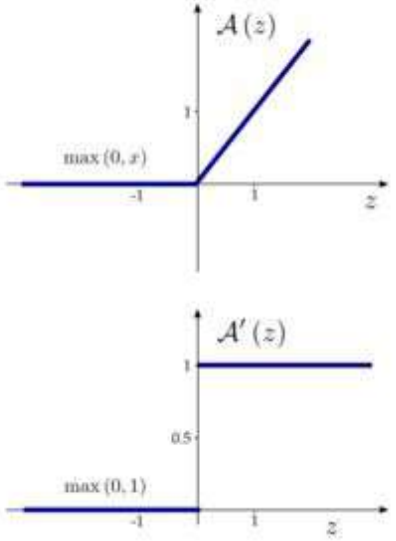
tanh

$$\tanh(x)$$

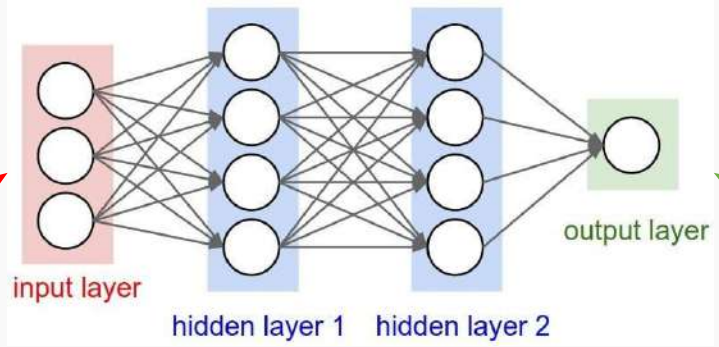


ReLU

$$\max(0, x)$$



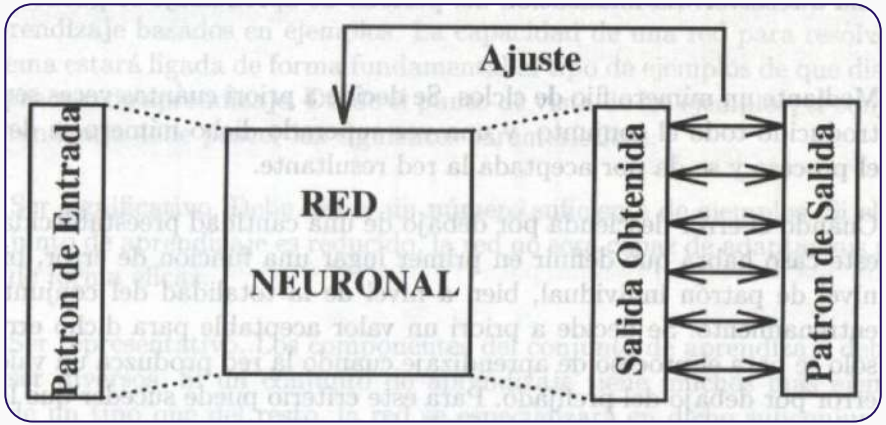
Esquema de una red de 4 capas totalmente interconectadas



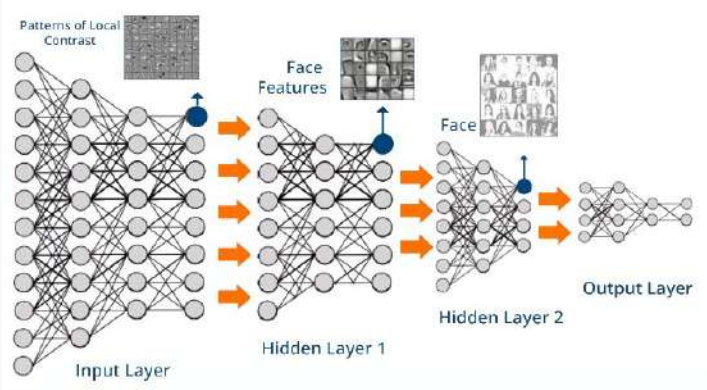

```
dataset.head()
```

	Price	Moneyness	Time	Vol
0	0.211	0.934	0.660	0.266
1	0.038	1.011	0.065	0.207
2	0.163	1.187	0.883	0.292
3	0.091	1.245	0.648	0.271
4	0.193	0.813	0.125	0.216

Aprendizaje supervisado

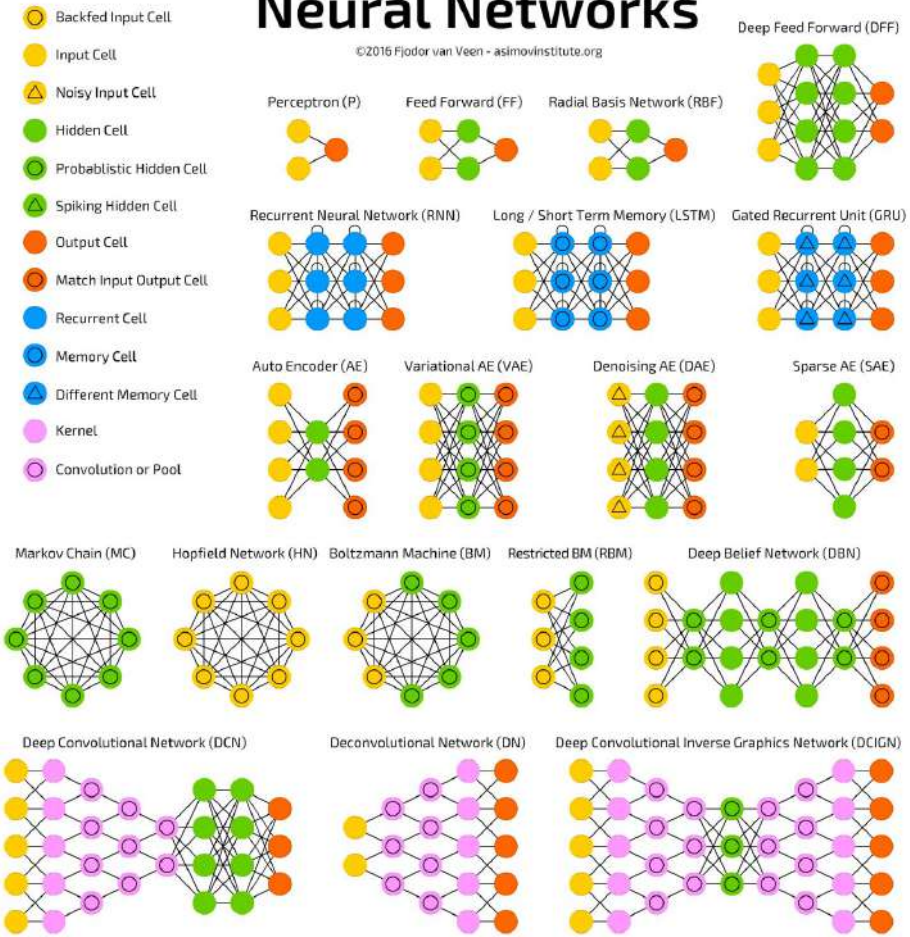


¿Por qué se le llama **aprendizaje profundo**?

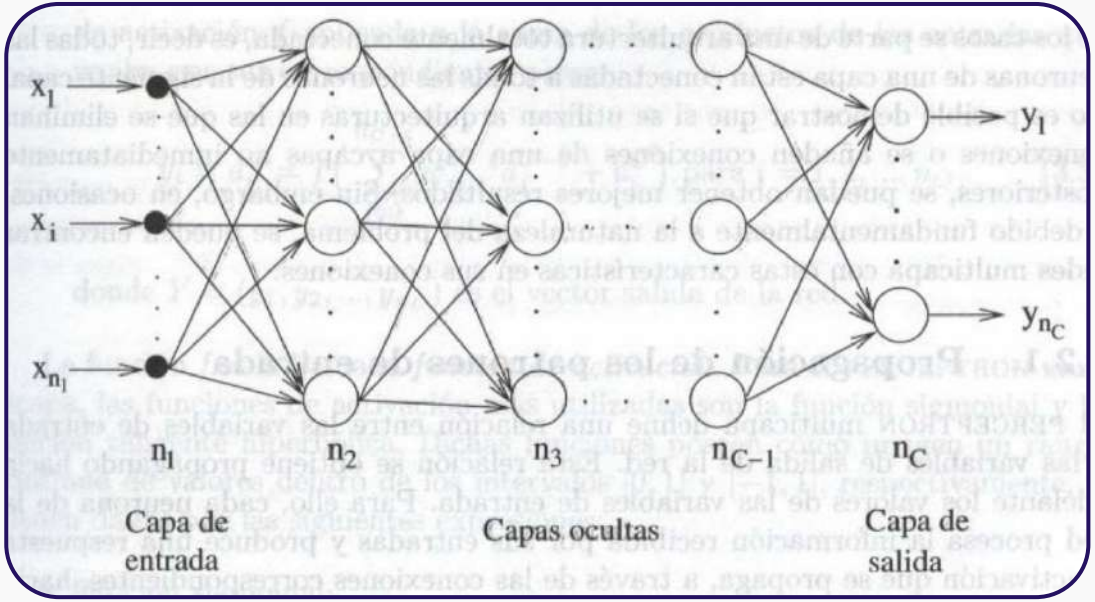


A mostly complete chart of

Neural Networks



Arquitectura de un perceptrón multicapa



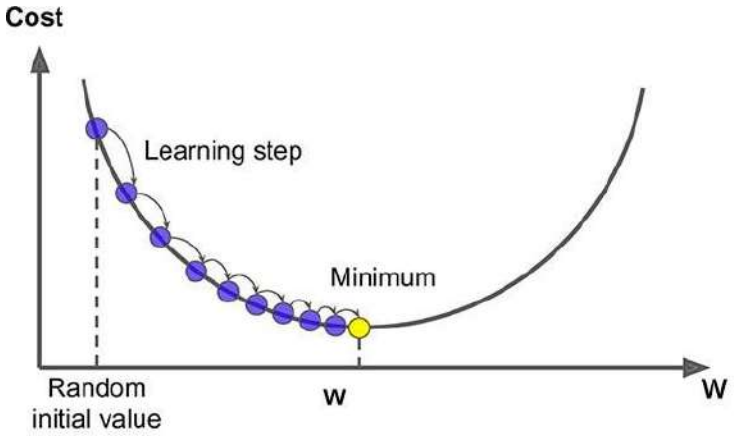
- Capa de normalización.
- Capa de entrada.
- Capas ocultas

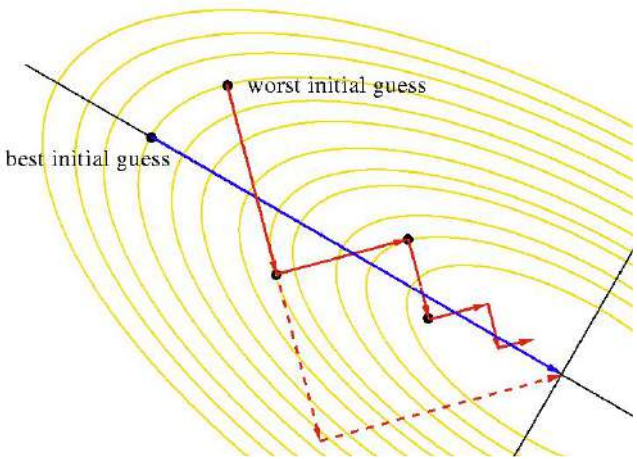
  (o de aprendizaje).
  - Cantidad de neuronas.
  - Función de activación.
- Capa de salida.
- Capa de desnormalización.
- Capa delimitadora.

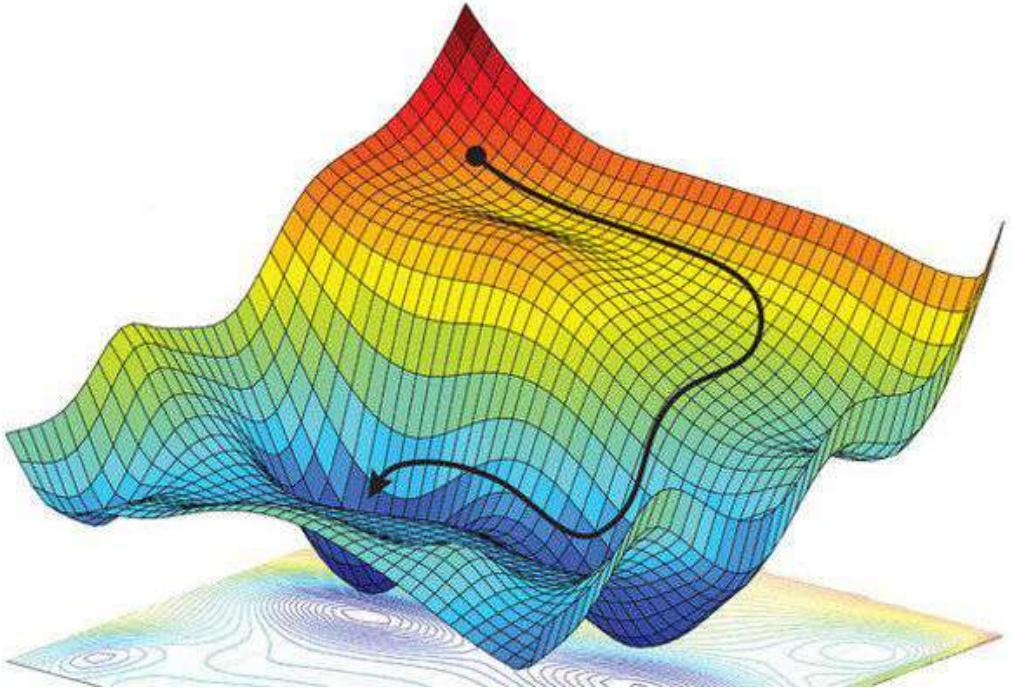
Proceso de aprendizaje

1. Inicializar pesos y umbrales con valores aleatorios.
2. Introducir una muestra “n” del conjunto de entrenamiento, obteniendo la respuesta de la red.
3. Obtener el error producido al predecir dicha muestra.
4. Aplicar el Algoritmo de RetroPropagación para modificar los pesos y umbrales de la red.
5. Repetir (2), (3) y (4) para todas las muestras del conjunto de entrenamiento (iteración / ciclo de aprendizaje).
6. Criterios de paro: error total E de la red ≈ 0, parámetros de la red estables (norma del gradiente final ≈ 0), cantidad máxima de iteraciones o tiempo máximo transcurrido.
7. Terminar el proceso de aprendizaje y dar como salida la red obtenida.

Aprendizaje mediante Descenso del Gradiente







Idea intuitiva del proceso de aprendizaje del Perceptrón Multicapa:

Partiendo de un punto aleatorio $W(0)$ del espacio $\mathbf{R}^{n_w}$, donde n_w es el número de parámetros de la red -pesos más umbrales-, el proceso de aprendizaje desplaza el vector de parámetros $W(n-1)$ en el espacio $\mathbf{R}^{n_w}$ siguiendo la dirección negativa del gradiente del error en dicho punto, alcanzando así un nuevo punto en dicho espacio, $W(n)$, que estará más próximo al mínimo de la función error que el anterior. El proceso continúa hasta que se encuentre un mínimo de la función error E, lo cual sucede cuando $\frac{\partial E}{\partial w} \approx 0$. En este momento, y de acuerdo con la Ecuación (3.10), los parámetros dejan de sufrir cambios significativos de una iteración a otra y el proceso de aprendizaje finaliza.

$$w(n) = w(n-1) - \alpha \frac{\partial e(n)}{\partial w}$$

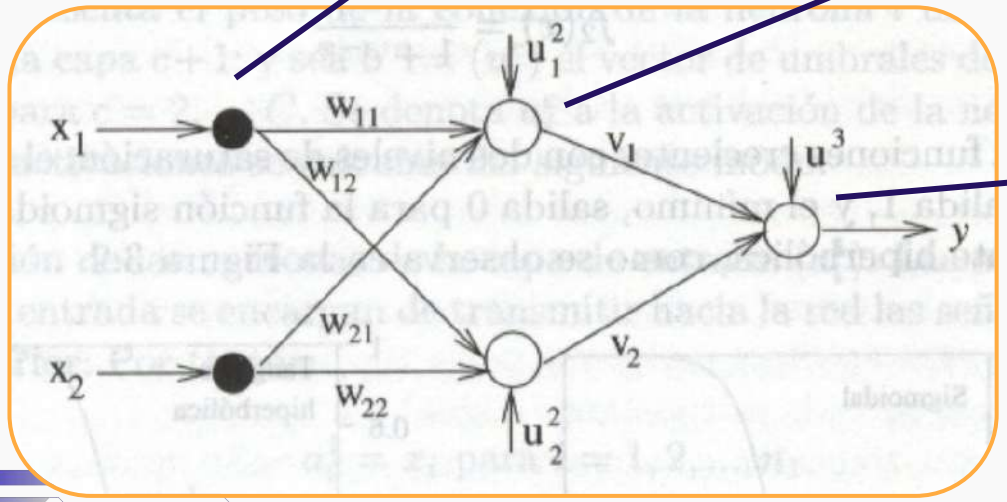
Activaciones de un Perceptrón Multicapa con dos neuronas de entrada, dos ocultas y una salida

Activaciones de las neuronas de entrada:

$$a_1^1 = x_1 \text{ y } a_2^1 = x_2$$

Activaciones de las neuronas ocultas:

$$a_1^2 = f\left(w_{11}a_1^1 + w_{21}a_2^1 + u_1^2\right) \text{ y } a_2^2 = f\left(w_{12}a_1^1 + w_{22}a_2^1 + u_2^2\right)$$



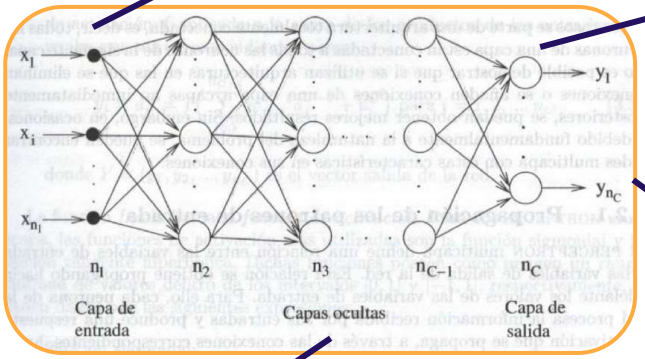
Activación de la neurona de salida:

$$y = a_1^3 = f\left(v_1a_1^2 + v_2a_2^2 + u^3\right)$$

Propagación de patrones de entrada (generalización)

$$X = (x_1, x_2, ..., x_{n_1})$$

$$a_i^1 = x_i \text{ para } i = 1, 2, ..., n_1$$



$$y_i = a_i^C = f\left(\sum_{j=1}^{n_{C-1}} w_{ji}^{C-1} a_j^{C-1} + u_i^C\right) \text{ para } i = 1, 2, ..., n_C$$

$$Y = (y_1, y_2, ..., y_{n_C})$$

$$f_1(x) = \frac{1}{1 + e^{-x}}$$

$$a_i^c = f\left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c\right) \text{ para } i = 1, 2, ..., n_c \text{ y } c = 2, 3..., C-1$$

https://playground.tensorflow.org/

Deep Playground

Comprendiendo las redes de neuronas artificiales

3.2

MLPs

Mi primer “Aproximador universal”

Bike Sharing

Donated on 12/19/2013

This dataset contains the hourly and daily count of rental bikes between years 2011 and 2012 in Capital bikeshare system with the corresponding weather and seasonal information.

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate	Social Science	Regression
Feature Type	**# Instances**	**# Features**
Integer, Real	17389	13

Dataset Information

Additional Information

Bike sharing systems are new generation of traditional bike rentals where whole process from membership, rental and return back has become automatic. Through these systems, user is able to easily rent a bike from a particular position and return back at another position. Currently, there are about over 500 bike-sharing programs around the world which is composed of over ...

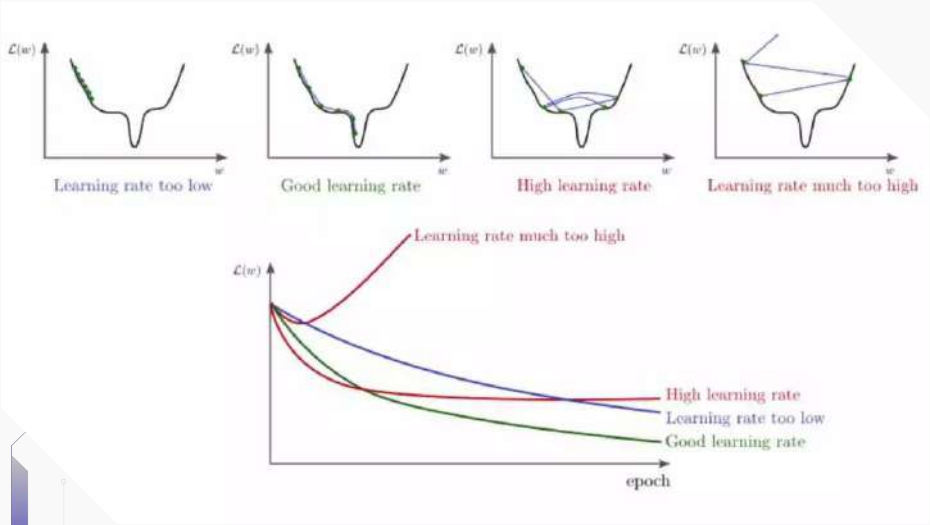
SHOW MORE

Has Missing Values?

No

Enlace

Hiper-parámetros para MLP Regressor

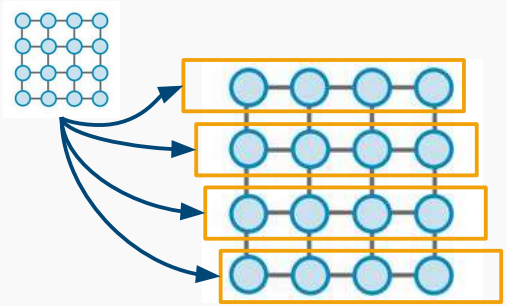


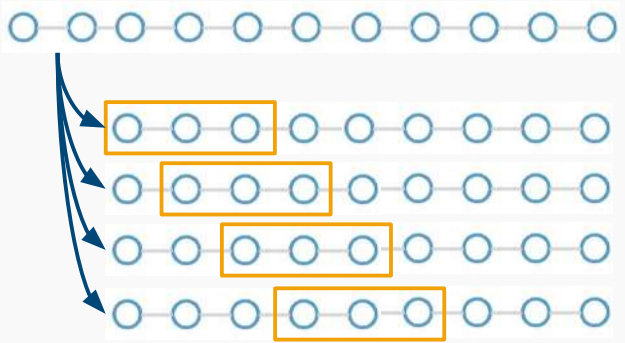
- **Función de activación (activation).**
- **Algoritmo de optimización (solver).**
- **Learning rate (learning_rate).**
- **Cantidad de iteraciones (max_iter).**
- **Inercia (momentum).**
- **Batch size (batch_size).**
- **Early stopping (early_stopping).**

Enlace

4.1

Series temporales

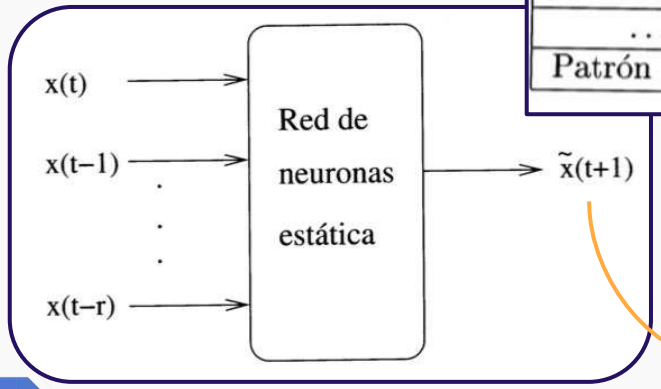




Predicción de series temporales: un paso en el futuro

$$\tilde{x}(t+1) = \tilde{F}\big(x(t), x(t-1), ..., x(t-r)\big)$$

	Entrada	Salida deseada
Patrón 1	$x(r), x(r-1), ..., x(1), x(0)$	$x(r+1)$
Patrón 2	$x(r+1), x(r), ..., x(2), x(1)$	$x(r+2)$
Patrón 3	$x(r+2), x(r+1), ..., x(3), x(2)$	$x(r+3)$
Patrón 4	$x(r+3), x(r+2), ..., x(4), x(3)$	$x(r+4)$
...	...	...
Patrón $N-r$	$x(N-1), x(N-2), ..., x(N-r), x(N-(r+1))$	$x(N)$



Time step	Value
1	10
2	11
3	18
4	15
5	20

X	Y
?	10
10	11
11	18
18	15
15	20
20	?

Con r = 1

X	Y
10,11	18
11,18	15
18,15	20

Con r = 2

X	Y
10,11,18	15
11,18,15	20

Con r = 3

Individual Household Electric Power Consumption

Donated on 8/29/2012

Measurements of electric power consumption in one household with a one-minute sampling rate over a period of almost 4 years. Different electrical quantities and some sub-metering values are available.

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate, Time-Series	Physics and Chemistry	Regression, Clustering
Feature Type	**# Instances**	**# Features**
Real	2075259	9

Dataset Information

Additional Information

This archive contains 2075259 measurements gathered in a house located in Sceaux (7km of Paris, France) between December 2006 and November 2010 (47 months).

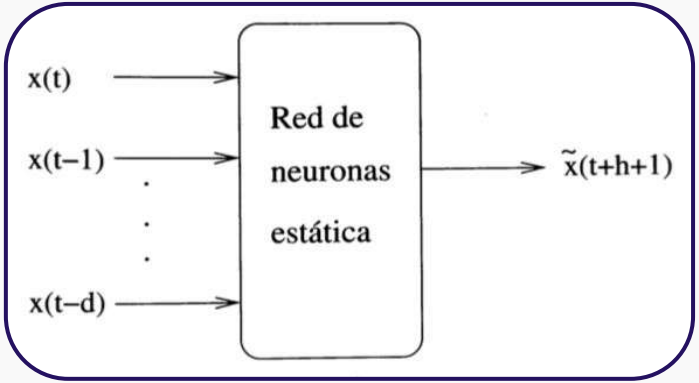
Notes:

1.(global_active_power*1000/60 - sub_metering_1 - sub_metering_2 - sub_metering_3) represents the active energy consumed every minute (in watt hour) in the household by electrical equipment not measured in sub-meterings 1, 2 and 3.

2.The dataset contains some missing values in the measurements (nearly 1,25% of the rows). All calendar timestamps are present in the dataset but for some timestamps, the measurement values are missing: a missing value is represented by the

Enlace

Predicción de series temporales: múltiples pasos en el futuro



$$\tilde{x}(t+h+1) = \tilde{G}\big(x(t), x(t-1), ..., x(t-d)\big)$$

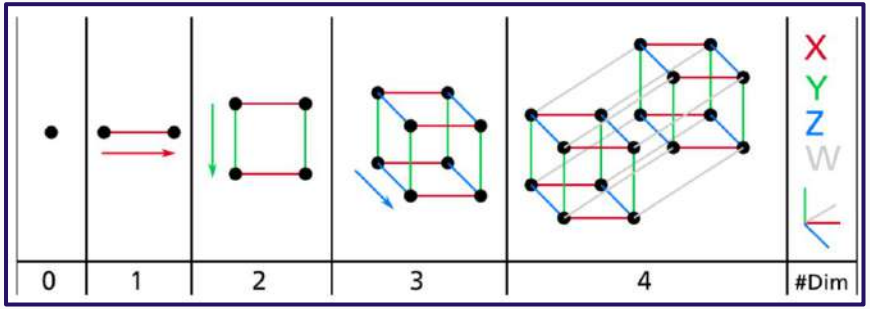
	Entrada	Salida deseada
Patrón 1	$x(d), x(d-1), ..., x(1), x(0)$	$x(d+h+1)$
Patrón 2	$x(d+1), x(d), ..., x(2), x(1)$	$x(d+h+2)$
Patrón 3	$x(d+2), x(d+1), ..., x(3), x(2)$	$x(d+h+3)$
Patrón 4	$x(d+3), x(d+2), ..., x(4), x(3)$	$x(d+h+4)$
...	...	...
Patrón N-d-h	$x(N-h-1), x(N-h-2), ..., x(N-1-h-d)$	$x(N)$

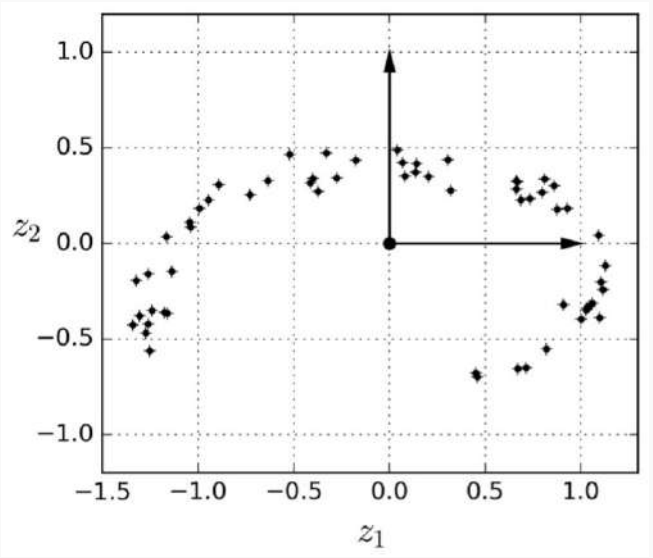
4.2

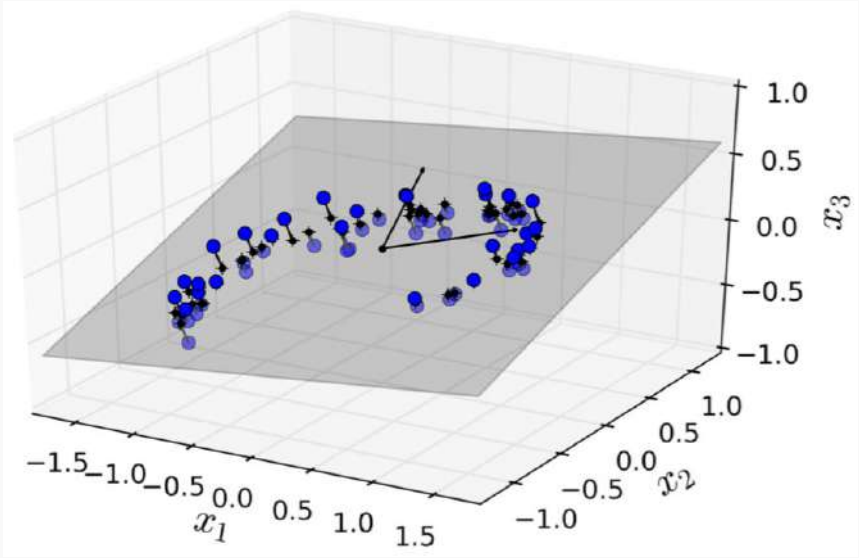
PCA & ML

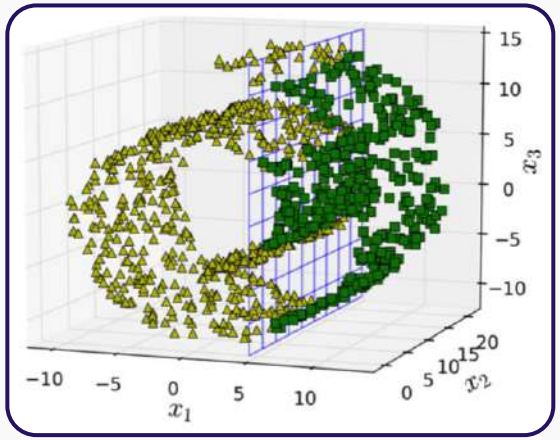
Reducción de la dimensionalidad

¿A qué nos referimos?



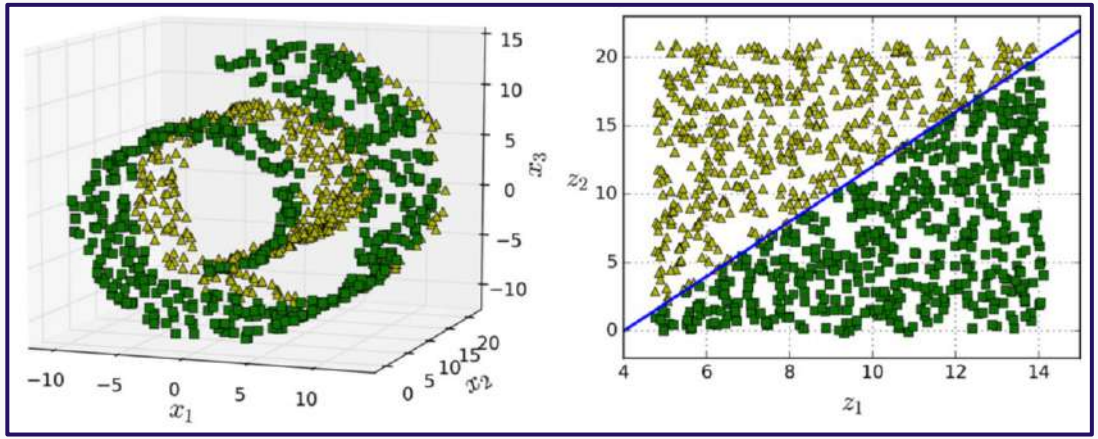






A veces no será necesario

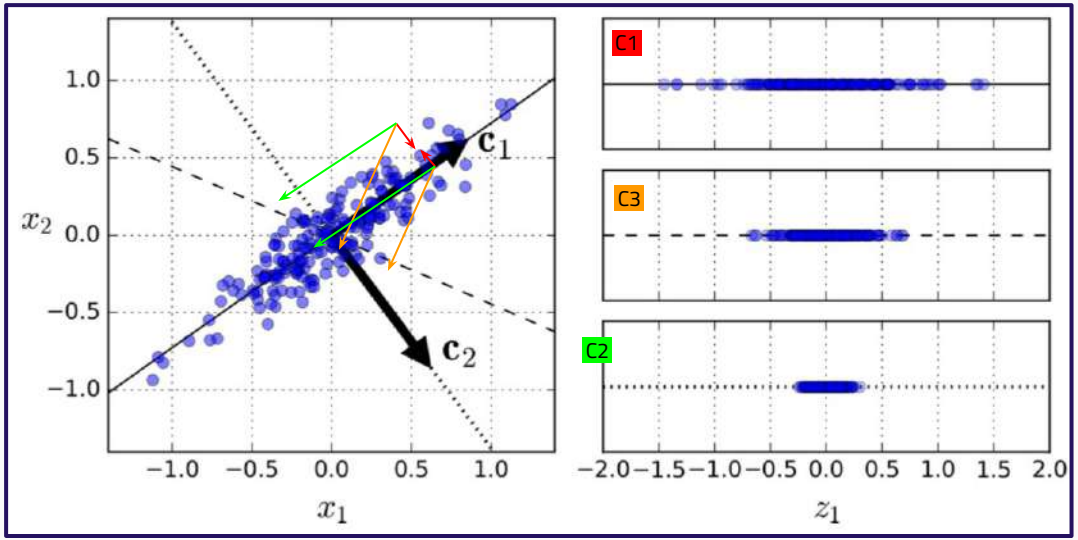
¿Por qué hacerlo en ML?

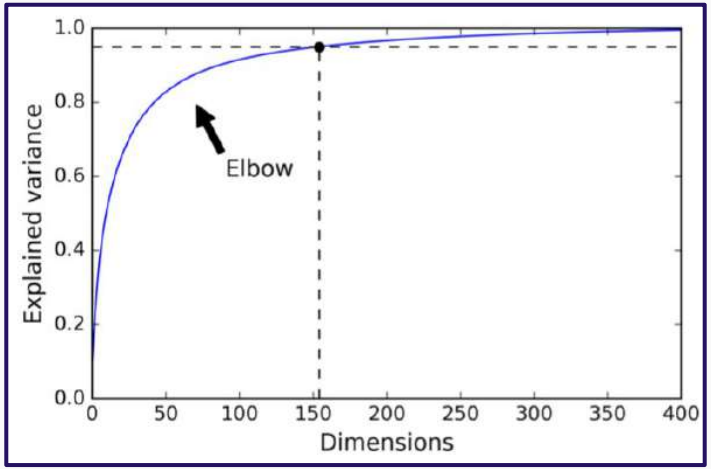


A veces nos ayudará a encontrar un modelo más preciso o ligero.

No basta con reducir dimensiones

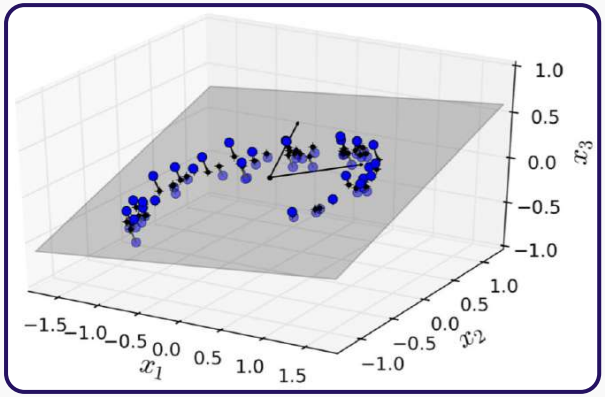
El algoritmo se encarga de encontrar la dimensión que retenga la mayor varianza entre los datos. Siempre se perderá información, pero se desea perder la menor cantidad posible.





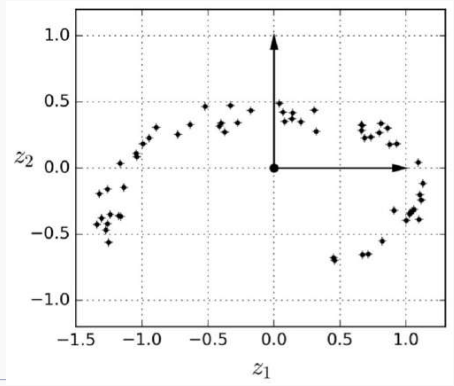
Se desea seleccionar la menor cantidad de dimensiones, sin perder mucha información.

Hiperparámetro: n_components




```
from sklearn.decomposition import PCA

pca = PCA(n_components = 2)
X2D = pca.fit_transform(X)
```

```
>>> print(pca.explained_variance_ratio_)
array([ 0.84248607,  0.14631839])
```

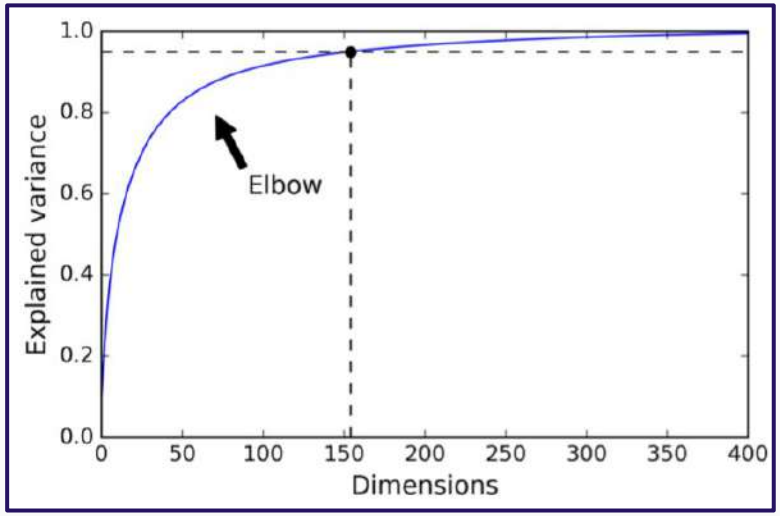


Para fijar una cantidad de dimensiones, hay que proveer un número natural mayor a uno

El 1.2% restante es el “precio a pagar” por buscar un modelo más sencillo de entrenar

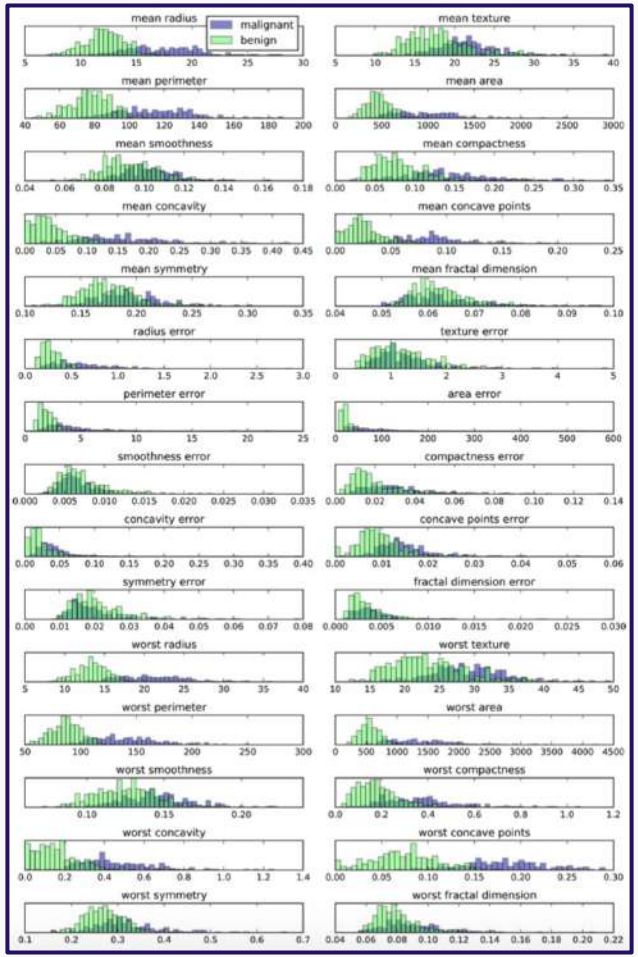
Hiperparámetro: n_components

En cambio, si lo que nos interesa es preservar cierto porcentaje de varianza (lo más utilizado)




```
pca = PCA(n_components=0.95)
X_reduced = pca.fit_transform(X)
```

Entonces se debe proveer un número real en el rango entre cero y uno



Ejemplo de aplicación: detección de cáncer de mama

```
from sklearn.datasets import load_breast_cancer
cancer = load_breast_cancer()

scaler = StandardScaler()
scaler.fit(cancer.data)
X_scaled = scaler.transform(cancer.data)
```

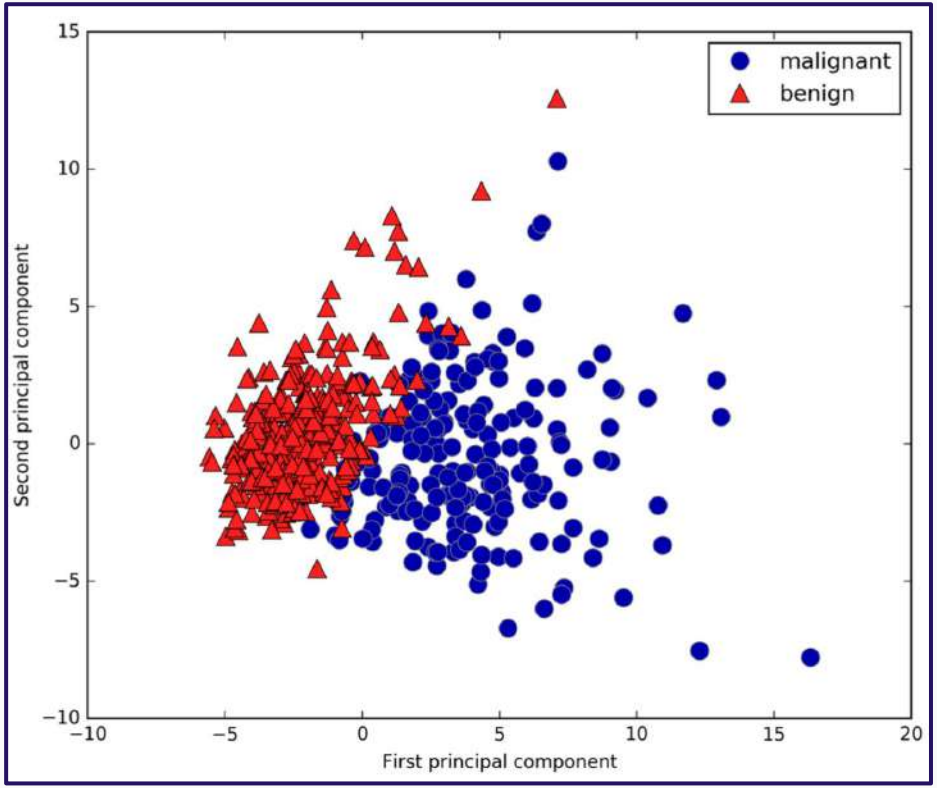
30 variables de entrada

¡Con solo 2 componentes, tenemos un muy buen conjunto de datos!

```python
from sklearn.decomposition import PCA
# keep the first two principal components of the data
pca = PCA(n_components=2)
# fit PCA model to breast cancer data
pca.fit(X_scaled)

# transform data onto the first two principal components
X_pca = pca.transform(X_scaled)
print("Original shape: {}".format(str(X_scaled.shape)))
print("Reduced shape: {}".format(str(X_pca.shape)))
```

```
Original shape: (569, 30)
Reduced shape: (569, 2)
```

```python
# plot first vs. second principal component, colored by class
plt.figure(figsize=(8, 8))
mglearn.discrete_scatter(X_pca[:, 0], X_pca[:, 1], cancer.target)
plt.legend(cancer.target_names, loc="best")
plt.gca().set_aspect("equal")
plt.xlabel("First principal component")
plt.ylabel("Second principal component")
```



Breast Cancer Wisconsin (Diagnostic)

Donated on 10/31/1995

Enlace

Diagnostic Wisconsin Breast Cancer Database.

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate	Health and Medicine	Classification
Feature Type	**# Instances**	**# Features**
Real	569	30

Spambase

Donated on 6/30/1999

Enlace

Classifying Email as Spam or Non-Spam

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate	Computer Science	Classification
Feature Type	**# Instances**	**# Features**
Integer, Real	4601	57

4.3

Clusterización

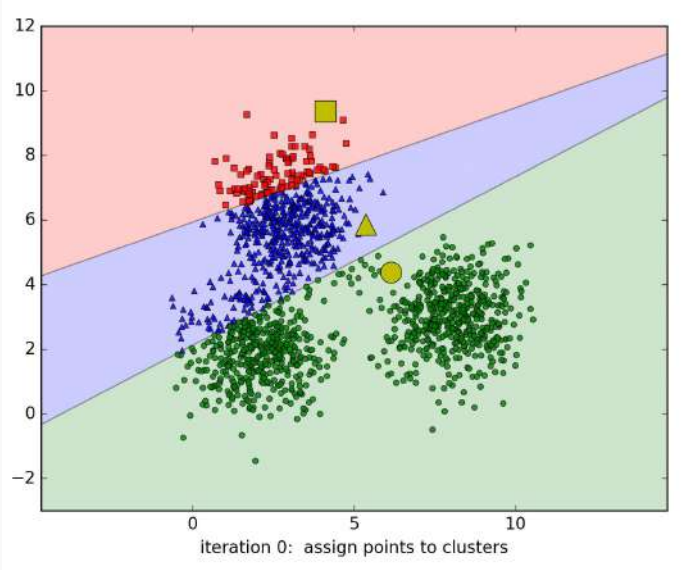
Aprendizaje no supervisado

Clusterización

El objetivo de la clusterización es **encontrar una forma natural de agrupar nuestros datos**, de tal manera que los datos que pertenecen a un subgrupo (llamado clúster) son **más similares entre sí** que con los datos de los demás subgrupos.

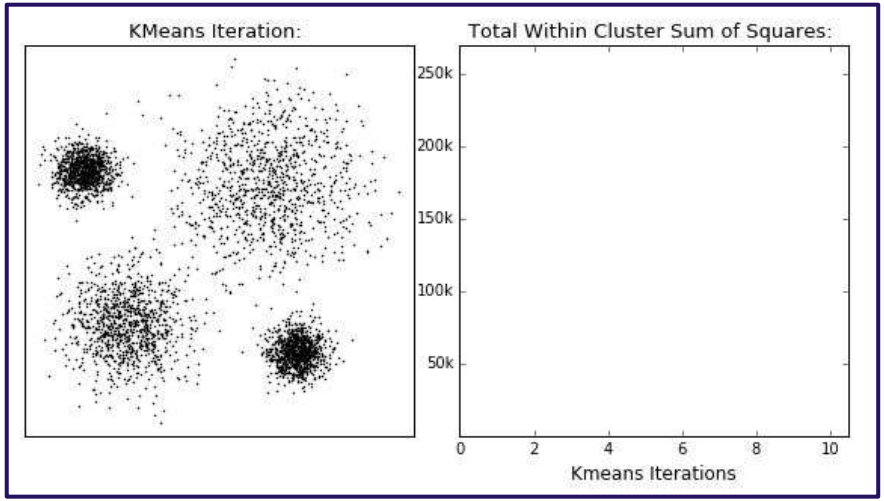
Es un algoritmo no supervisado, ya que **no se sabe de antemano** cuántos subgrupos (clusters) se formarán, ni cuántos datos pertenecerán a cada uno de ellos.

También permite la **categorización automática** de nuevos datos mediante la regla que ha aprendido.



Algoritmo de k-medias

El objetivo es **encontrar k centroides** y asignar uno de ellos a cada registro, de tal forma que se **minimice** la varianza intra-clúster (llamada **inercia**). Usualmente se utiliza la distancia Euclidiana (distancia entre dos puntos), pero es posible ocupar otras métricas.



K-medias se encarga de encontrar un mínimo para una “k” dada, de la siguiente manera:

1. **Se elige una cantidad de clústers.**
2. **Algunos puntos se eligen aleatoriamente como los centroides de los clústers.**
3. **Cada dato se asigna al clúster de cuyo centro esté más cercano.**
4. **El centroide del clúster se actualiza con la media de los puntos asignados a él.**
5. **Los pasos 3 al 4 se repiten hasta que todos los centroides se mantengan sin cambios.**

Hiper-parámetros de k-medias:

Cantidad de clusters

- Clusters (y centroides) a generar.

Cantidad máxima de iteraciones

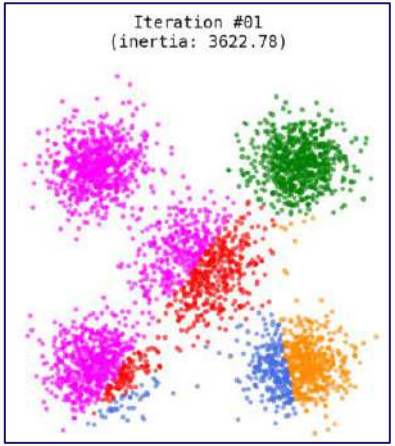
- Para **una ejecución** del algoritmo.

```
from sklearn.cluster import KMeans
#Fit with k-means
k_means = KMeans(n_clusters=nclust)
k_means.fit(X)
```

“Número inicial”

- Cantidad de veces que el algoritmo se ejecutará, utilizando **diferentes semillas** para la generación de los **centroides** El resultado final será el que mejor rendimiento haya tenido (en términos de la **inercia**).
- En la librería sklearn, el algoritmo se ejecuta **al menos 10 veces, con diferentes valores iniciales**.

Características de k-medias

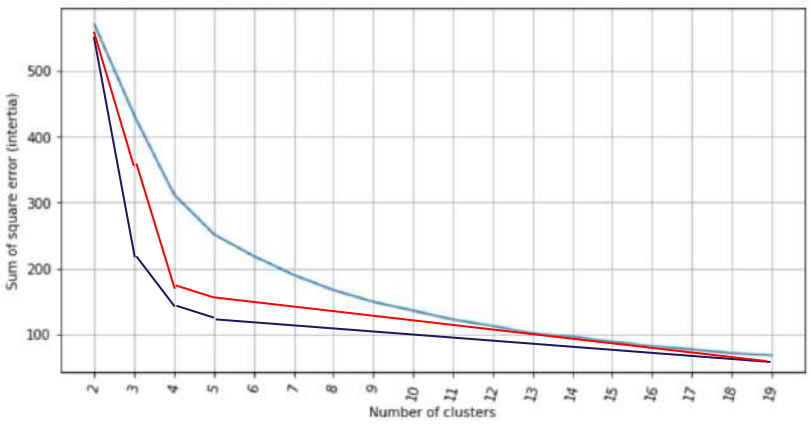


Ventajas

- Su principal ventaja radica en su simplicidad, su amplio rango de aplicabilidad, rápida velocidad de convergencia y su escalabilidad para conjuntos grandes de datos.

Desventajas

- Falta de garantía para encontrar el mínimo global (sin tener que ejecutar el algoritmo una considerable cantidad de veces).
- Es sensible a valores atípicos.



Wholesale customers

Donated on 3/30/2014

The data set refers to clients of a wholesale distributor. It includes the annual spending in monetary units (m.u.) on diverse product categories

Dataset Characteristics
Multivariate

Subject Area
Business

Associated Tasks
Classification, Clustering

Feature Type
Integer

 # Instances
440

 # Features
7

Dataset Information

Has Missing Values?
No

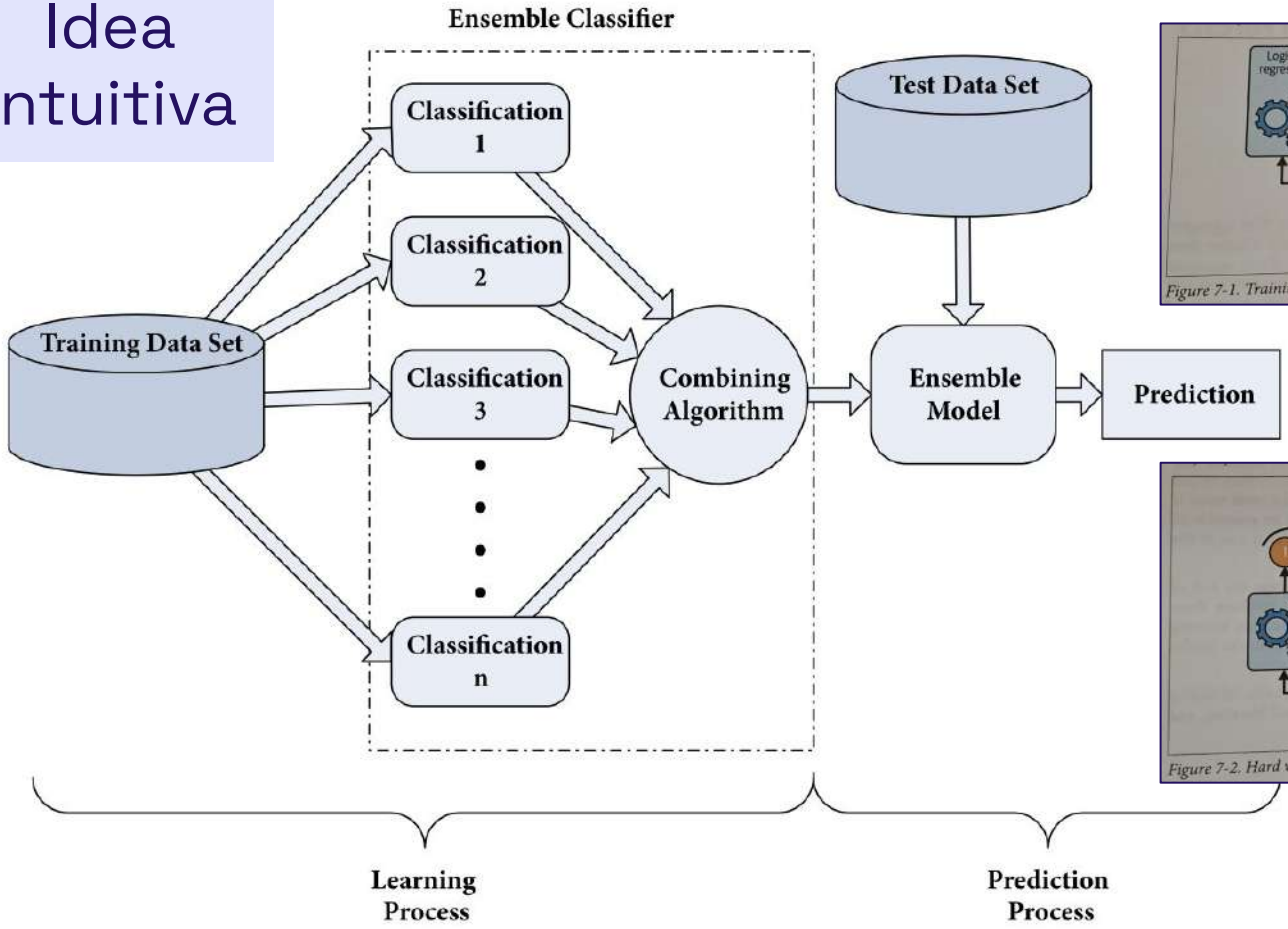
Enlace

5.1

Métodos de ensamble

Varias cabezas piensan mejor que una, ¿no?

Idea intuitiva



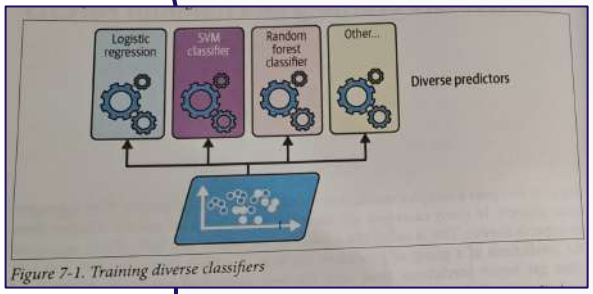


Figure 7-1. Training diverse classifiers

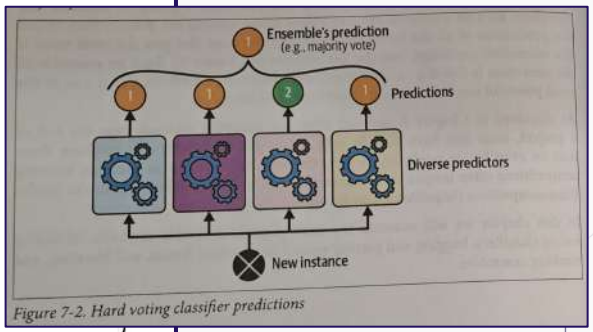


Figure 7-2. Hard voting classifier predictions

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import VotingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC

log_clf = LogisticRegression()
rnd_clf = RandomForestClassifier()
svm_clf = SVC()

voting_clf = VotingClassifier(
        estimators=[('lr', log_clf), ('rf', rnd_clf), ('svc', svm_clf)],
        voting='hard'
    )
voting_clf.fit(X_train, y_train)
```

```
LogisticRegression 0.864
RandomForestClassifier 0.872
SVC 0.888
VotingClassifier 0.896
```

```
>>> from sklearn.metrics import accuracy_score
>>> for clf in (log_clf, rnd_clf, svm_clf, voting_clf):
>>>     clf.fit(X_train, y_train)
>>>     y_pred = clf.predict(X_test)
>>>     print(clf.__class__.__name__, accuracy_score(y_test, y_pred))
```

Tipos de métodos de ensamble más utilizados:

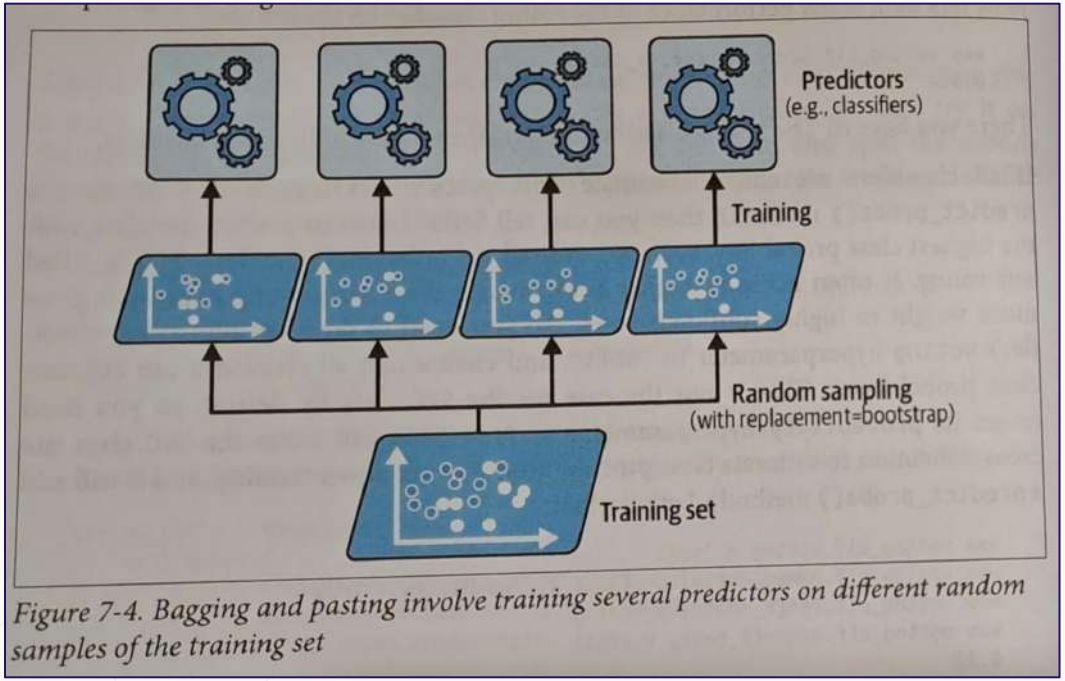


Figure 7-4. Bagging and pasting involve training several predictors on different random samples of the training set

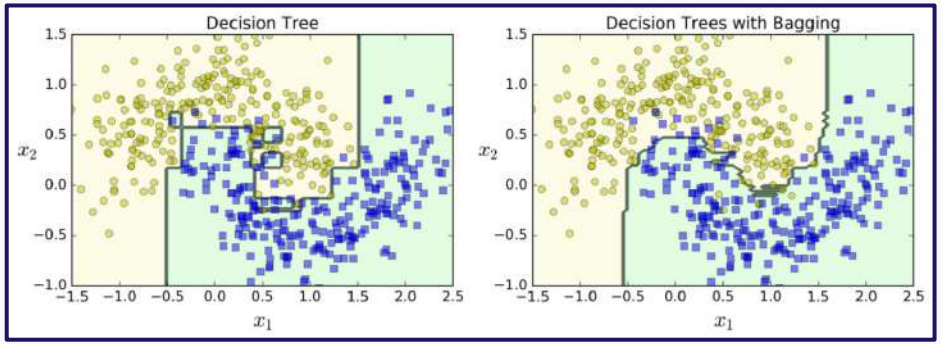
Utiliza **el mismo algoritmo** de entrenamiento para cada predictor, pero **entrenarlos en diferentes subconjuntos** aleatorios del conjunto de entrenamiento.
Cuando el muestreo se realiza con reemplazo, este método se denomina bagging (**abreviatura de bootstrap aggregating**).

Bagging

Una vez entrenados todos los predictores, el conjunto puede hacer una predicción para una nueva instancia **simplemente agregando las predicciones de todos los predictores**.

La función de agregación suele ser el modo estadístico (es decir, la predicción más frecuente, **como un clasificador de voto duro**) para la clasificación, o la media para la regresión. Cada predictor individual tiene un sesgo mayor que si se entrenase con el conjunto de entrenamiento original, pero la agregación reduce tanto el sesgo como la varianza. **En general, el resultado neto tiene un sesgo similar pero una varianza menor que un predictor individual entrenado en el conjunto de entrenamiento original.**

```
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier

bag_clf = BaggingClassifier(
        DecisionTreeClassifier(), n_estimators=500,
        max_samples=100, bootstrap=True, n_jobs=-1
    )
bag_clf.fit(X_train, y_train)
y_pred = bag_clf.predict(X_test)
```



Boosting

El boosting (originalmente llamado refuerzo de hipótesis) se refiere a cualquier método de ensamble que pueda combinar varios aprendices débiles en un aprendiz fuerte.

La idea general de la mayoría de los métodos de refuerzo es **entrenar predictores de forma secuencial, cada uno de los cuales intenta corregir a su predecesor.**

Existen muchos métodos de boosting, pero los más populares son **AdaBoost** (abreviatura de Adaptive Boosting) y **Gradient Boosting**.

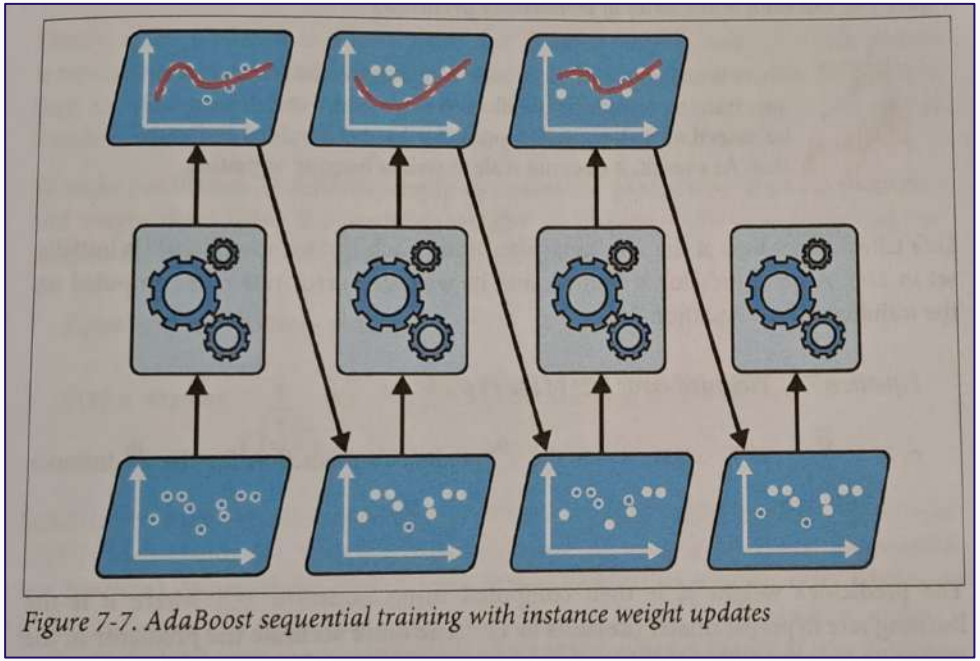


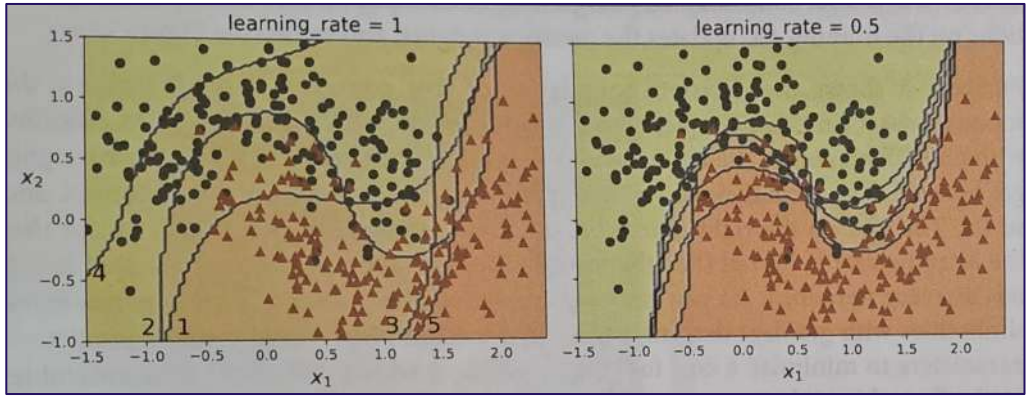
Figure 7-7. AdaBoost sequential training with instance weight updates

Una forma de que un nuevo predictor corrija a su predecesor es **prestar un poco más de atención a las instancias de entrenamiento que el predecesor infravaloró.** Esto hace que los nuevos predictores se centren cada vez más en los casos difíciles. Esta es la técnica utilizada por Ada-Boost.

```
from sklearn.ensemble import AdaBoostClassifier

ada_clf = AdaBoostClassifier(
        DecisionTreeClassifier(max_depth=1), n_estimators=200,
        algorithm="SAMME.R", learning_rate=0.5
    )
ada_clf.fit(X_train, y_train)
```

Por ejemplo, para construir un clasificador **AdaBoost**, se entrena un primer clasificador base (como un árbol de decisión) y se utiliza para hacer predicciones sobre el conjunto de entrenamiento. A continuación, **se aumenta el peso relativo de las instancias de entrenamiento mal clasificadas**.

Se entrena un segundo clasificador utilizando los pesos actualizados y, de nuevo, realiza predicciones sobre el conjunto de entrenamiento, se actualizan los pesos, y así sucesivamente.



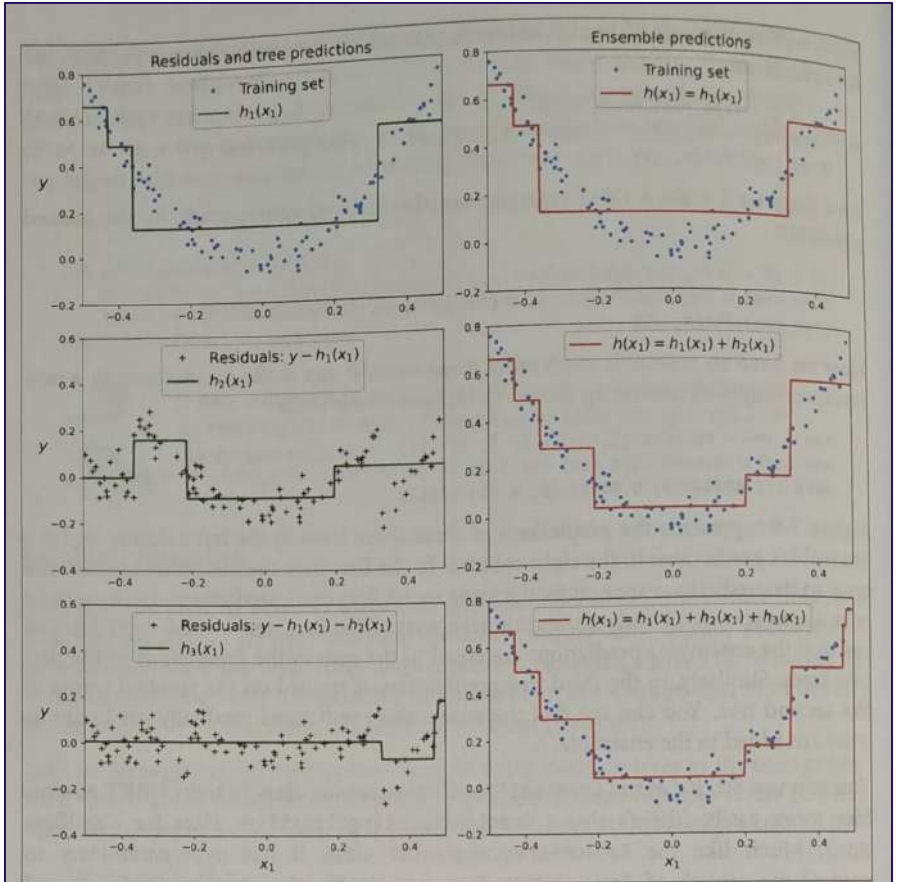


Figure 7-9. In this depiction of gradient boosting, the first predictor (top left) is trained normally, then each consecutive predictor (middle left and lower left) is trained on the previous predictor's residuals; the right column shows the resulting ensemble's predictions

Otro algoritmo de Boosting muy popular es el **Gradient Boosting.** Al igual que AdaBoost, Gradient Boosting funciona añadiendo secuencialmente predictores a un conjunto, cada uno de los cuales corrige a su predecesor.

Sin embargo, en lugar de ajustar los pesos de las instancias en cada iteración, como hace AdaBoost, **este método intenta ajustar el nuevo predictor a los errores residuales cometidos por el predictor anterior**.

```python
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

X_train, X_val, y_train, y_val = train_test_split(X, y)

gbrt = GradientBoostingRegressor(max_depth=2, n_estimators=120)
gbrt.fit(X_train, y_train)

errors = [mean_squared_error(y_val, y_pred)
          for y_pred in gbrt.staged_predict(X_val)]
bst_n_estimators = np.argmin(errors)

gbrt_best = GradientBoostingRegressor(max_depth=2,n_estimators=bst_n_estimators)
gbrt_best.fit(X_train, y_train)
```

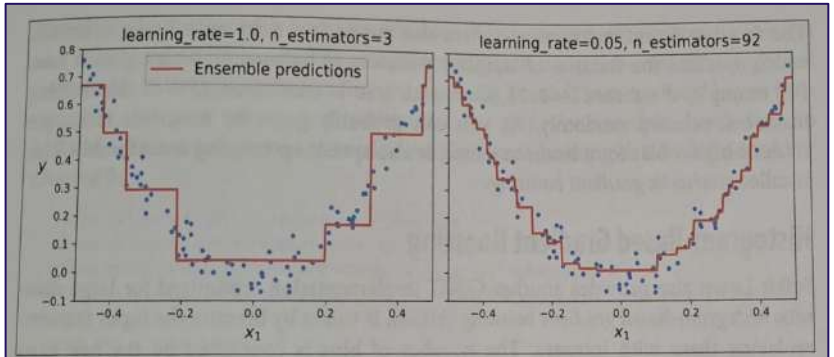


Figure 7-10. GBRT ensembles with not enough predictors (left) and just enough (right)

Enlace

HistGradientBoostingClassifier

```
class sklearn.ensemble.HistGradientBoostingClassifier(loss='log_loss', *,
learning_rate=0.1, max_iter=100, max_leaf_nodes=31, max_depth=None,
min_samples_leaf=20, l2_regularization=0.0, max_features=1.0, max_bins=255,
categorical_features='warn', monotonic_cst=None, interaction_cst=None,
warm_start=False, early_stopping='auto', scoring='loss', validation_fraction=0.1,
n_iter_no_change=10, tol=1e-07, verbose=0, random_state=None, class_weight=None)
```

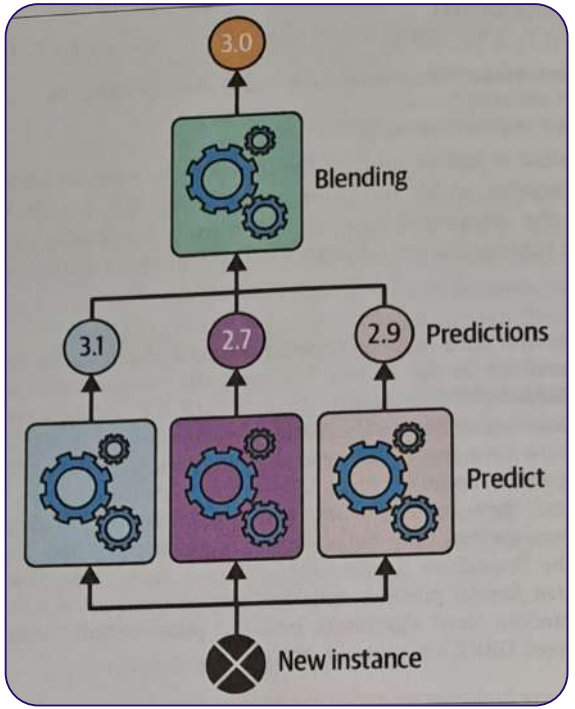
Histogram-based Gradient Boosting Classification Tree. [source]

This estimator is much faster than `GradientBoostingClassifier` for big datasets (n_samples >= 10 000).

Stacking

Stacking (abreviatura de stacked generalization). Se basa en una idea sencilla: en lugar de utilizar funciones triviales (como el voto duro) para agregar las predicciones de todos los predictores en un conjunto, **¿por qué no entrenamos un modelo para realizar esta agregación?**

```
from sklearn.ensemble import StackingClassifier

stacking_clf = StackingClassifier(
    estimators=[
        ('lr', LogisticRegression(random_state=42)),
        ('rf', RandomForestClassifier(random_state=42)),
        ('svc', SVC(probability=True, random_state=42))
    ],
    final_estimator=RandomForestClassifier(random_state=43),
    cv=5  # number of cross-validation folds
)
stacking_clf.fit(X_train, y_train)
```



La figura muestra un conjunto de este tipo realizando una tarea de regresión sobre una nueva instancia.

Default of Credit Card Clients

Donated on 1/25/2016

This research aimed at the case of customers' default payments in Taiwan and compares the predictive accuracy of probability of default among six data mining methods.

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate	Business	Classification
Feature Type	**# Instances**	**# Features**
Integer, Real	30000	23

Enlace

The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients

I. Yeh, Che-hui Lien · Published in Expert systems with... 1 March 2009 · Computer Science, Business, Economics

872 Citations

Highly Influential Citations	53
Background Citations	251
Methods Citations	254
Results Citations	12

Artículo

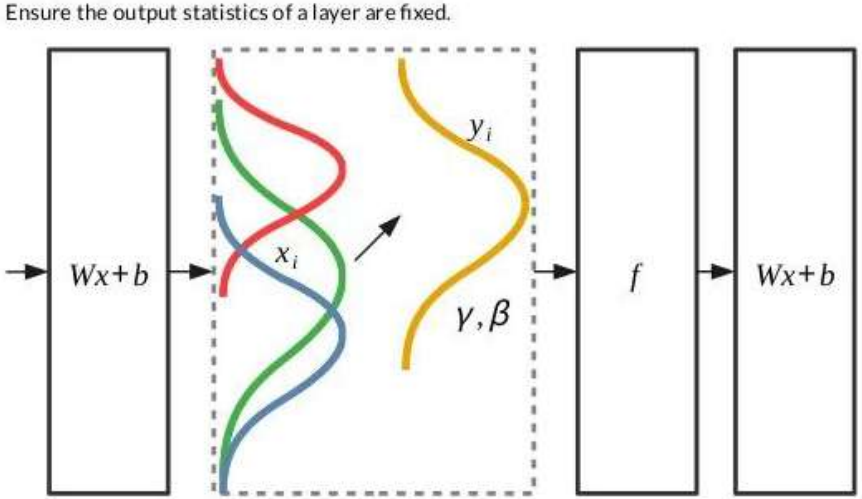
Documento PDF

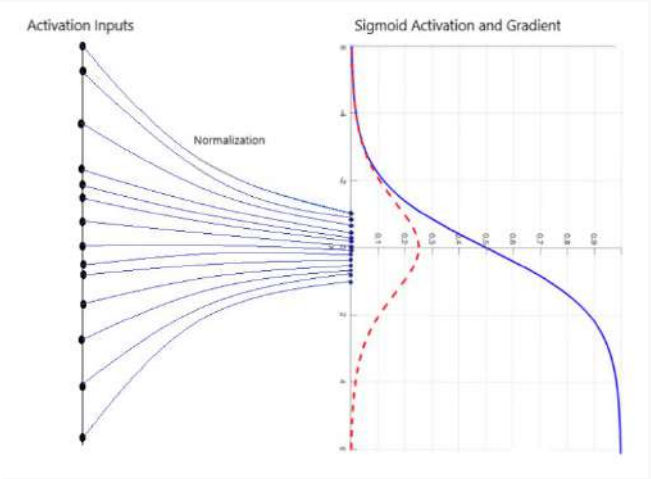
5.2

Tópicos avanzados

Cierre del módulo

Batch normalization



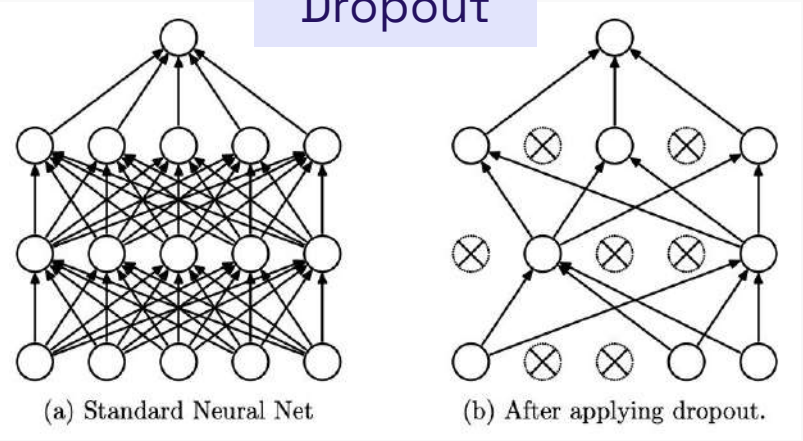


PyTorch

TensorFlow

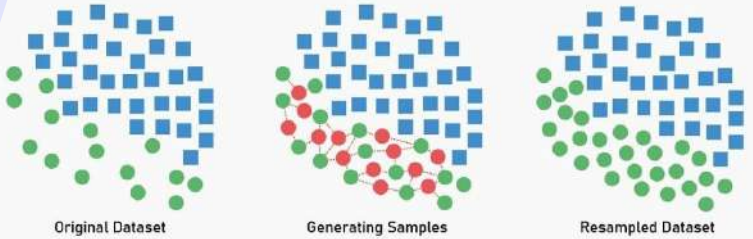
Frameworks más potentes y personalizables

Dropout

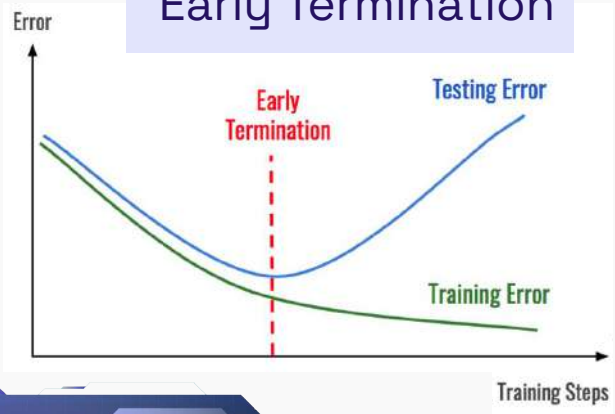


SMOTE

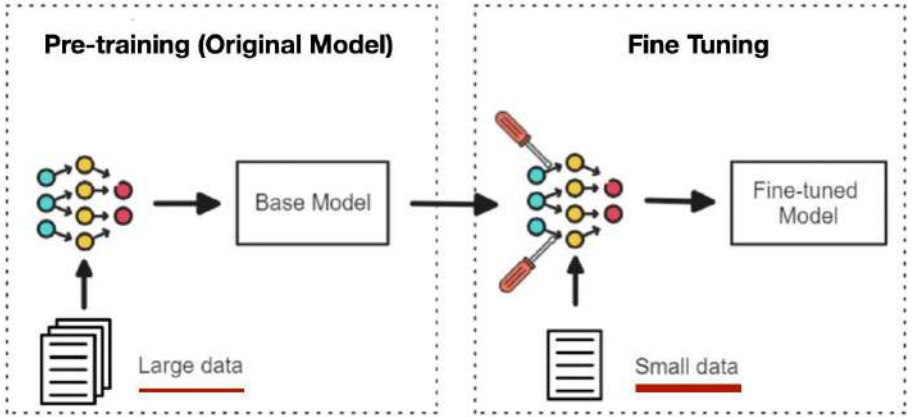
Synthetic Minority Oversampling Technique



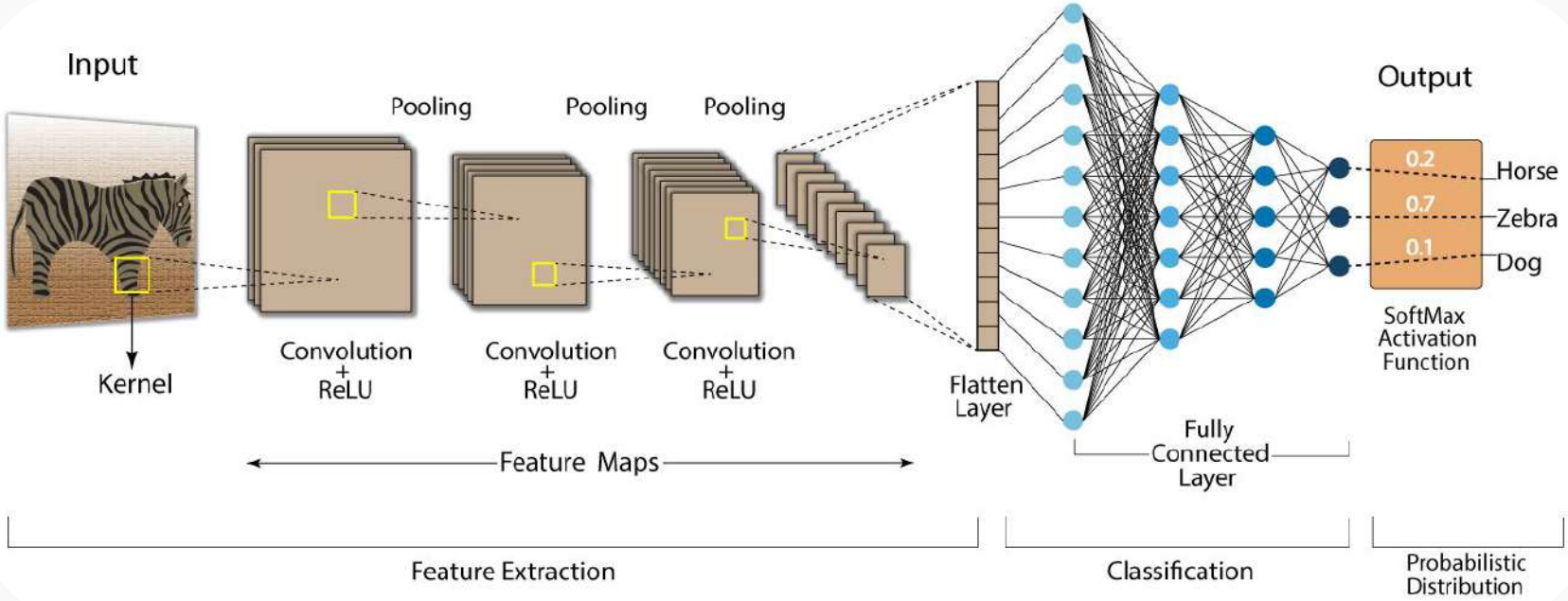
Early Termination



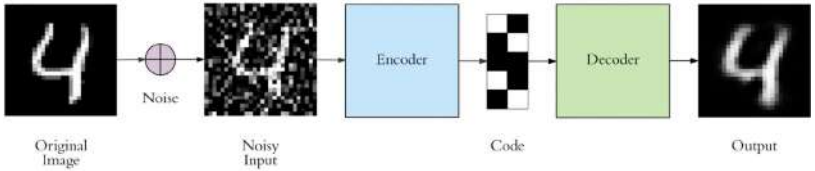
Fine Tuning

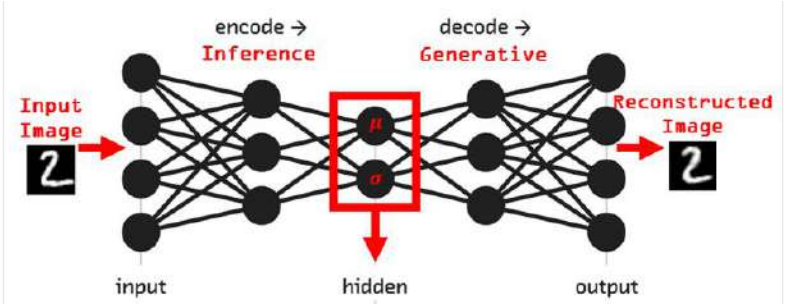


Convolutional Neural Networks (CNNs)

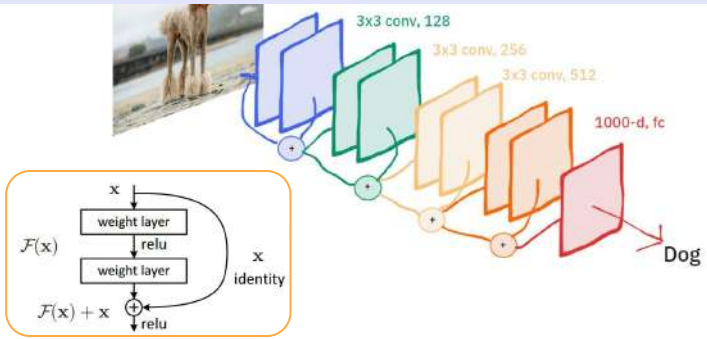


Autoencoders

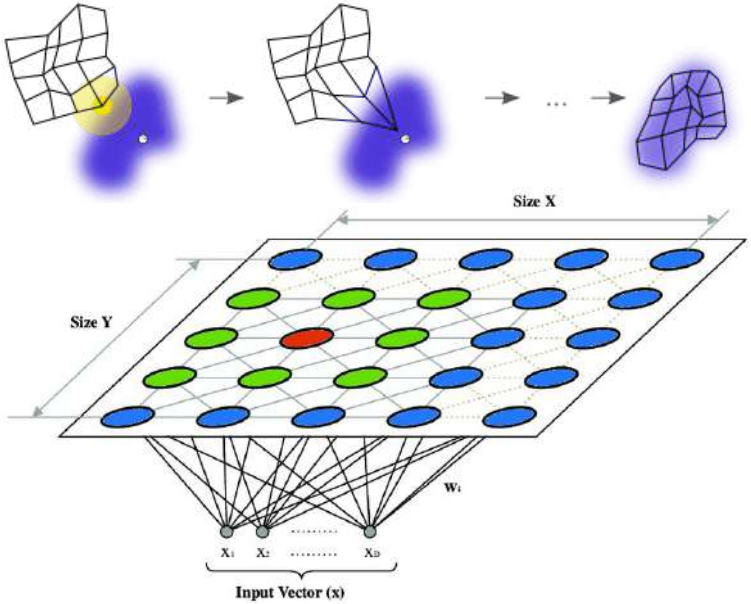




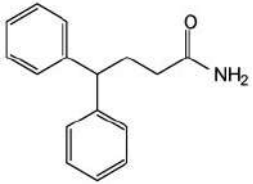
Residual Neural Networks



Self-organizing maps

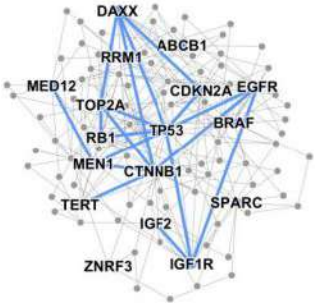


Transporte/energía



Química

Redes/Internet

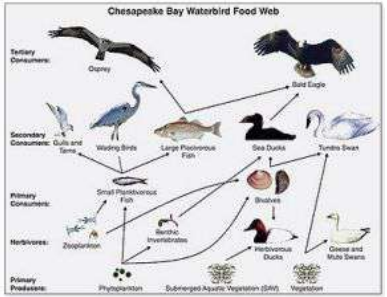


Medicina

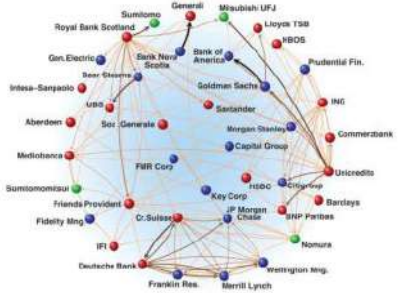
Redes sociales

Estructuras de Datos complejas como grafos (o redes)

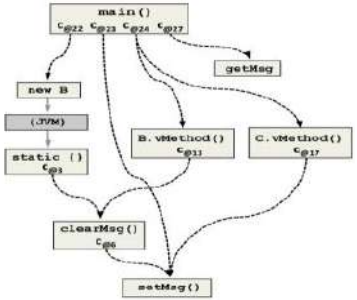
Los grafos son ampliamente utilizados para describir y analizar **entidades** y sus **interacciones** en muchas disciplinas.



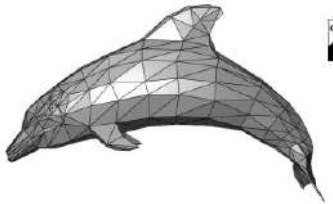
Biología



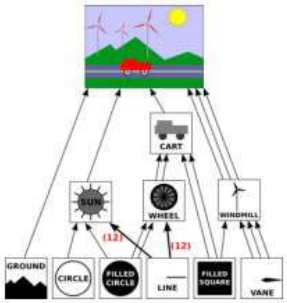
Economía



Software

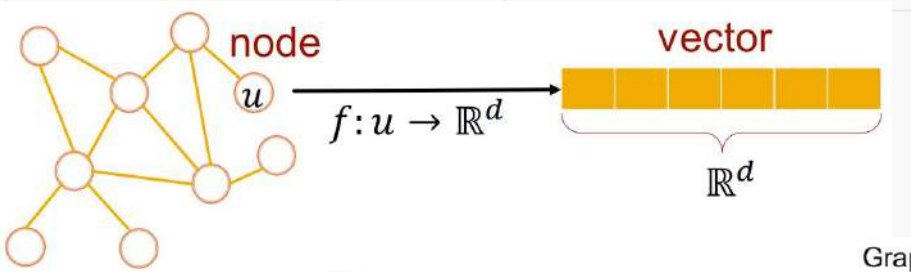


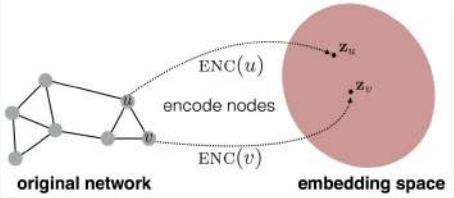
Simulación

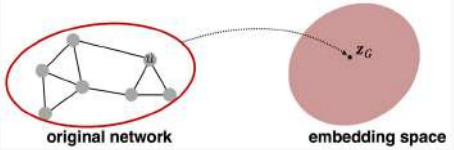


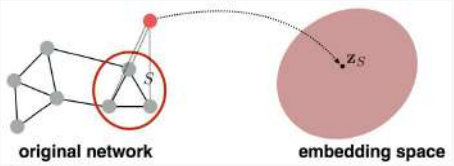
Computer vision

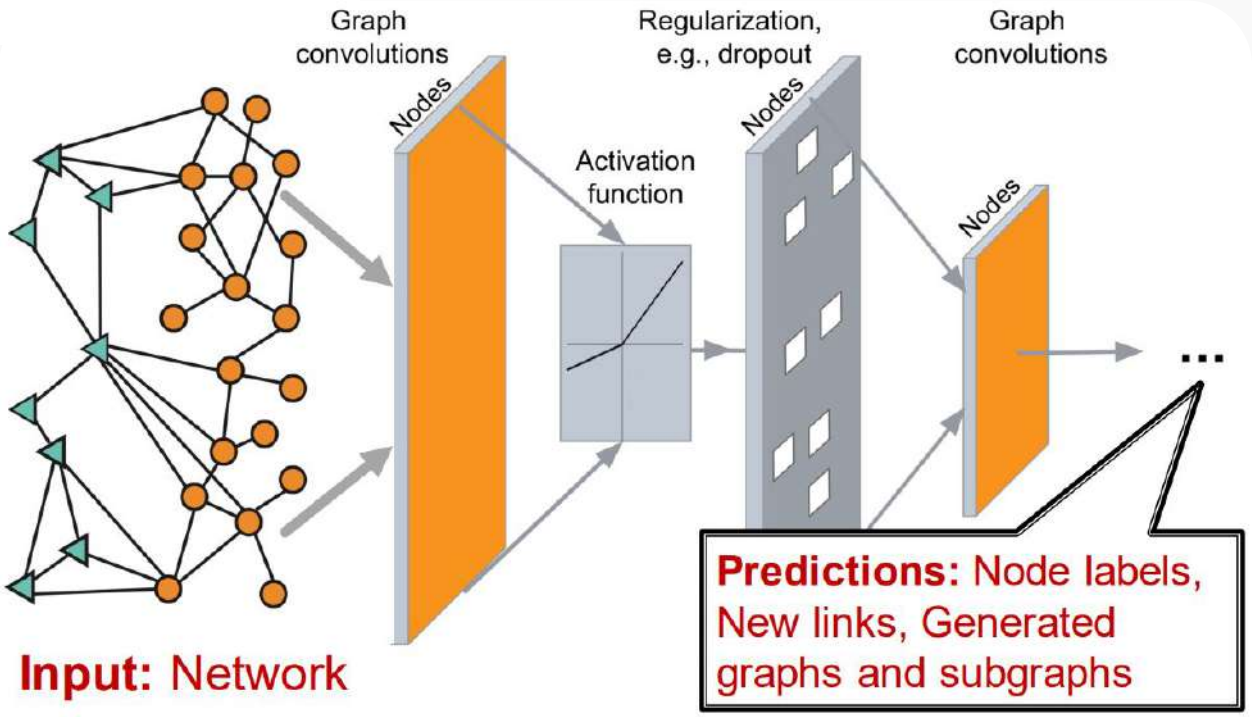
Graph Machine Learning









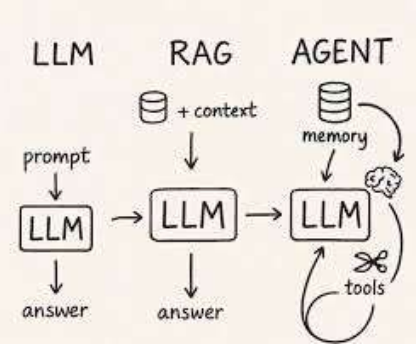


Arquitectura Transformer (aka. LLMs)

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Lukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* † illia.polosukhin@gmail.com

El mundo cambió...



ChatGPT · LLaMA · LangChain

deepseek · Claude · Gemma

Ollama · Gemini

Actividad: aplicar lo aprendido en un problema de interés personal

- **Paso 1: [Dataset]** Encontrar un problema que le llame la atención a usted a nivel personal/profesional. Se sugieren:
  - https://archive.ics.uci.edu/ https://www.kaggle.com/datasets
  - https://www.openml.org/ https://paperswithcode.com/datasets
  - Alternativamente, algún dataset de interés personal.
- **Paso 2: [Problem definition]** Evaluar si el problema es de regresión, clasificación, clusterización, o predicción de series temporales.
  - Explicar cuales son las variables de entrada y salida.
  - ¿Es posible resolver dicho problema de forma eficiente sin recurrir a inteligencia artificial?
- **Paso 3: [Data preparation]** Realizar todas las operaciones necesarias para que los datos estén listos para el modelaje.
  - Eliminar datos nulos, crear dummies o codificar datos categóricos.
  - Tomar en cuenta la existencia de datos atípicos (outliers) y/o clases desbalanceadas (detección de anomalias).
  - (Opcional) Normalizar los datos de entrada StandardScaler.
  - Separar en subsets de entrenamiento y prueba.
- **Paso 4: [Modelado]** Entrenar al menos dos tipos de modelos distintos.
  - Redes neuronales MLPRegressor, Máquinas de soporte vectorial SVM, Árboles de decisión DecisionTree, Bosques aleatorios RandomForest, o Métodos de ensamble AdaBoost y GradientBoosting.
- **Paso 5: [Model tuning]** Probar tantas combinaciones de hyper-parámetros como usted desee (y su equipo se lo permita).
  - Utilizar un enfoque de train-validation-test o validación cruzada en lugar de train-test.
- **Paso 6: [Output]** Seleccionar el mejor modelo en base a alguna métrica vista en el módulo y explicar porqué usted lo considera así.
  - Si su problema es de clasificación, proveer matriz de confusión.
  - Si su problema es de regresión, proveer gráfico de salidas reales vs. predichas.

Se debe subir un archivo ipynb (júpiter notebook) a la plataforma e-campus antes del comienzo del siguiente módulo.

¡Gracias por su atención!

Datos de contacto:

🏠 Fort Collins, Colorado, USA

📧 rcanizalest@outlook.com

📧 rcanizales@uca.edu.sv

📧 rcanizal@colostate.edu

🌐 www.cs.colostate.edu/~rcanizal