

# APRENDIZAJE AUTOMÁTICO CON PYTHON

***PRIMERA SESIÓN***

*TOULOUSE LAUTREC  
EDUCACIÓN CONTINUA*

*ALEXANDER VALDEZ PORTOCARRERO*

**TOULOUSE  
LAUTREC**

# CONTENIDO

## Introducción al Machine Learning

### SESIÓN - 1

#### Introducción al Machine Learning

- Fundamentos de Machine Learning
- Tipos de Aprendizaje
- Librerías de Python para Machine Learning
- Algoritmos de Regresión
- Evaluación de Modelos de Regresión

## Introducción al Machine Learning

### SESIÓN - 2

#### Clasificación

- Algoritmos de Clasificación
- Balanceo de Datos
- Regresión Logística
- Evaluación de Modelos de Clasificación
- Árboles de Decisión

# CONTENIDO

## Introducción al Machine Learning

### SESIÓN - 3

#### Clustering

- Algoritmos de Agrupación
- Clustering No Jerárquico: Kmeans, PAM, CLARA
- Clustering Jerárquico: AGNES, DIANA
- Clustering Basado en Densidad: DBSCAN

## Introducción al Machine Learning

### SESIÓN - 4

#### Reducción de Dimensionalidad

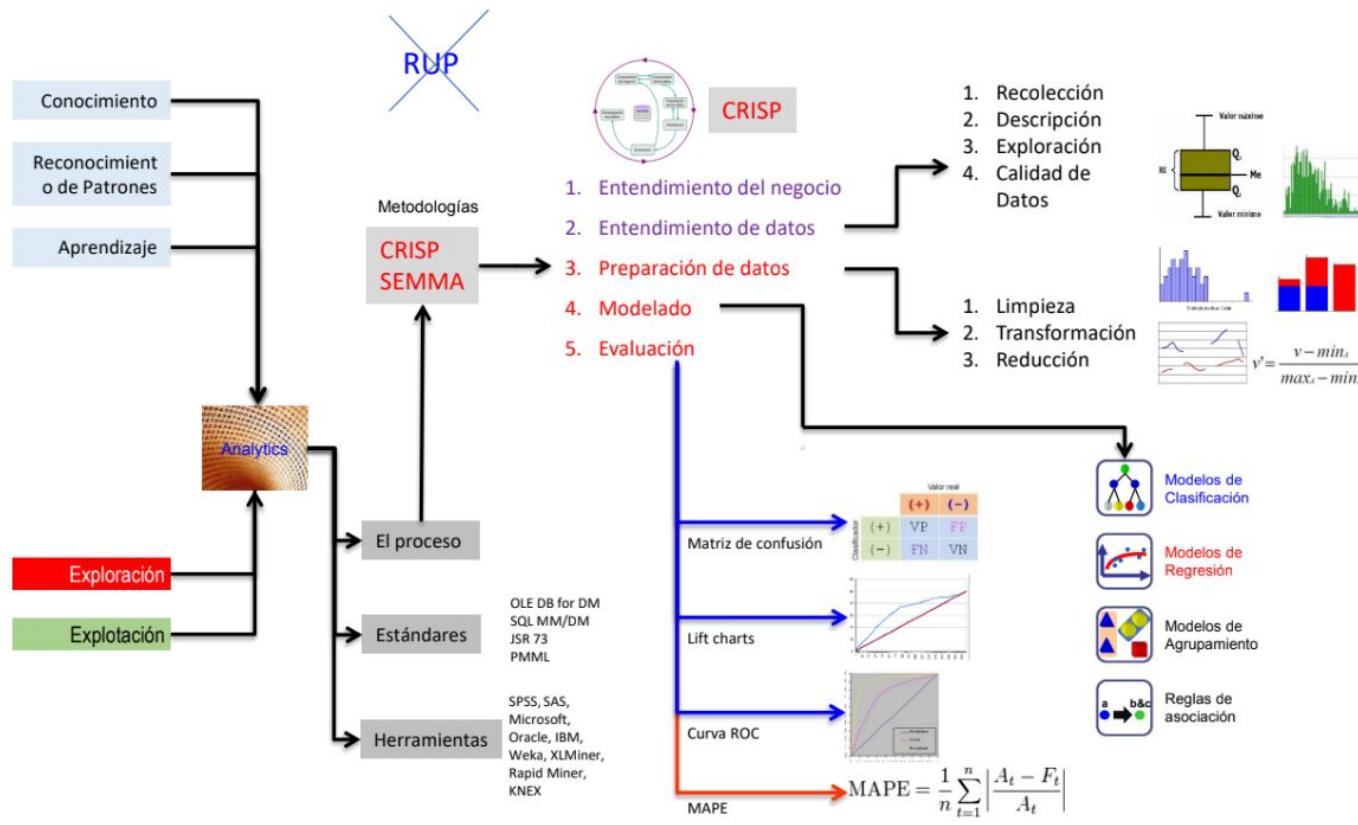
- Análisis de Componentes Principales
- Análisis Factorial
- Selección de Variables

## Normas de clase online:

- Habrá un **break de 10 min** después de la parte teórica y antes de la parte práctica en Google Collaboratory.
- La evaluación se realizará **durante la sesión 4 como trabajo final** y obtendrás puntos adicionales en función de las tareas resueltas y enviadas al correo del profesor:

TAREAS PRESENTADAS	PUNTOS ADICIONALES
Ninguna	0
1	+2
2	+4
3	+6

# Mapa Conceptual del Curso

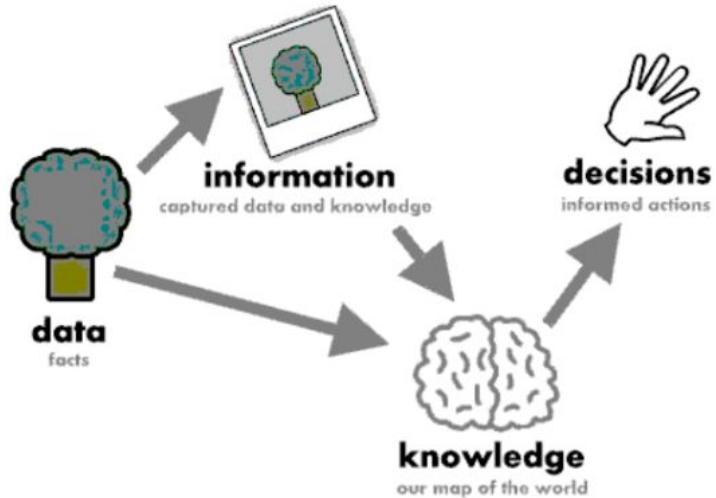


The CRoss Industry Standard Process for Data Mining

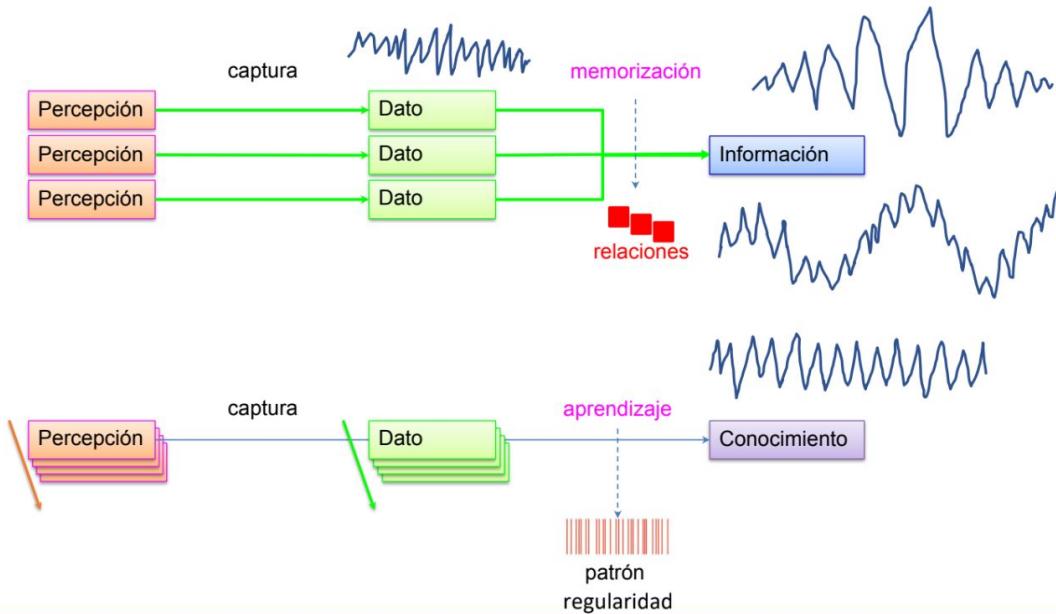
SEMMA es un acrónimo que significa Muestrear, Explorar, Modificar, Modelar y Evaluar.

Rational Unified Process (RUP) es una metodología de desarrollo de software orientado a objeto que establece las bases, plantillas..

# DATOS, INFORMACIÓN Y CONOCIMIENTO



## Información y Conocimiento

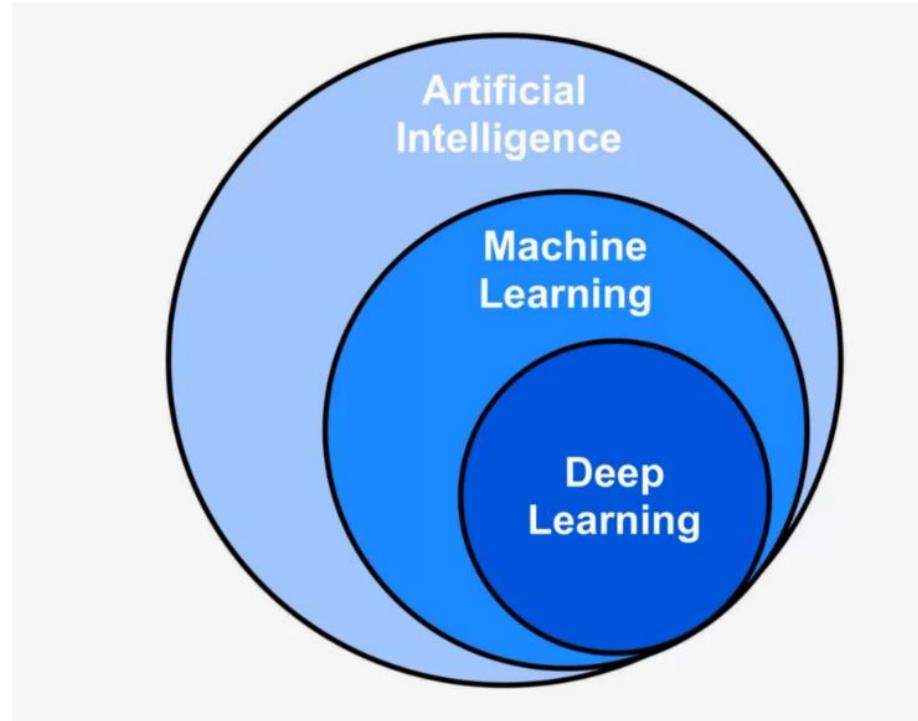


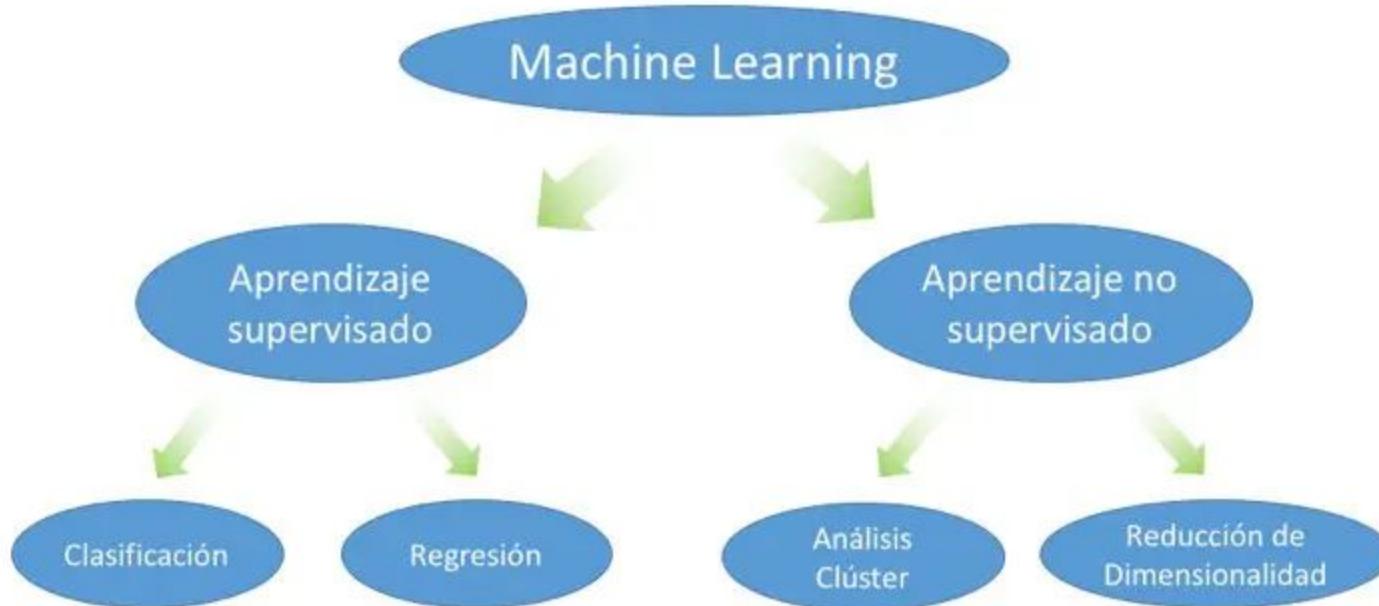
¿Qué es lo que tiene que saber hacer o decidir el responsable de un equipo para acompañarlo en la consecución de sus objetivos? ¿Qué es lo que tiene que saber hacer o decidir un operario especializado para reparar una máquina de la forma más eficientemente posible?

Solemos definir el conocimiento como “la capacidad de las personas para tomar una decisión (o un conjunto de ellas) cuyo impacto permitirá acercarnos o alcanzar los objetivos planteados”. El conocimiento se adquiere con la práctica y a través de la experiencia.

La información es el conjunto de datos que adecuadamente organizados nos dan una idea de lo que está sucediendo a nuestro alrededor. La información es la base para la toma de decisiones.

# Aprendizaje Automático





# Aprendizaje Automático



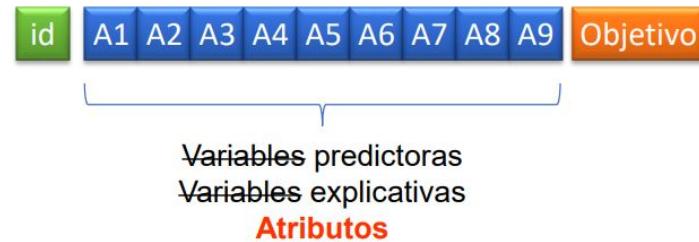
## Tabla de Contenido

- EL DATASET
- EL PROCESO DE APRENDIZAJE
- EJEMPLOS DE MODELOS PREDICTIVOS
- EVALUACION DE MODELOS
- PREGUNTAS

# **EL DATASET**

# DataSet

- Es un instrumento para representar los datos que requieren las herramientas de análisis de datos.
- Es la fuente de datos que usarán los algoritmos de Aprendizaje Automático.
- Es una estructura de datos que contiene tipos de atributos.



$$\text{Objetivo} = f( \text{A1, A2, A3, A4, A5, A6, A7, A8, A9} )$$

**p ( Objetivo = True)**

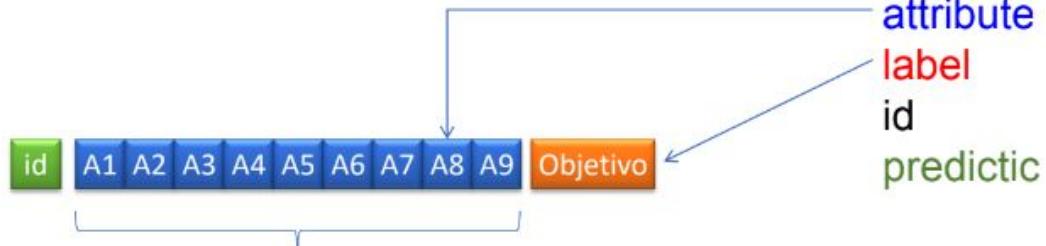
↑  
modelo

# DataSet

Es una estructura de datos, donde c/u tiene un tipo de dato y un rol.

## Tipo de Dato

numeric  
integer  
real  
nominal  
binomial  
polynomic  
date\_time  
date  
time  
text



Variables predictoras  
Variables explicativas  
**Atributos**

Numérico (regresión)  
Categórico (clasificación)

**Rol**  
attribute  
**label**  
id  
predictic

# Ejemplo de DataSet

atributo discreto

Real

Predecida

atributos

instancias o individuos

atributo continuo

Ejemplo	Sitio de acceso A <sub>1</sub>	1 <sup>a</sup> cantidad gastada A <sub>2</sub>	Vivienda (zona) A <sub>3</sub>	Última compra A <sub>4</sub>	Clase	
1	1	0	2	Libro	Bueno	
2	1	0	1	Disco	Malo	
3	1	2	0	Libro	Bueno	
4	0	2	1	Libro	Bueno	
5	1	1	1	Libro	Malo	
6	2	2	1	Libro	Malo	

# EJERCICIO 1

Leer el dataset entregado,  
Identificar el ID, las  
características y la clase.

multiclass

Estado 1	1	True
Estado 2	2	False
Estado 3	3	

Multi-state

Estado 1
Estado 2
Estado 3

```
import pandas as pd
import numpy as np

# define nombre de archivo
datafile = '.\\data\\E001.xlsx'

# lee los datos, con id = paciente
dataset_ori = pd.read_excel(datafile, index_col=0)
# lee dataset
print('dataset_ori')
print(dataset_ori)
print(dataset_ori.shape)
dataset = dataset_ori.to_numpy() # numpy
print('dataset')
print(dataset)
print(dataset.shape)

# separa los datos en X y target
X = dataset[:, :-1]
y = dataset[:, -1:]

print('X')
print(X)
print('y')
print(y)
```

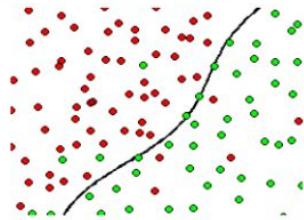
# DataSet

## Modelos Supervisados

clasificación

id	A1	A2	A3	A4	A5	A6	A7	A8	A9	Objetivo
										Categórico

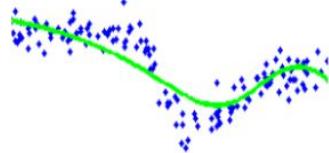
P



regresión

id	A1	A2	A3	A4	A5	A6	A7	A8	A9	Objetivo
										Numérico

P

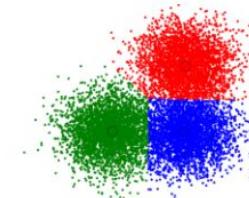


## Modelos No Supervisados

agrupamiento

id	A1	A2	A3	A4	A5	A6	A7	A8	A9

D



asociación

id	E1	E2	E3	E4	E5	E6	E7	E8	E9

D



## Ejemplo de DataSet

- Para los siguientes datasets, diga si se aplica un modelo de clasificación o de regresión.

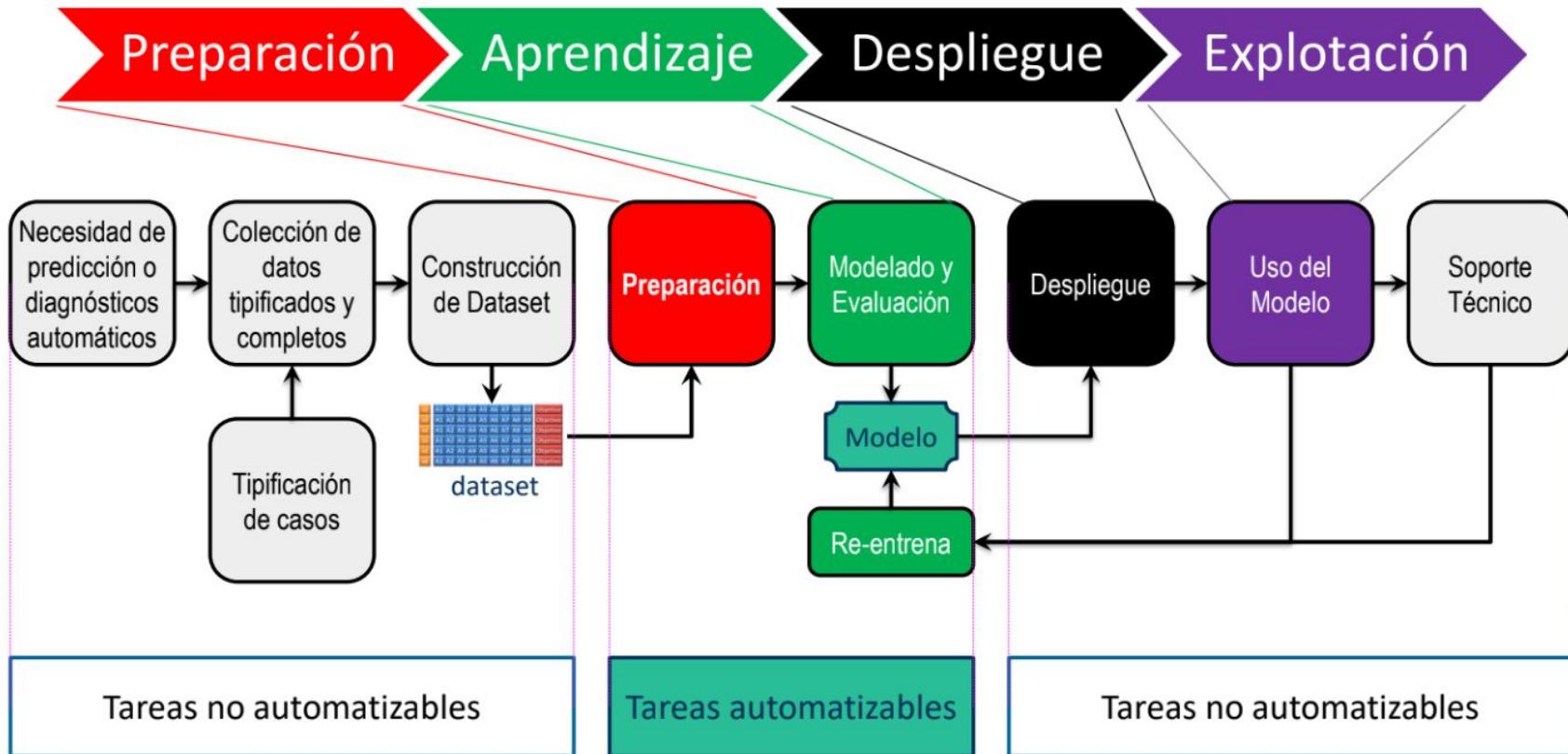
V1	V2	V3	Clase
A	P	1	SI
B	Q	2	SI
A	Q	2	NO
A	P	5	SI
B	P	4	NO
B	P	3	NO
A	Q	2	SI
A	P	4	NO
B	Q	3	SI
A	P	1	SI

C1	C2	C3	Clase
A	XW	P	1
B	WX	Q	1
A	RW	Q	1
A	XW	P	0
B	PE	P	0
B	WX	P	1
A	XW	Q	1
A	XW	P	0
B	RE	Q	1
A	RW	P	1

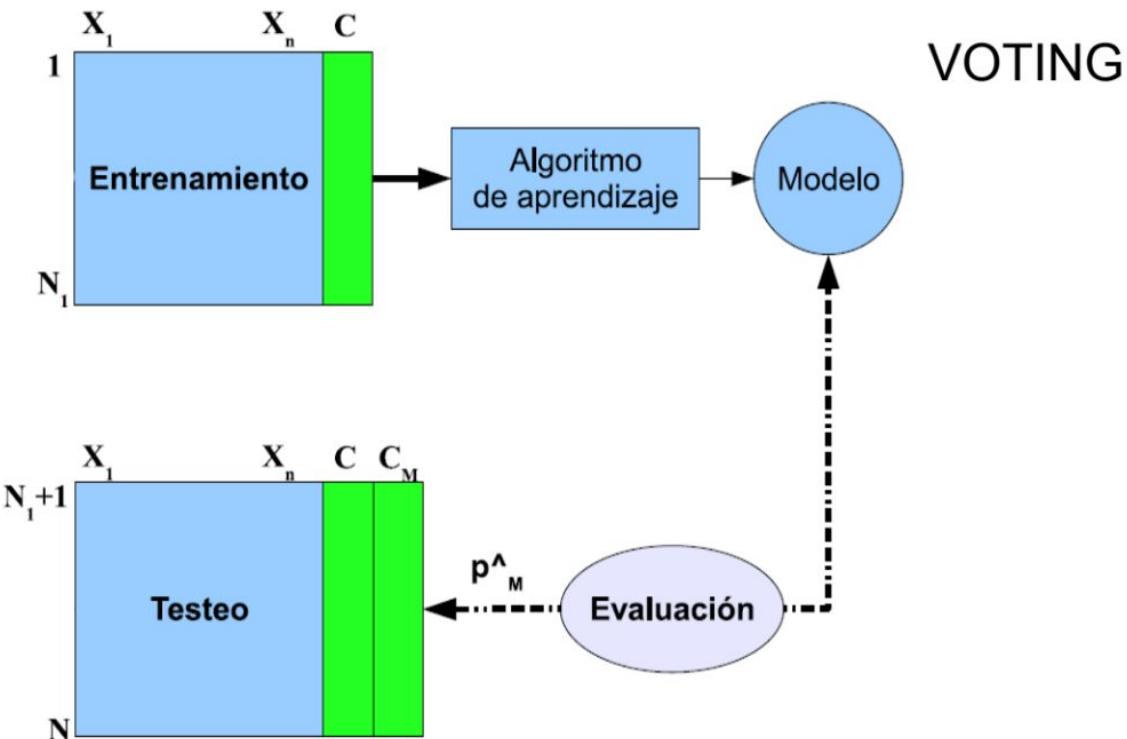
C1	C2	C3	Clase
FA	W	PA	1.2
FB	X	QB	0.3
FA	R	QB	0.1
RA	X	PB	0.9
RB	P	PA	2.3
FB	X	PB	0.0
RA	W	QA	0.7
FA	W	PA	0.7
RB	E	QA	1.3
FA	R	PB	1.9

# **EL PROCESO DE APRENDIZAJE**

# El proceso del aprendizaje automático



# Partición del Dataset



## Ejercicio 2

Leer el dataset entregado,  
Identificar el ID, las  
características y la clase.  
Particione la data para  
entrenamiento y para  
pruebas

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OrdinalEncoder
import numpy as np

datafile = '.\\data\\E001.xlsx' # define nombre de archivo

# lee los datos, con id = paciente
dataset_ori = pd.read_excel(datafile) # lee dataset, sin index
print('dataset_ori', dataset_ori)
print(dataset_ori.shape)
dataset = dataset_ori.to_numpy() # numpy
print('dataset', print(dataset))
print(dataset.shape)

X = dataset[:, :-1] # separa los datos en X y target
y = dataset[:, -1:]
print('X', X, print('y', y))

#Split data
train_X, test_X, train_y, test_y = train_test_split(X, y, train_size=0.75)
print(train_X, test_X, train_y, test_y)

train = np.concatenate((train_X, train_y), axis=1) # concatena
print(train)
test = np.concatenate((test_X, test_y), axis=1)
print(test)
```

# Ejercicio 3

## Leer datos para clasificación

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OrdinalEncoder
import numpy as np

# define nombre de archivo
datafile = '.\\data\\E002.xlsx'

# lee los datos, con id = paciente, con index
dataset_ori = pd.read_excel(datafile, index_col=0)
print(dataset_ori.shape)
print(dataset_ori)
dataset_ori = dataset_ori.to_numpy() # numpy
print(dataset_ori)

enc = OrdinalEncoder()
enc.fit(dataset_ori)

# codifica resultados
dataset = enc.transform(dataset_ori)

print(dataset)
print(dataset.shape)
```

```
# separa los datos en X y target
X = dataset[:, :-1]
y = dataset[:, -1:]

print(X)
print(y)

# Split the data into a training and a testing set
train_X, test_X, train_y, test_y = train_test_split(X,
y, train_size=0.75)

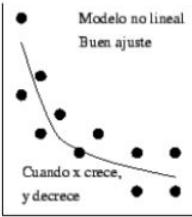
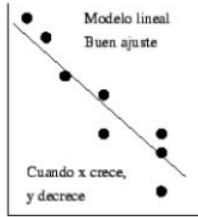
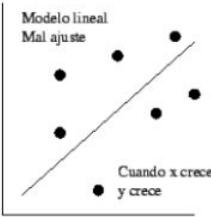
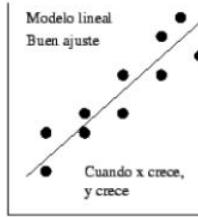
print(train_X)
print(test_X)
print(train_y)
print(test_y)

# decodifica resultados
train = np.concatenate((train_X, train_y), axis=1)
print(train)
train = enc.inverse_transform(train)
print(train)
```

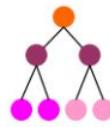
# EJEMPLOS DE MODELOS PREDICTIVOS

# Modelo de Regresión

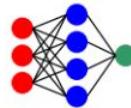
- Intenta determinar la función que mapea un conjunto de variables de entrada  $X$  (*independiente*), en una (o más) variables de salida  $Y$  (*dependiente*), .
- Es básicamente numérica.
- Está basada en supuestos estadísticos.



- Árboles de decisión.



- Redes Neuronales



# EJERCICIO 4. REGRESIÓN LINEAL

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import numpy as np
import pandas as pd
import sys

np.set_printoptions(threshold=sys.maxsize)

# define nombre de archivo
datafile = '.\\data\\E004.xlsx'

# lee datos
dataset_ori = pd.read_excel(datafile, index_col=0) # lee dataset

# describe dataset
dataset_ori.describe()
dataset = dataset_ori.to_numpy() # numpy
print(dataset)

# separa los datos en X y target
X = dataset[:, :-1]
y = dataset[:, -1:]
print('X', X)
print('y', y)

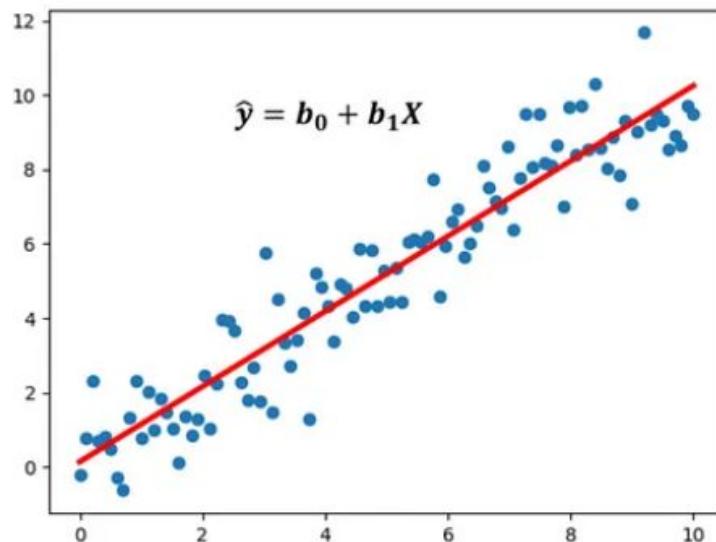
# separa los datos para entrenamiento y para pruebas
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)

# entrena regresion lineal
regressor = LinearRegression()
rl = regressor.fit(X_train, y_train)
print(regressor)

print('inter:', regressor.intercept_, 'coef:', regressor.coef_)

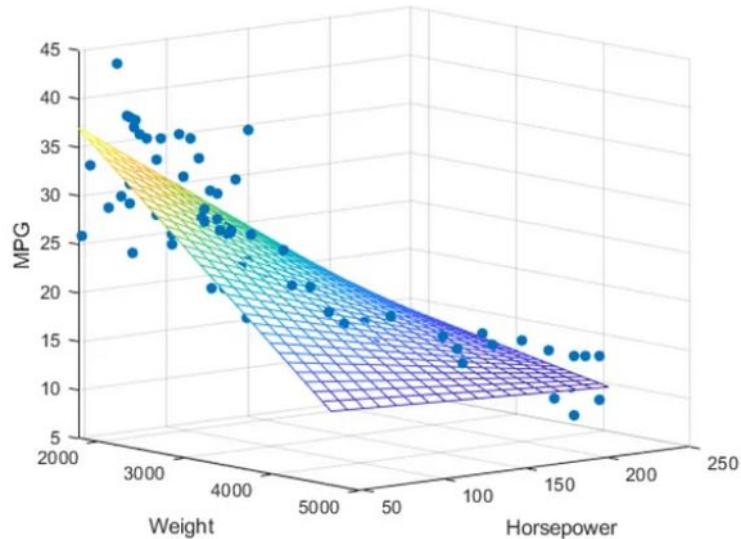
# haciendo predicciones
y_test_pred = regressor.predict(X_test)
print('X_test', X_test)
print('y_test_pred', y_test_pred)
```

## Regresión Lineal



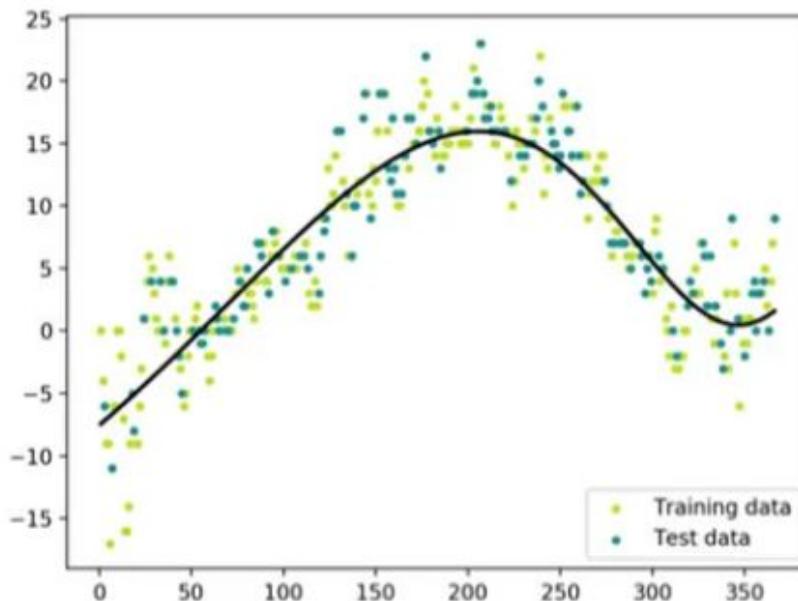
[source](#)

## Regresión Lineal Múltiple



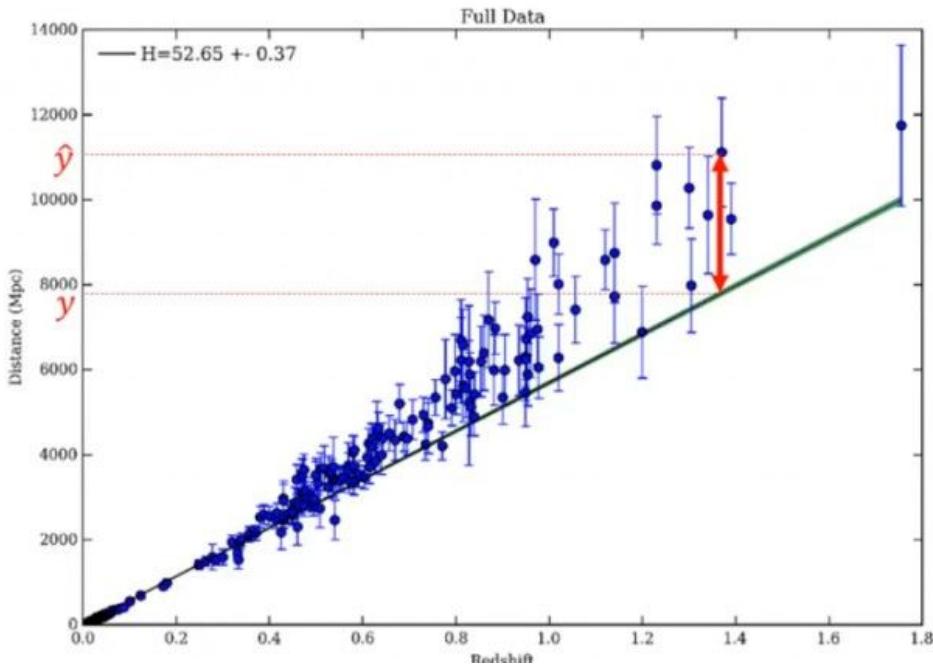
$$Y_i = \beta_0 X_{i0} + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_k X_{ik} + \varepsilon_i$$

## Regresión Polinomial



$$y = a + bx + cx^2 + dx^3 \dots$$

# EVALUACIÓN DE MODELOS DE REGRESIÓN



$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j|$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}$$

$$RAE = \frac{\sum_{j=1}^n |y_j - \hat{y}_j|}{\sum_{j=1}^n |y_j - \bar{y}|}$$

$$RSE = \frac{\sum_{j=1}^n (y_j - \hat{y}_j)^2}{\sum_{j=1}^n (y_j - \bar{y})^2}$$

# MEDICIÓN DE RESULTADOS

Medidas de la bondad de ajuste (intentan medir en cuánto se desvía el nivel de la predicción del nivel real).

Nombre	Unidad medida	Fórmula
<b>MAPE</b> Mean Average Percentage Error	%	$100\% * \frac{\sum_{t=1}^{t=N} \ E_t/Y_t\ }{N}$
<b>MAE</b> Mean Absolute Error	de la serie	$\frac{\sum_{t=1}^{t=N} \ E_t\ }{N}$
<b>RMSE</b> Root Mean Square Error	de la serie	$\sqrt{\frac{\sum_{t=1}^{t=N} E_t^2}{N}}$

Siendo:

$Y_t$  Es el valor de la variable independiente

$F_t$  Es el valor predecido

Es el error de la predicción  $E_t = Y_t - F_t$

$N$  Número de puntos predecidos.

$$MAPE = \frac{\sum_{i=1}^n 100 |Real_i - Pronóstico_i|}{Real_i n}$$

## EJERCICIO 5

Nro	Y	$Y_{predic}$
1	28.4	27.8
2	21.4	21.9
3	21.4	21.2
4	19.2	19.2
5	26.5	26.1
6	26.2	26.1
7	17.0	17.2
8	26.0	26.1
9	29.0	29.5
10	10.0	9.9
11	16.5	16.6
12	26.0	25.6
13	22.9	22.8
14	18.2	17.9
15	30.0	30.4
16	26.2	26.1
17	33.0	33.0
18	16.1	16.5
19	20.7	20.5
20	11.5	11.6

```
import pandas as pd
import numpy as np
import seaborn as sns

def percentage_error(actual, predicted):
    res = np.empty(actual.shape)
    for j in range(actual.shape[0]):
        if actual[j] != 0:
            res[j] = (actual[j] - predicted[j]) / actual[j]
        else:
            res[j] = predicted[j] / np.mean(actual)
    return res

def mean_absolute_percentage_error(y_true, y_pred):
    return np.mean(np.abs(percentage_error(np.asarray(y_true),
                                           np.asarray(y_pred))))

# define nombre de archivo
datafile = '.\\data\\E011.xlsx'

# lee los datos, con id = paciente
dataset = pd.read_excel(datafile) # lee dataset, con index
dataset = dataset.to_numpy() # numpy

# separa los datos en X y target
y = dataset[:, -2:]
print(y)

y_real = y[:,0] # real
y_pred = y[:,1] # predecido

# matriz de confusion
mape = mean_absolute_percentage_error(y_real, y_pred)
print(mape)
```

# APRENDIZAJE AUTOMÁTICO CON PYTHON

Gracias por su atencion.