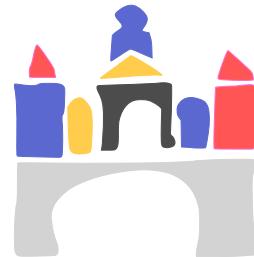


# Introducción a la Minería de Datos

Alicia Olivares Gil  
Dr. David García García  
Dr. Mario Juez Gil  
Dr. José Francisco Díez Pastor  
Dr. Álvar Arnaiz González



Área de Lenguajes y Sistemas  
Informáticos del departamento de  
Ingeniería Informática de la Universidad  
de Burgos



# Materiales

Todo el material de este taller se encuentra en el siguiente enlace:

<https://bit.ly/mineriabie>

- Descargar el fichero “data.zip” y descomprimirlo en el ordenador.
- Descargar el fichero “weka.jar”.
- Tenéis también esta presentación

# ¿Quiénes somos?

- Alicia Olivares Gil.
- David García García



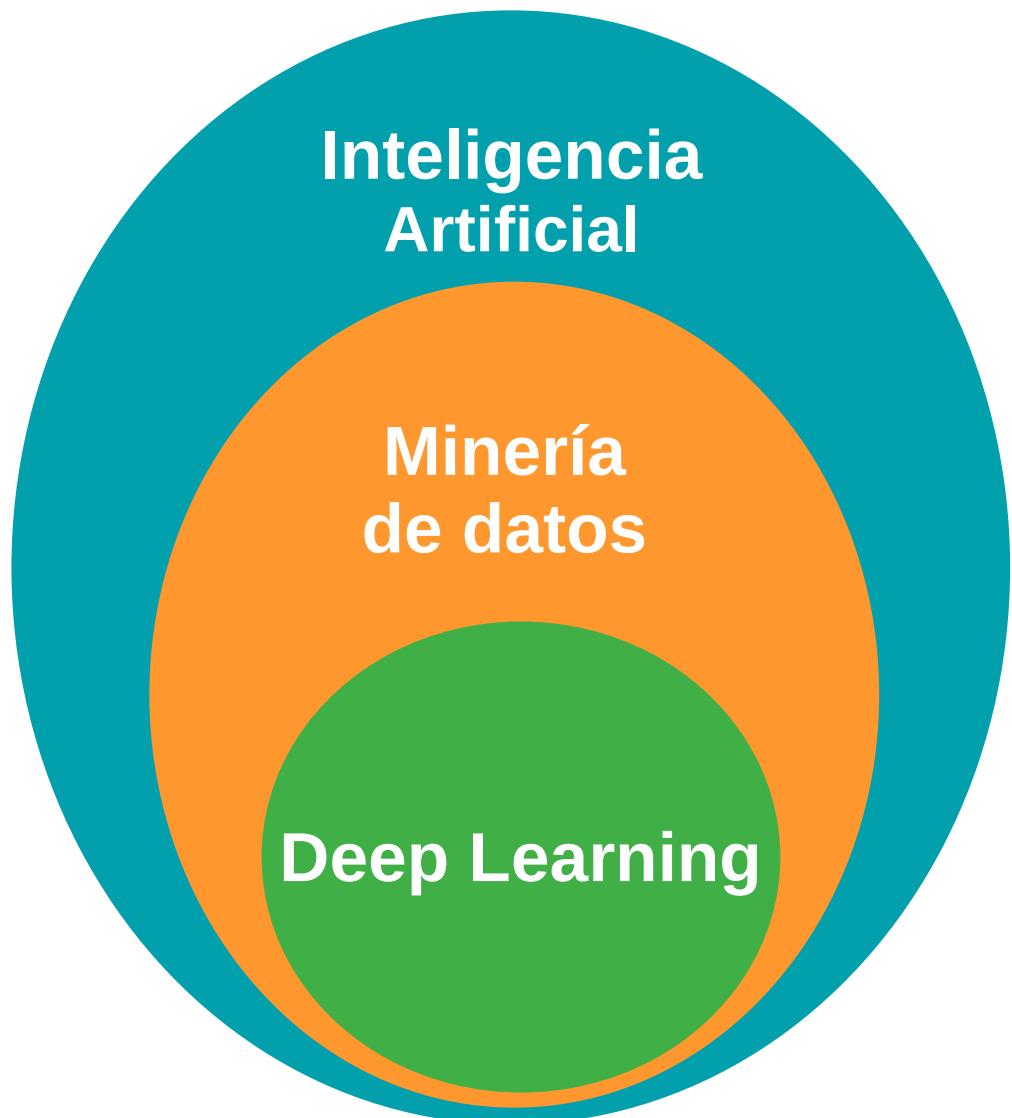
Miembros del grupo de investigación  
**Admirable**.



Profesores del Área de Lenguajes y  
Sistemas Informáticos del departamento  
de Ingeniería Informática de la Universidad  
de Burgos.

# ¿Qué es la Minería de datos?

- Creación de **sistemas informáticos con un comportamiento inteligente**.
- Creación de **sistemas que aprenden por sí mismos** o que extraen conocimiento de los datos.
- Usa redes neuronales profundas con grandes conjuntos de datos.



# ¿Quiénes usan Minería de Datos?



# ¿Quiénes usan Minería de Datos?



## Frequently Bought Together



Total price: \$250.95

Add both to Cart

Add both to List

i These items are shipped from and sold by different sellers. Show details

This item: Canon 0111C001 PowerShot SX610 HS, Wi-Fi Enabled - Black \$229.00

Essential Accessories Bundle Kit For Canon PowerShot ELPH 500 HS, SX600 HS, SX700 HS, SX610 HS... \$21.95

## Customers Who Bought This Item Also Bought



Essential Accessories  
Bundle Kit For Canon  
PowerShot ELPH 500 HS,  
SX600 HS, SX700 HS,...  
  
\$21.95



STK Canon NB-6L NB-6LH  
Battery 1600mAh for  
Powershot SX710 HS,  
.SX600 HS, SX530 HS...  
  
\$11.99



NB-6L Deluxe Accessory  
Bundle for Canon  
PowerShot SX610, SX710,  
D30, and S120 along...  
  
\$39.99



Evecase Digital Camera  
Nylon Pouch Carrying  
Protector Case with Strap  
Black / Red for Canon...  
  
\$9.99



SanDisk 32GB Ultra Class  
10 SDHC UHS-I Memory  
Card Up to 80MB,  
Grey/Black (SDSDUNC...  
  
\$13.95



Transcend 32 GB Class 10  
SDHC Flash Memory Card  
(TS32GSDHC10E)  
  
\$11.99

The screenshot shows the Netflix homepage with the following content:

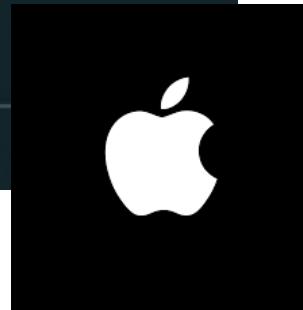
- Top 10 Popular Series:** A list titled "Las 10 series más populares hoy en este país: España" featuring:
  - 1. THE UMBRELLA ACADEMY
  - 2. INTIMIDAD
  - 3. STRANGER THINGS
  - 4. PEAKY BLINDERS
  - 5. LA CASA DE PAPEL: COREA
- Critic Acclaimed Movies:** A section titled "Películas aclamadas por la crítica" featuring:
  - EL PODER DEL PERRO
  - WHIPLASH
  - INFLITRADO EN EL KKKLAN
  - DJANGO DESPIDO DE HABLA
- TV Dramas:** A section titled "Dramas TV de suspense" featuring:
  - EL ABOGADO
  - MR. ROBOT
  - DARK
  - THE WALKING DEAD
  - THE SINNER
  - NAR

## Sistemas de recomendación

# ¿Quiénes usan Minería de Datos?



Reconocimiento de voz y búsqueda inteligente



Reconocimiento de actividad

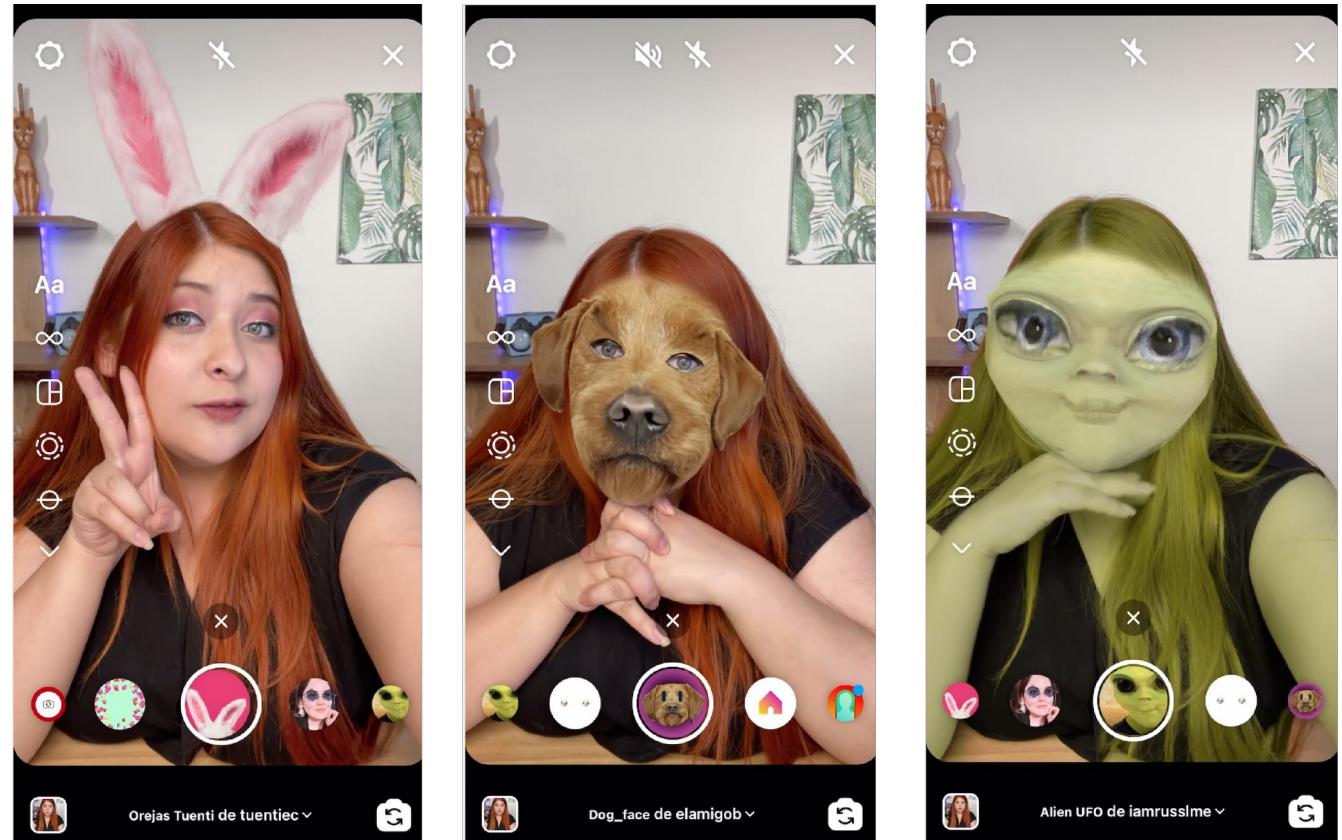


# ¿Quiénes usan Minería de Datos?

## Reconocimiento facial



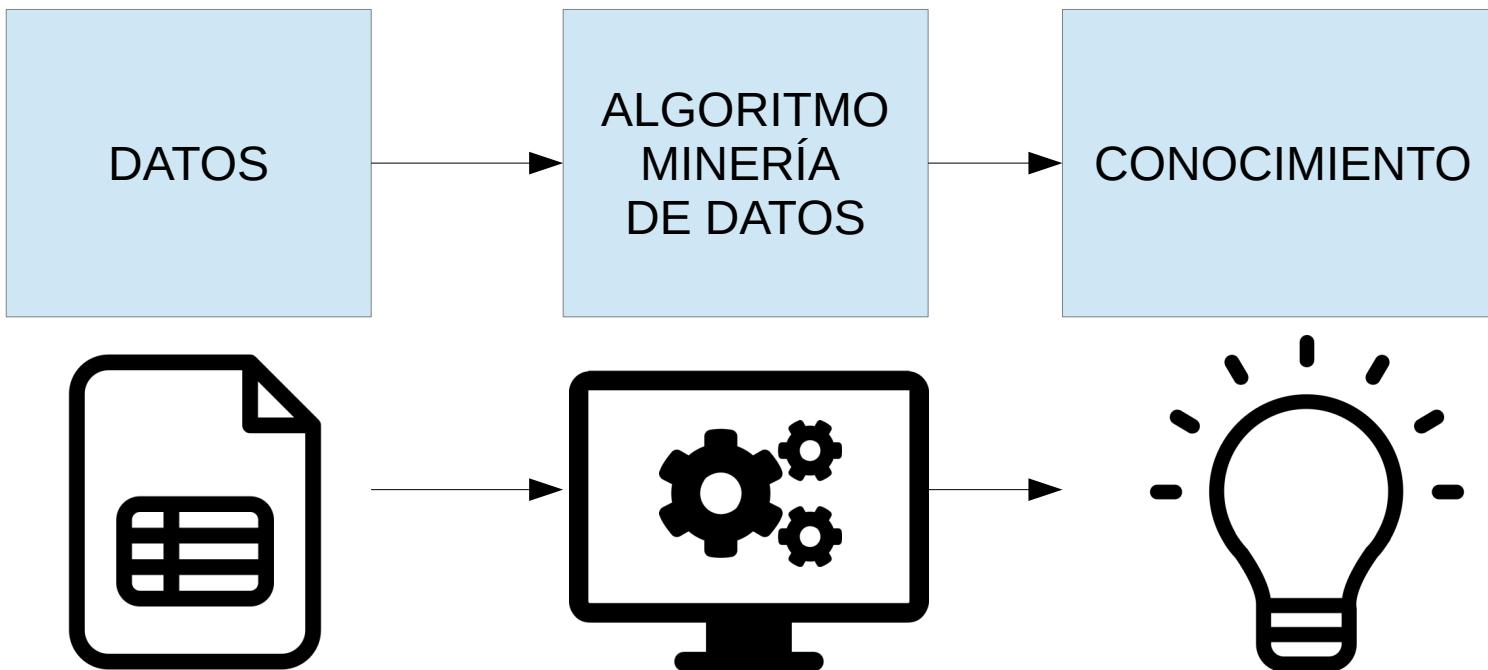
Instagram



# ¿Quiénes usan Minería de Datos?

- Banca (predicción de fraude, predicción de riesgos...)
- Industria manufacturera
- Medicina
- Medioambiente
- ...

# Esquema general



# Datos

Outlook	Temperature	Humidity	Windy	Class
sunny	65	85	false	Don't play
sunny	80	90	true	Don't play
overcast	83	78	false	Play
rain	70	96	false	Play
rain	68	80	false	Play
rain	65	70	true	Play

@RELATION golf

@ATTRIBUTE outlook {sunny, overcast, rain}  
@ATTRIBUTE temperature\_Fahrenheit integer  
@ATTRIBUTE humidity integer  
@ATTRIBUTE windy {false, true}  
@ATTRIBUTE class {dont\_play, play}

@DATA  
sunny, 65, 85, false, dont\_play  
sunny, 80, 90, true, dont\_play  
overcast, 83, 78, false, play  
rain, 70, 96, false, play  
rain, 68, 80, false, play  
rain, 65, 70, true, play

**1.** Instancias o ejemplos = 6

**2.** Atributos = 4:

- Uno nominal (categórico)
- Dos numéricos (real)
- Uno binario (dos posibles valores).

**3.** Una clase binaria (por lo tanto, es un problema de clasificación).

# Tipos de tareas

- **Aprendizaje supervisado** (conocemos la clase)
  - Regresión
  - Clasificación
- **Aprendizaje no supervisado** (no conocemos la clase)
  - *Clustering* (agrupamiento)
  - Reglas de asociación
  - Detección de anomalías
- **Otros:** semisupervisado, aprendizaje con refuerzo, summarización, visualización...

# Tipos de tareas

- De cada tarea vamos a ver:
  - Definición.
  - Ejemplos.
  - Ejercicio en Weka.
  - Casos prácticos.

# Aprendizaje supervisado

# Tipos de tareas

- De cada tarea vamos a ver:
  - Definición.
  - Ejemplos.
  - Ejercicio en Weka.
  - Casos prácticos.

# Regresión

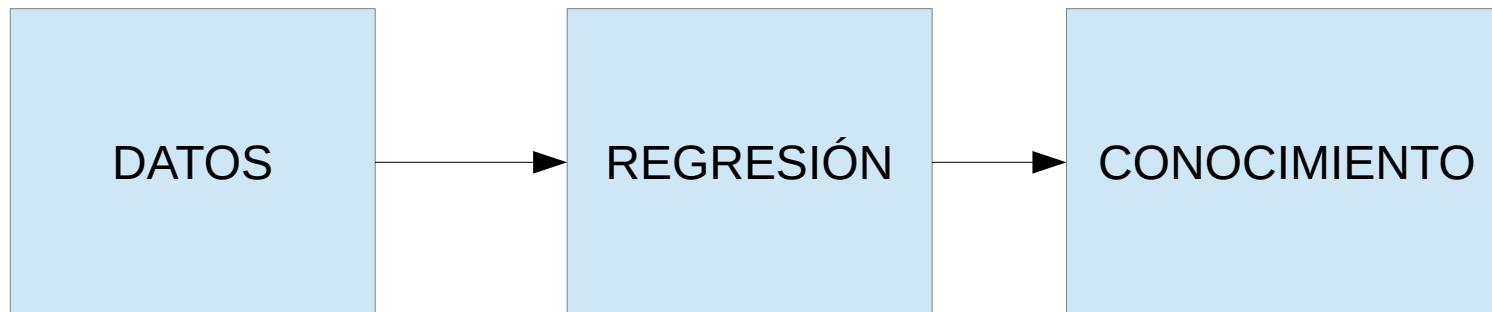
- Tenemos datos: un conjunto de ejemplos.
- Cada ejemplo se compone de:
  - variables independientes o atributos ( $a_1, a_2, \dots, a_n$ )
  - variable dependiente **de tipo numérico** ( $y$ ).
- Se quiere hallar la función que relacione los atributos de entrada con la variable dependiente (con el menor error posible).

$$F(a_1, a_2, \dots, a_n) = y$$

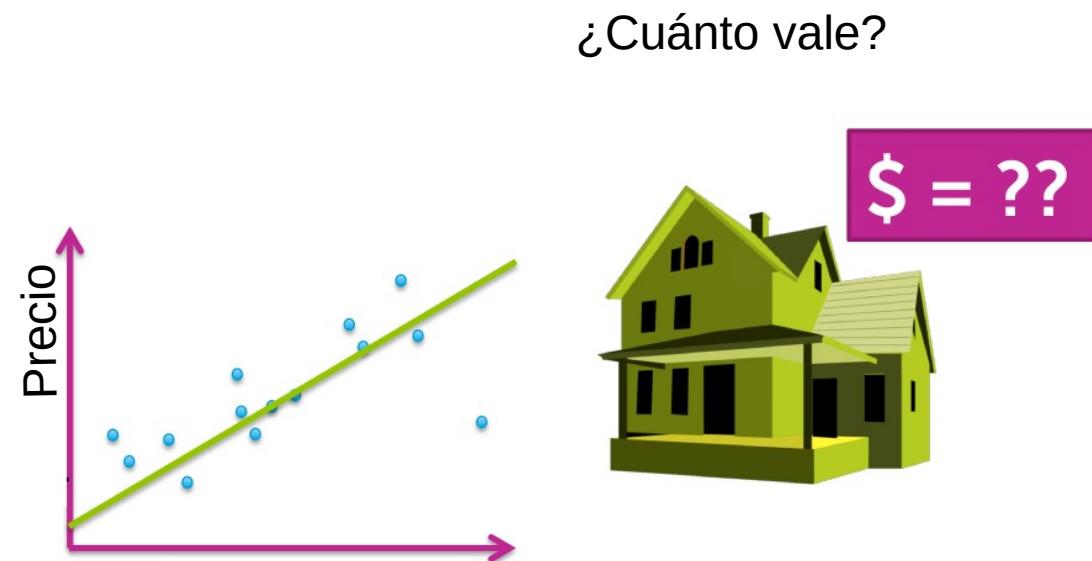
# Regresión

- **Predicción del salario:**
  - **Atributos:** formación, edad, experiencia, ciudad etc.
  - **Valor a predecir:** salario mensual en euros.
- **Predicción de bolsa:**
  - **Atributos:** histórico de valores anteriores, noticias sobre la empresa, valores de empresas similares.
  - **Valor a predecir:** valor futuro de la empresa.
- **Predecir la edad a partir del histórico del navegador de Internet.**

# Regresión



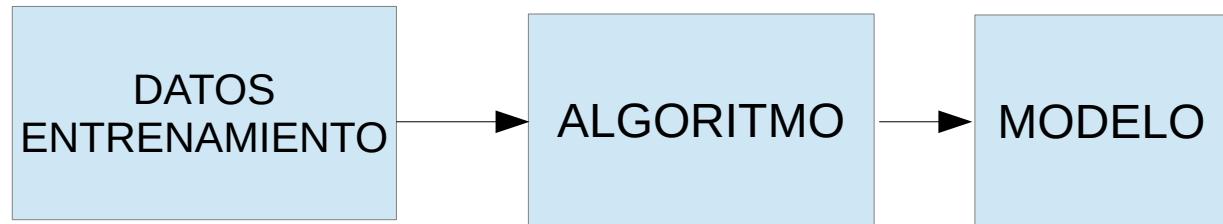
Características  
de las casas



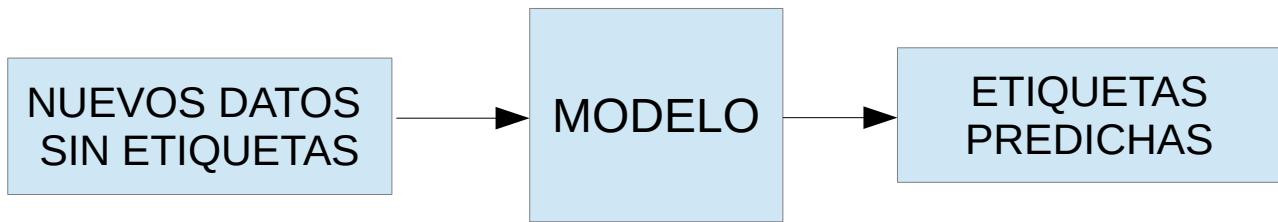
¿Cuánto vale?

# Esquema general

1er paso:  
**Entrenamiento**



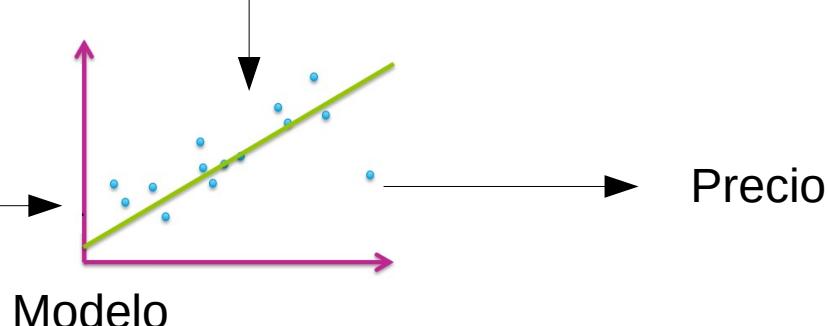
2do paso:  
**Predicción**



Características  
de las casas



¿Cuanto vale?



# Regresión

- Vamos a predecir el precio de coches.
- Vamos a utilizar algoritmos de Weka.

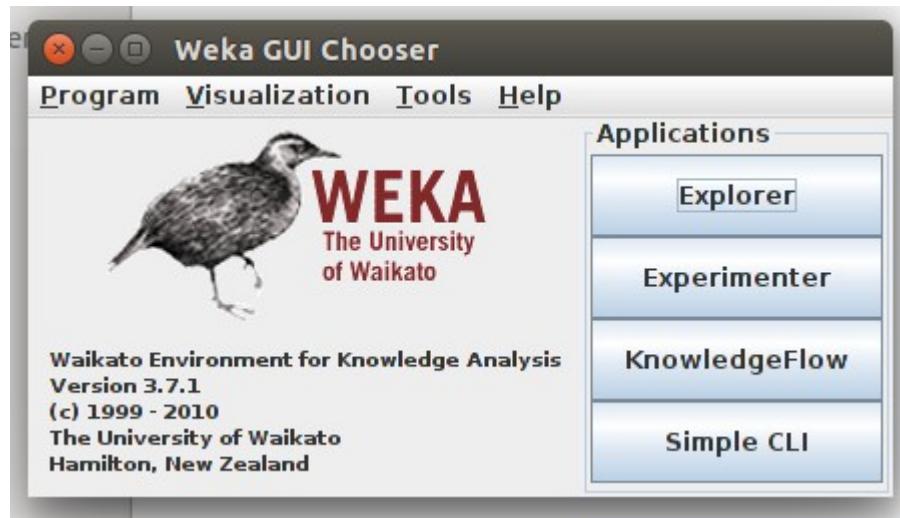


**WEKA**

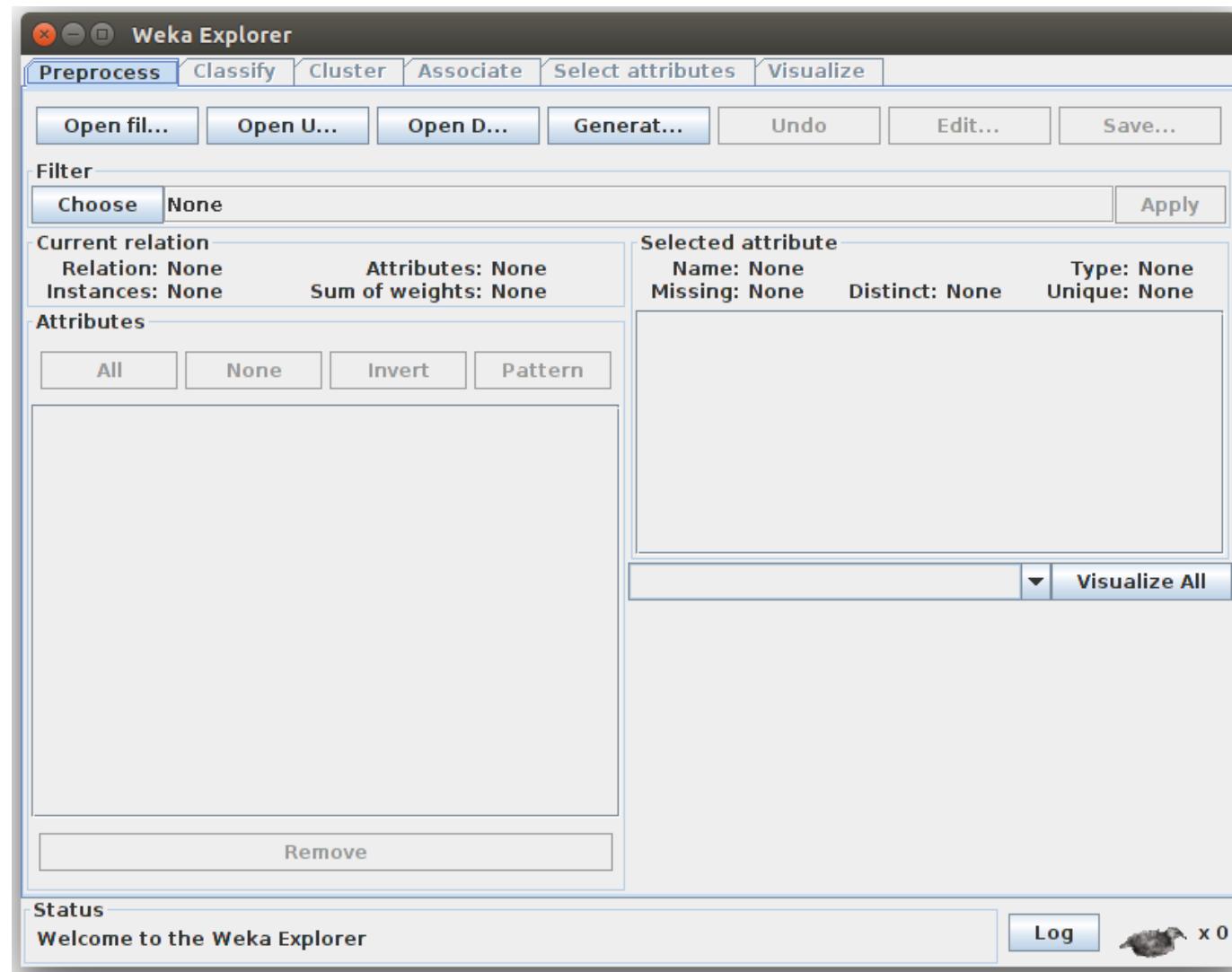
The workbench for machine learning

# Regresión

- Hacer click en el botón “Explorer”.



# Regresión

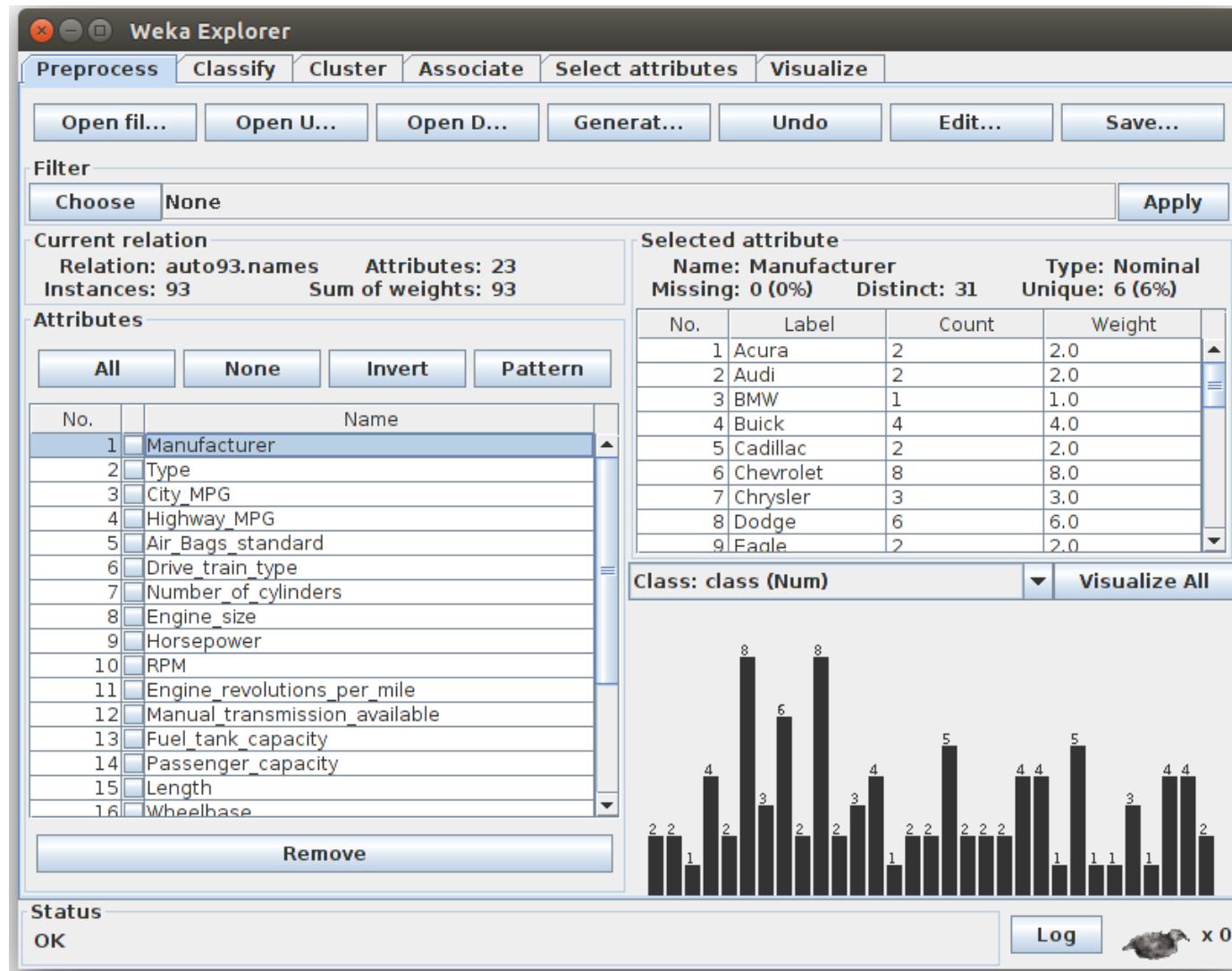


# Regresión

- **Preprocess:** Abre y modifica conjuntos de datos.
- **Classify:** Aplica algoritmos de clasificación y regresión.
- **Cluster:** Aplica algoritmos de *clustering*.
- **Associate:** Reglas de asociación.
- **Select attributes:** Seleccionar los mejores atributos.
- **Visualize:** Visualizar el conjunto de datos.

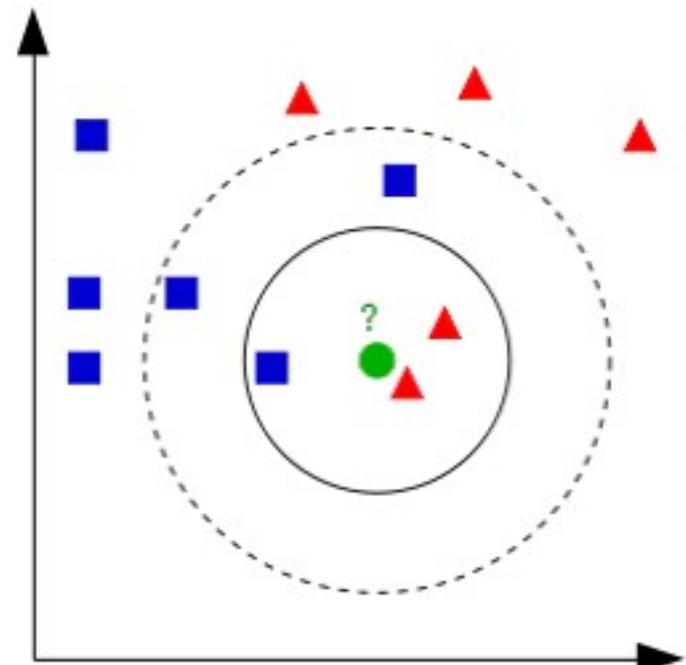
# Regresión

- Hacer click sobre "open file" y vamos a abrir el fichero “auto93.arff”.



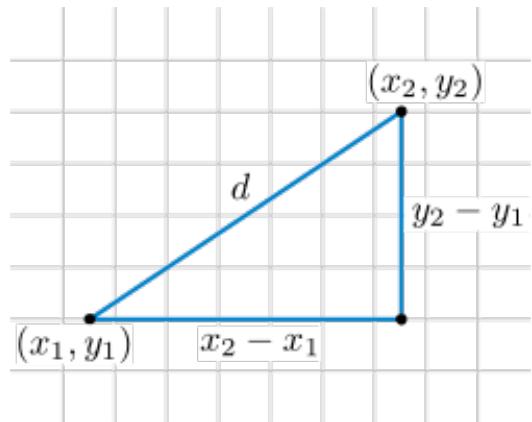
# *k*-vecinos más cercanos

- Busca los *k* vecinos más cercanos del ejemplo a predecir.
  - Se normalizan los atributos numéricos (Restando el mínimo y dividiendo entre el rango).
  - Se calcula la distancia entre cada ejemplo (sumando las distancias de cada atributo)
    - En nominales, la distancia es 0 si son iguales o 1 si son distintos.
    - En numéricos es la diferencia de valores.
  - Se eligen los *k* vecinos más cercanos.
  - Se predice:
    - la moda (clasificación)
    - la media (regresión).



# Cálculo de distancia

## 2 dimensiones



$$d_E(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$$

## n dimensiones

$$d_E(P, Q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

# *k*-vecinos más cercanos

- Vamos a *Classify* → *Choose*.
- Elegimos *lazy* → *Ibk*.
- Seleccionamos *Use training set*
- En *Show properties*:
  - K=1 (por el momento)
  - Opcional: *Distance weighting* hace la media ponderada por distancia)
- *More options*. En *output predictions* ponemos *plain text*.
- Hacer click en *start*.

# *k*-vecinos más cercanos

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose **IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""**

Test options

Use training set

Supplied test set Set...

Cross-validation Folds 10

Percentage split % 70

More options...

(Num) class

Start Stop

Result list (right-click for options)

Classifier output

weka.gui.GenericObjectEditor

weka.classifiers.lazy.IBk

About

K-nearest neighbours classifier.

More Capabilities

KNN 1

batchSize 100

crossValidate False

debug False

distanceWeighting No distance weighting

doNotCheckCapabilities False

meanSquared False

nearestNeighbourSearchAlgorithm Choose **LinearNNSearch -A "weka.core.**

numDecimalPlaces 2

windowSize 0

Open... Save... OK Cancel

Status OK Log x 0

```
IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
```

```
weka.classifiers.lazy.IBk
```

```
KNN 1
```

```
batchSize 100
```

```
crossValidate False
```

```
debug False
```

```
distanceWeighting No distance weighting
```

```
doNotCheckCapabilities False
```

```
meanSquared False
```

```
nearestNeighbourSearchAlgorithm Choose LinearNNSearch -A "weka.core."
```

```
numDecimalPlaces 2
```

```
windowSize 0
```

# $k$ -vecinos más cercanos

- Qué vemos en la pantalla:
  - Valor actual, predicción y error
  - *Mean absolute error.* La media de todos los errores.

# ¿Cómo evaluar los errores justamente?

- Si usamos para evaluar el propio conjunto de entrenamiento nos va a dar un error muy bajo (0 con vecinos más cercanos).
- No sabemos si el algoritmo funciona bien con ejemplos nunca vistos, es decir, si generaliza bien.
- Solución: Dividir el conjunto de datos en entrenamiento y test.

# ¿Cómo evaluar los errores justamente?

## Particiones train y test:

Original Data

X <sub>1</sub>	X <sub>2</sub>	X <sub>p</sub>	Y

train\_test\_split()



X\_train

X <sub>1</sub>	X <sub>2</sub>	X <sub>p</sub>

y\_train

Y

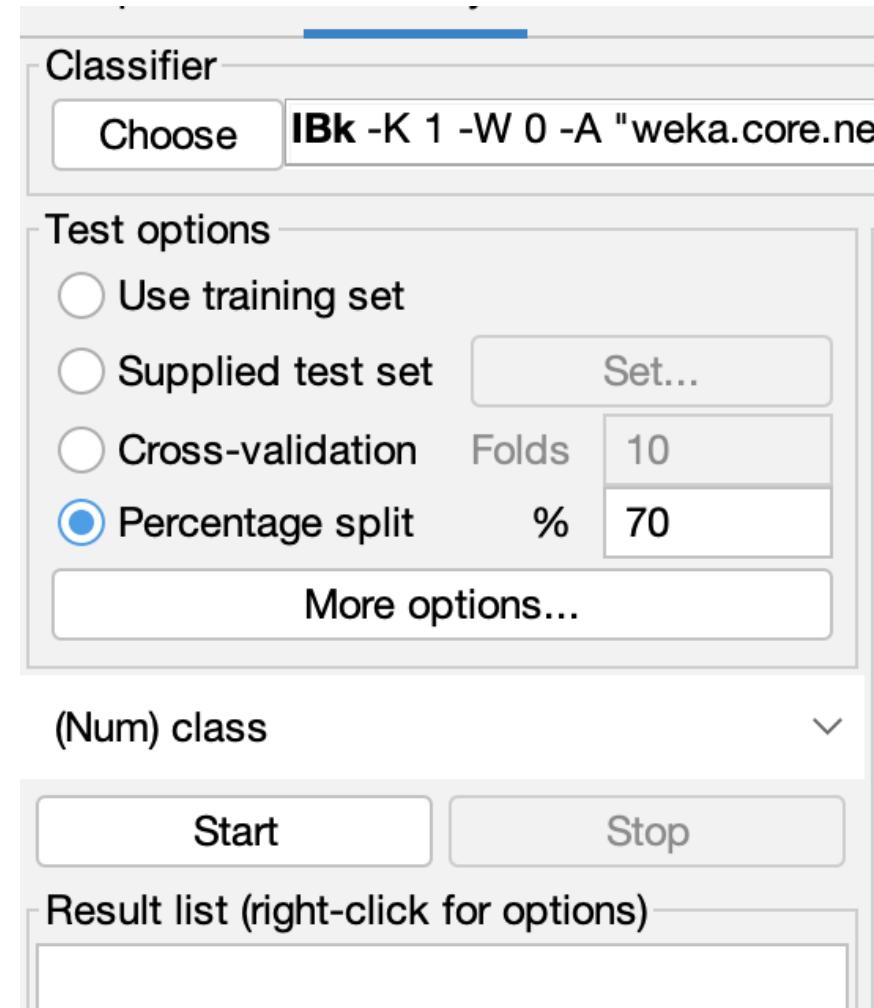
X\_test

X <sub>1</sub>	X <sub>2</sub>	X <sub>p</sub>

y\_test

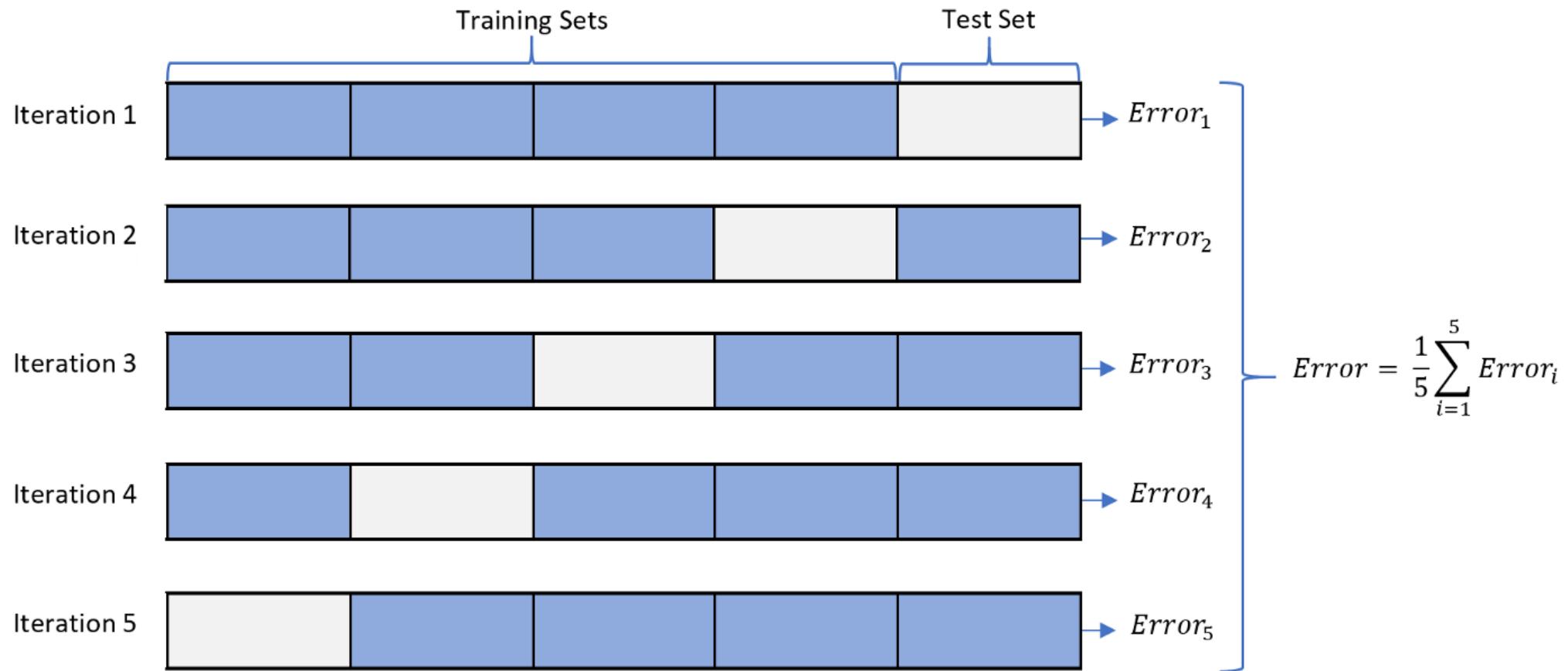
Y

# *k*-vecinos más cercanos



# ¿Cómo evaluar los errores justamente?

## Validación cruzada:



# $k$ -vecinos más cercanos

Test options

Use training set

Supplied test set

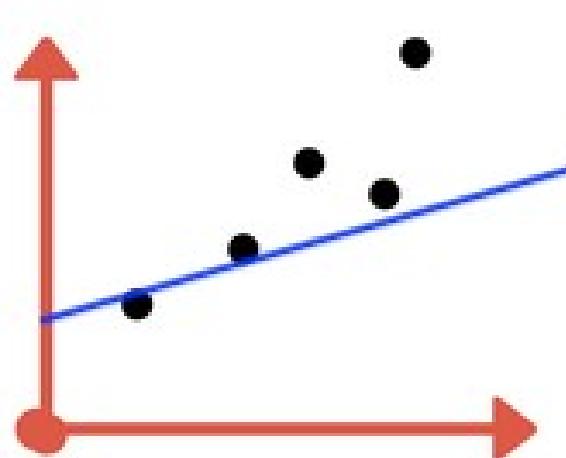
Cross-validation      Folds   
                                  %

Percentage split

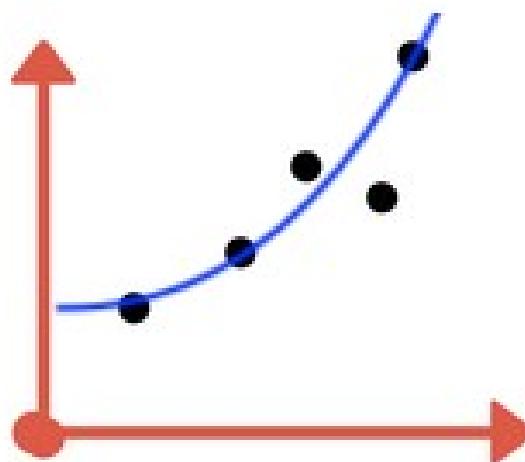
(Num) class

Result list (right-click for options)

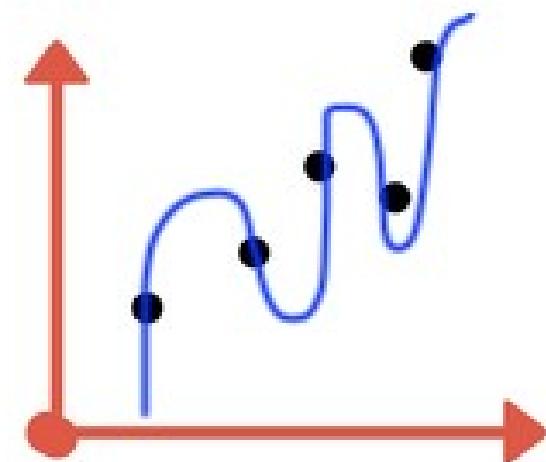
# Mala generalización



Subajuste

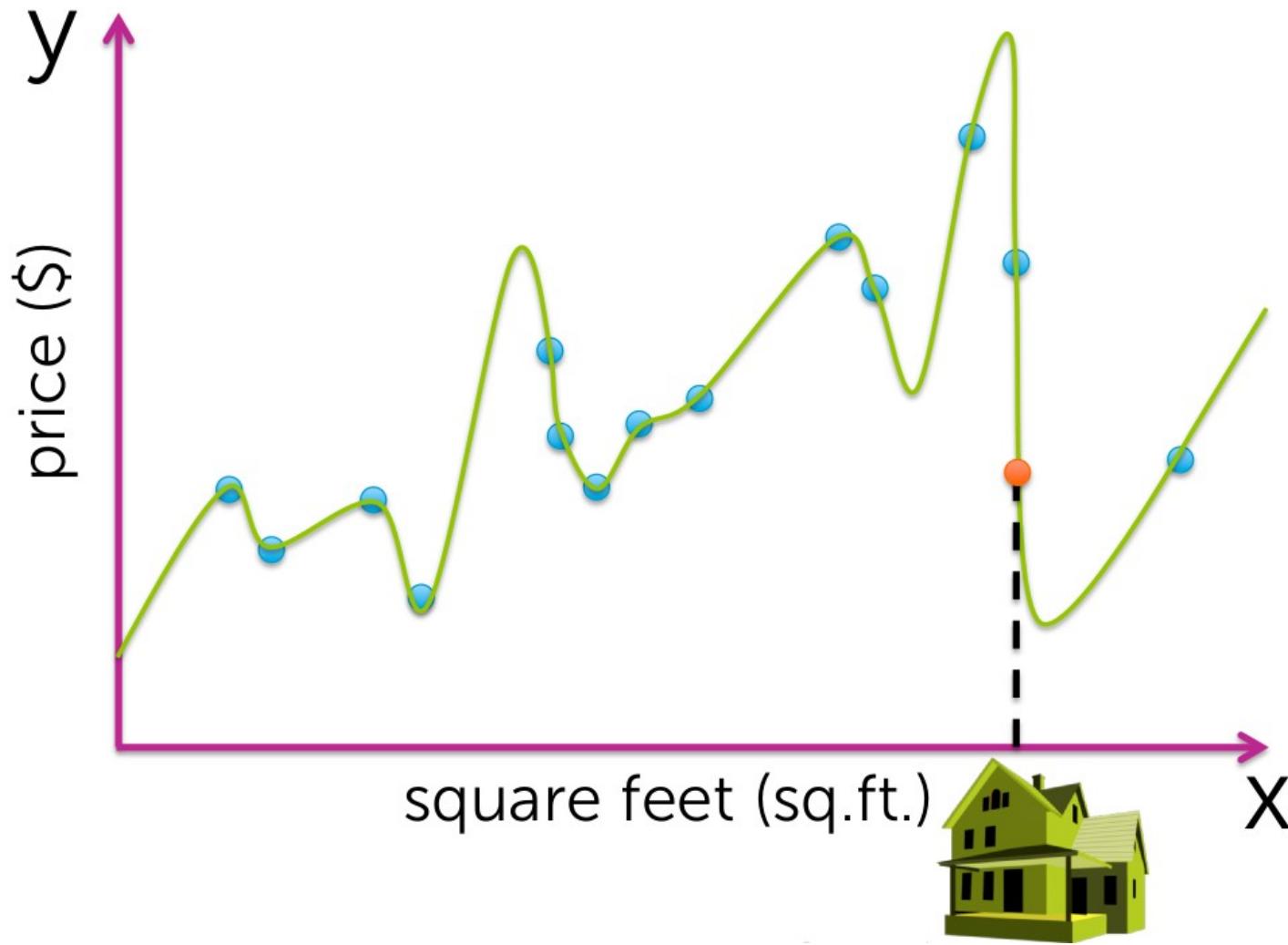


Correcto



Sobreajuste

# Mala generalización



# Regresión lineal

- Se quiere aproximar el valor a predecir ( $y$ ) mediante combinación lineal de los atributos (que ahora solo pueden ser numéricos).

$$F(a_1, a_2, \dots, a_n) = y = x_1a_1 + x_2a_2 + \dots + x_na_n$$

- Ejemplo: predecir la nota a partir de la media de horas de estudio y el número de faltas.

$$\text{Nota} = x_1*\text{horas} + x_2*\text{faltas} + x_3$$

- Hallar las  $X$ s que mejor se ajustan a los datos.

# Regresión lineal

- Los detalles matemáticos son complicados  
(ver [http://web.uam.es/personal\\_pdi/ciencias/cifus/biologia/metodos/ME4.pdf](http://web.uam.es/personal_pdi/ciencias/cifus/biologia/metodos/ME4.pdf))
- Lo vamos a hacer con weka.
- Abrimos bodyfat.arff
  - Predecir la grasa corporal a partir de la densidad del cuerpo, edad, altura, peso, diámetro de distintas partes del cuerpo...
- Elegimos functions/linearRegression

# Regresión Lineal

- Weka calcula el modelo de regresión

```
-410.2167 * Density +  
    0.0124 * Age +  
    0.0253 * Chest +  
    0.0314 * Abdomen +  
    446.1513
```

- ¿Cuál será la grasa corporal de un paciente?

Densidad = 1.05, Edad 30, Pecho 90, Abdomen 70

20.270765 % de grasa corporal

# Clasificación

- Tenemos un conjunto de datos.
- Cada ejemplo tiene un montón de atributos, esta vez en lugar de querer predecir un valor numérico, se quiere predecir una categoría.
- Se quiere encontrar el modelo que minimice el número de errores.

# Clasificación

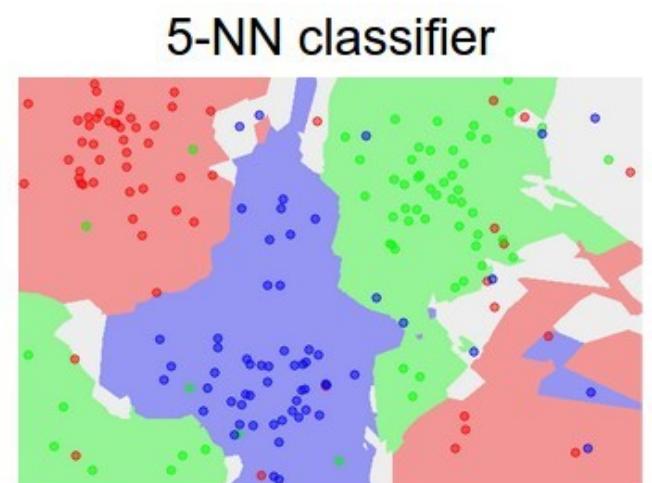
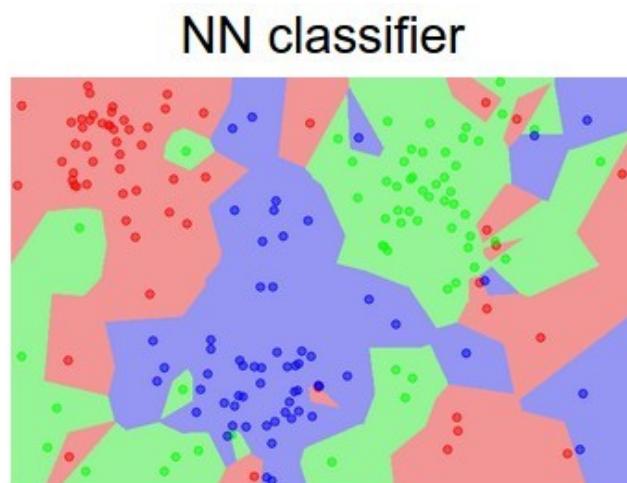
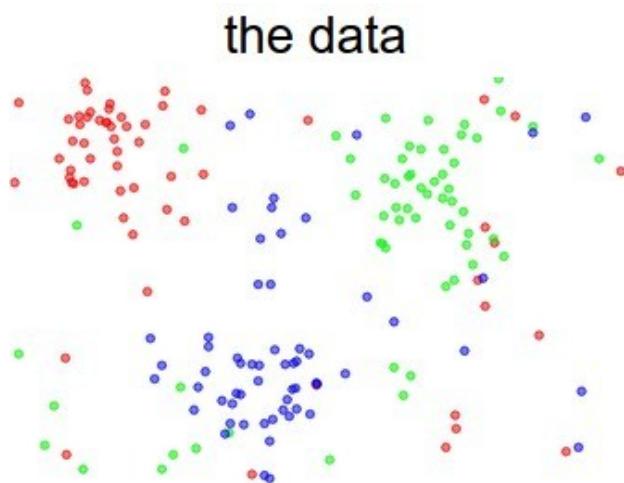
- **Predecir el diagnóstico de un paciente.**
  - **Atributos:** Valores de análisis de sangre, de orina etc.
  - **Clase:** Tiene Lupus Si/No
- **Clasificador de SPAM.**
  - **Atributos:** Frecuencias de determinadas palabras en el email.
  - **Clase:** Es SPAM Sí/No
- **OCR (Optical character recognition)**
  - **Atributos:** valores de los píxeles de un dígito de 16x16.
  - **Clase:** el carácter que se corresponde con la imagen.

# K-vecinos más cercanos

Cada atributo es un eje.

Queremos dividir el espacio en regiones en las que cada una pertenezca a una clase diferente.

*k*-NN es el clasificador más sencillo.

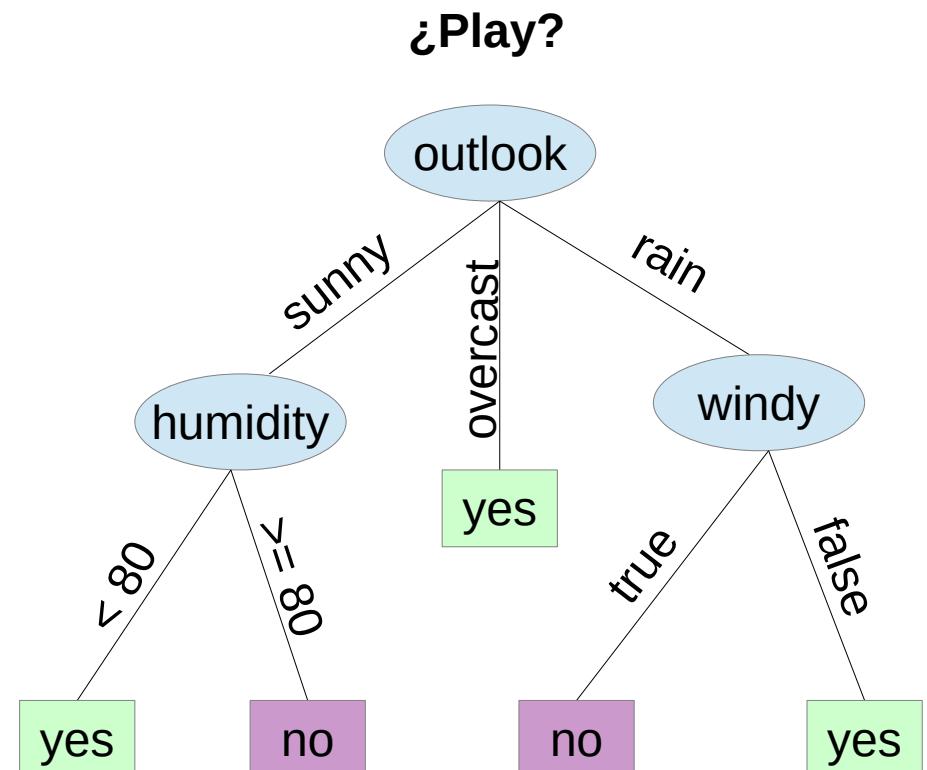
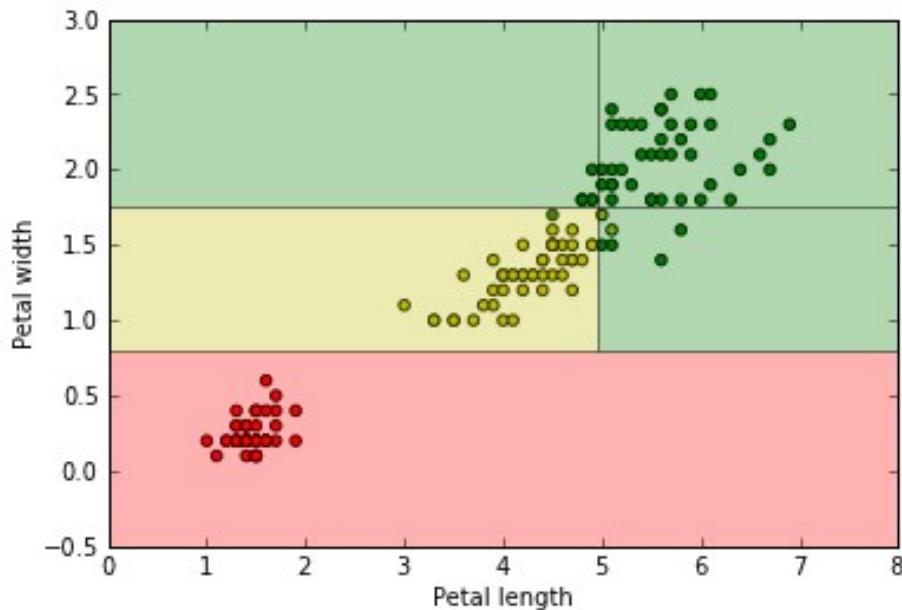


# Árboles de decisión

Otro clasificador sencillo y popular son los árboles de decisión.  
Parten el espacio.

Cada partición es una hoja. Predicen la clase mayoritaria en cada hoja.

También funcionan en regresión. Devuelven la media de los ejemplos que caen en esa hoja.



# Árboles de decisión

- Algoritmo de construcción de árboles de decisión: árbol de decisión
  - Si (casi) todos los ejemplos son de la misma clase:
    - Hacer una hoja.
  - Si no: → bifurcación en dos ramas
    - Best-Atr = atributo que mejor divide los ejemplos.
    - Se parte ejemplos en ejemplos1 y ejemplos2 usando Best-Atr.
    - `árbolDecisión(ejemplos1)`
    - `árbolDecisión(ejemplos2)`

# Árboles de decisión

- ¿Cómo saber cual es el mejor atributo?  
Se ordenan los ejemplos de mayor a menor usando ese atributo.

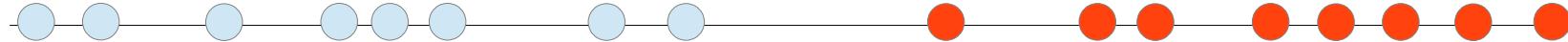
Este atributo es malo



Este atributo es regular



Este atributo es bueno, hay un punto que separa las dos clases



# Árboles de decisión

- ¿Cómo se calcula esto?
  - Ganancia de información (infoGain)

$$\begin{aligned} \text{infoGain}([x,y]) &= \text{entropía}\left(\frac{x}{x+y}, \frac{y}{x+y}\right) \\ &= -\frac{x}{x+y} \log\left(\frac{x}{x+y}\right) - \frac{y}{x+y} \log\left(\frac{y}{x+y}\right) \end{aligned}$$

Por ejemplo. Hay 16 valores. En un punto hay 6 rojas y 2 azules para un lado y 6 azules y dos rojas para otro.

El valor de ese atributo es  $8/16 * \text{InfoGain}(6,2) + 8/16 * \text{InfoGain}(2,6)$

# Árboles de decisión

- Pero no os preocupéis, que los hace Weka
- Abrimos iris.arff
- En *classify* elegimos trees → J48
- Pulsamos sobre el botón *start*.

# Árboles de decisión



Iris Setosa



Iris Virginica

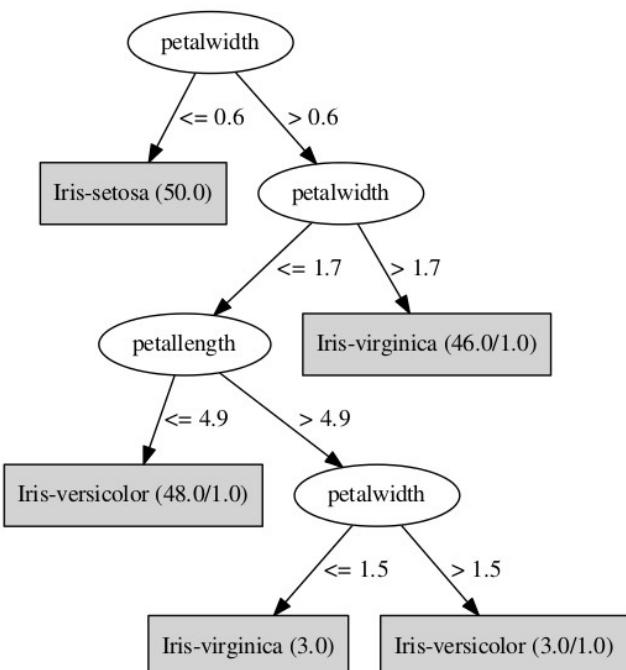


Iris Versicolor



```
@RELATION iris  
@ATTRIBUTE sepalength REAL  
@ATTRIBUTE sepalwidth REAL  
@ATTRIBUTE petallength REAL  
@ATTRIBUTE petalwidth REAL  
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}
```

```
@DATA  
5.1,3.5,1.4,0.2,Iris-setosa  
4.9,3.0,1.4,0.2,Iris-setosa  
4.7,3.2,1.3,0.2,Iris-setosa  
4.6,3.1,1.5,0.2,Iris-setosa  
5.0,3.6,1.4,0.2,Iris-setosa  
5.4,3.9,1.7,0.4,Iris-setosa  
...  
5.0,2.3,3.3,1.0,Iris-versicolor  
5.6,2.7,4.2,1.3,Iris-versicolor  
5.7,3.0,4.2,1.2,Iris-versicolor  
5.7,2.9,4.2,1.3,Iris-versicolor  
6.2,2.9,4.3,1.3,Iris-versicolor  
5.1,2.5,3.0,1.1,Iris-versicolor  
5.7,2.8,4.1,1.3,Iris-versicolor  
...  
6.3,3.3,6.0,2.5,Iris-virginica  
5.8,2.7,5.1,1.9,Iris-virginica  
5.9,3.0,5.1,1.8,Iris-virginica
```



# Ensembles

- La idea de los *ensembles* es que en lugar de tener un clasificador o un regresor tenemos muchos.
- Cada uno tiene su opinión y votan.
- Un *ensemble* es un conjunto de expertos.  
Las claves son:
  - Los expertos deben ser precisos.
  - Los expertos son diversos.

# Ensembles. Diversidad.

- ¿Como hacemos *ensembles* diversos?
  - Haciendo que cada uno se estudie unos temas diferentes y se especialice en cosas diferentes.
  - Tenemos un conjunto de datos:
    - Hacemos múltiples versiones de ese conjunto de datos: remuestreo de ejemplos, distintos grupos de atributos, etc.

# Ensembles: Bagging y Boosting

- **Bagging:** Cada clasificador entrena con un remuestreo aleatorio del conjunto de datos.
  - Ejemplo: Hay 10 temas y cada experto se lee 10 temas al azar. Un experto se puede leer un mismo tema varias veces y otro ninguna.
- **Boosting:** El clasificador N se entrena con los ejemplos que le han resultado más difíciles al clasificador N-1.
  - Ejemplo. Hay 10 temas, el primer experto se lee 10 temas al azar, luego le hacen preguntas de todos los temas y el segundo experto se lee 10 temas, pero los temas que ha fallado el 1 salen más veces y los que ha acertado menos.

# Ensembles: ¿por qué?

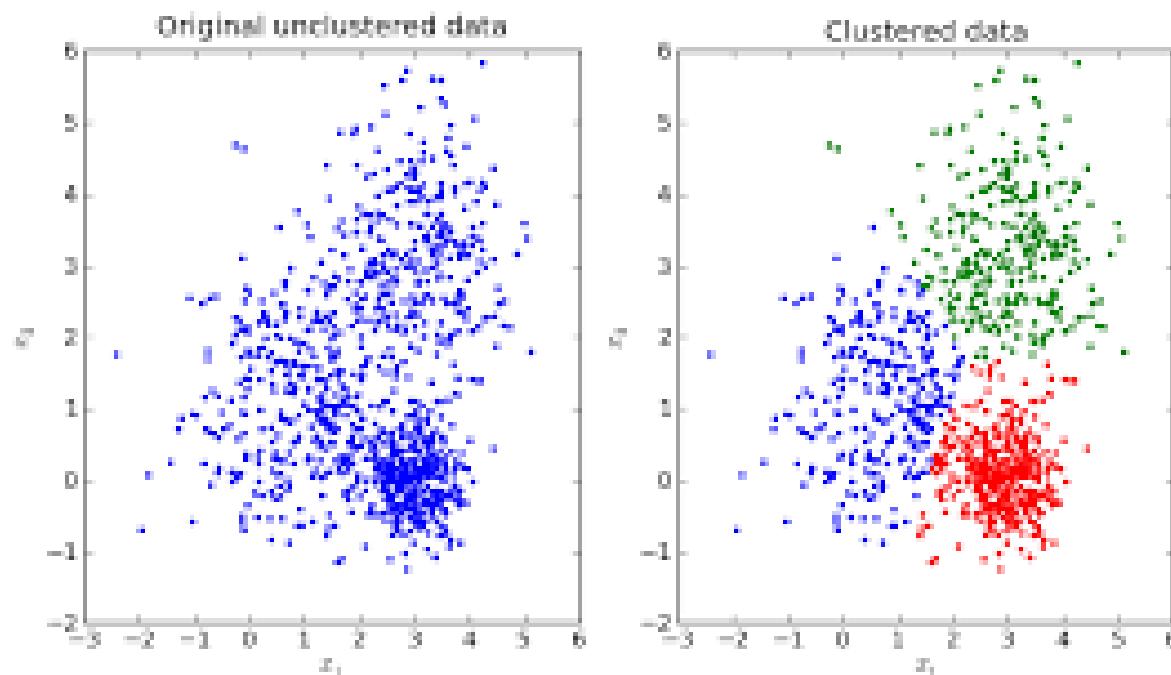
- Es más fácil entrenar muchos clasificadores buenos y combinarlos en uno muy bueno que tratar de hacer un clasificador muy bueno desde el principio.



# Aprendizaje no supervisado

# Clustering

- El clustering o agrupación no tiene en cuenta la clase, ya que se desconoce. Consiste en agrupar los ejemplos en grupos que tengan características parecidas.



# Clustering

- **Segmentación de imágenes médicas.**
- **Estudios de mercados:** Identificar clientes con gustos similares.
- **Análisis de redes sociales:** Identificar intereses a partir de datos de redes sociales.
- **Búsqueda de imágenes:** Imágenes similares.
- **Geología:** Búsqueda de regiones con características del suelo similares (búsqueda de petróleo).

# K-means

- Parte de un conjunto de semillas, ejemplos elegidos aleatoriamente.
  - Usando distancias asigna cada ejemplo a la semilla más cercana.
  - La nueva semilla es la media de todos los ejemplos asignados a esa semilla,
- Se repiten los pasos anteriores hasta que el algoritmo converge (las asignaciones no cambian)
- <https://www.youtube.com/watch?v=5I3Ei69I40s>
- <https://www.youtube.com/watch?v=BVFG7fd1H30>

# K-means

- Volvemos a Weka.
- En *cluster* elegimos *SimpleKMeans*.
- *Dentro de SimpleKMeans modificamos numClusters a placer.*
- Pinchamos en *Ignore attributes* y seleccionamos *class*.
- Start para ejecutar el algoritmo.
- Click derecho en el resultado → *Visualize cluster assignments*

# K-means

Preprocess Classify Cluster Associate Select attributes Visualize

**Clusterer**

Choose SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-

**Cluster mode**

Use training set  
 Supplied test set Set...  
 Percentage split % 66  
 Classes to clusters evaluation (Nom) class  
 Store clusters for visualization

Ignore attributes Start Stop

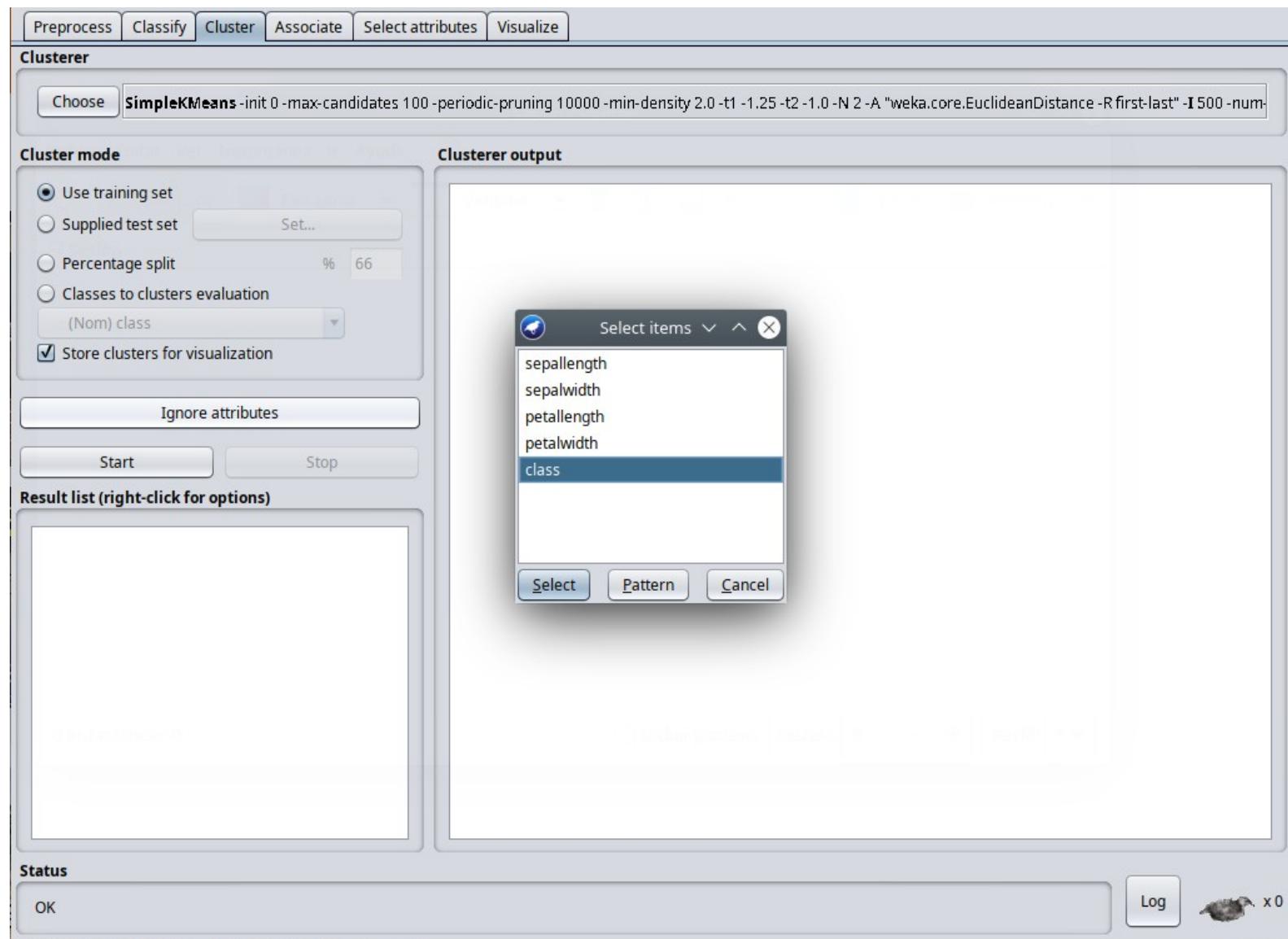
**Result list (right-click for options)**

**Clusterer output**

Select items

sepallength  
sepalwidth  
petallength  
petalwidth  
class

Status OK Log  x0



# K-means



# Deep Learning

# Deep Learning

Métodos de Machine Learning  
(supervisados y no supervisados) que usan  
**redes neuronales profundas** que  
aprenden de **grandes conjuntos de datos**.

- Son capaces de aprender directamente a partir de imágenes, vídeos, audios, textos... sin necesidad de convertirlos en atributos numéricos o categóricos.
- Son capaces de aprender de grandes conjuntos de datos (Big Data).

# Deep Learning

## Features for machine learning

Images



Image



Vision features

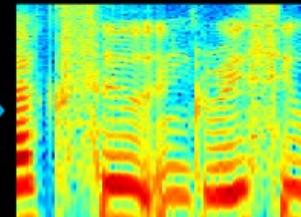


Detection

Audio



Audio



Audio features

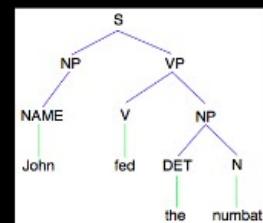


Speaker ID

Text



Text



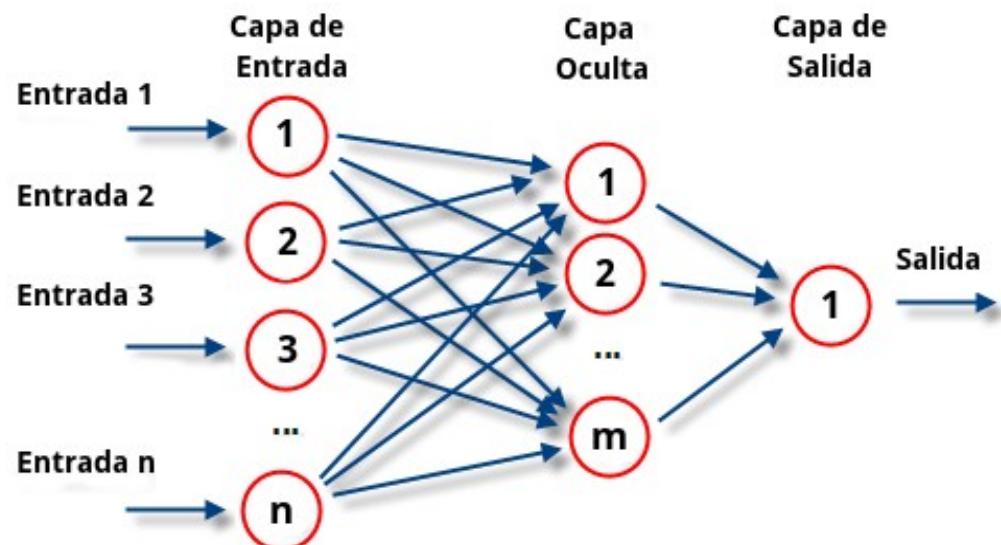
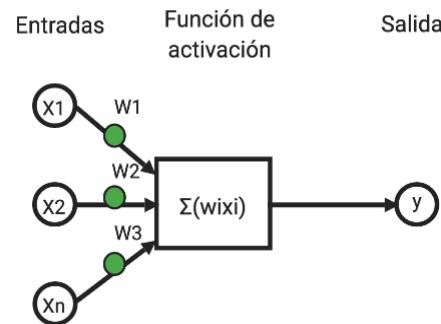
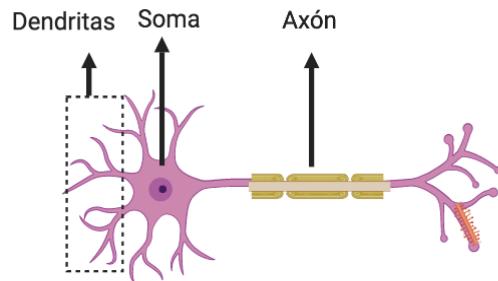
Text features

Web search

...

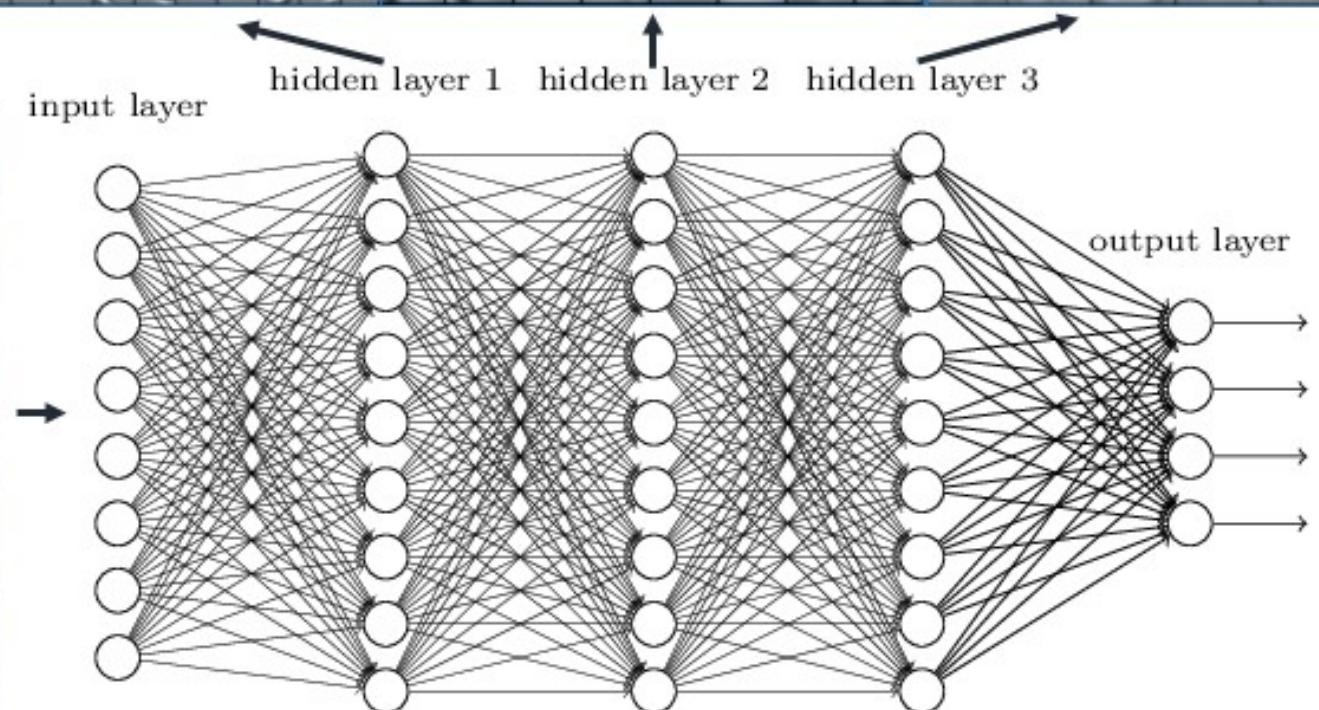
# Deep Learning

Las **redes neuronales artificiales** tratan de imitar el comportamiento del sistema neuronal de los mamíferos.



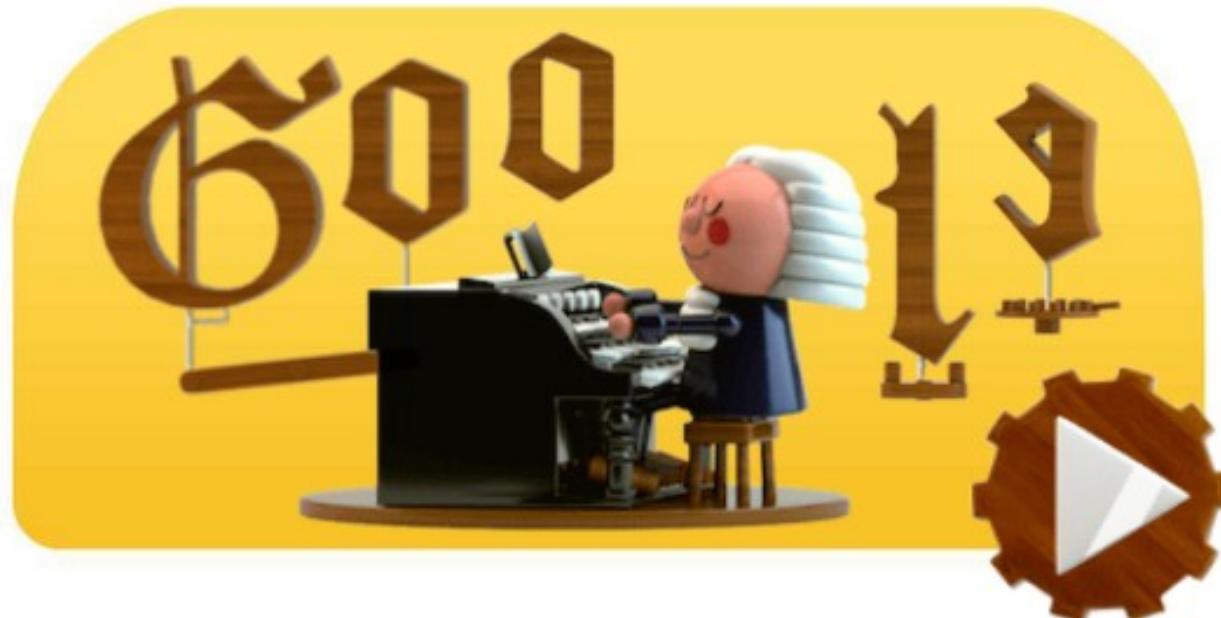
# Deep Learning

Deep neural networks learn hierarchical feature representations



# Caso de Uso

- Composición/Armonización de canciones.



- Acceder: <https://www.google.com/doodles/celebrating-johann-sebastian-bach>

# Caso de Uso

- Creaciones artísticas



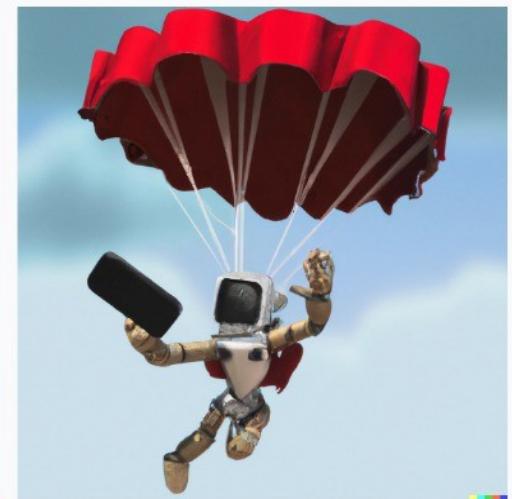
- Deep dream generator:

<https://deeppdreamgenerator.com/#gallery>

# Caso de Uso

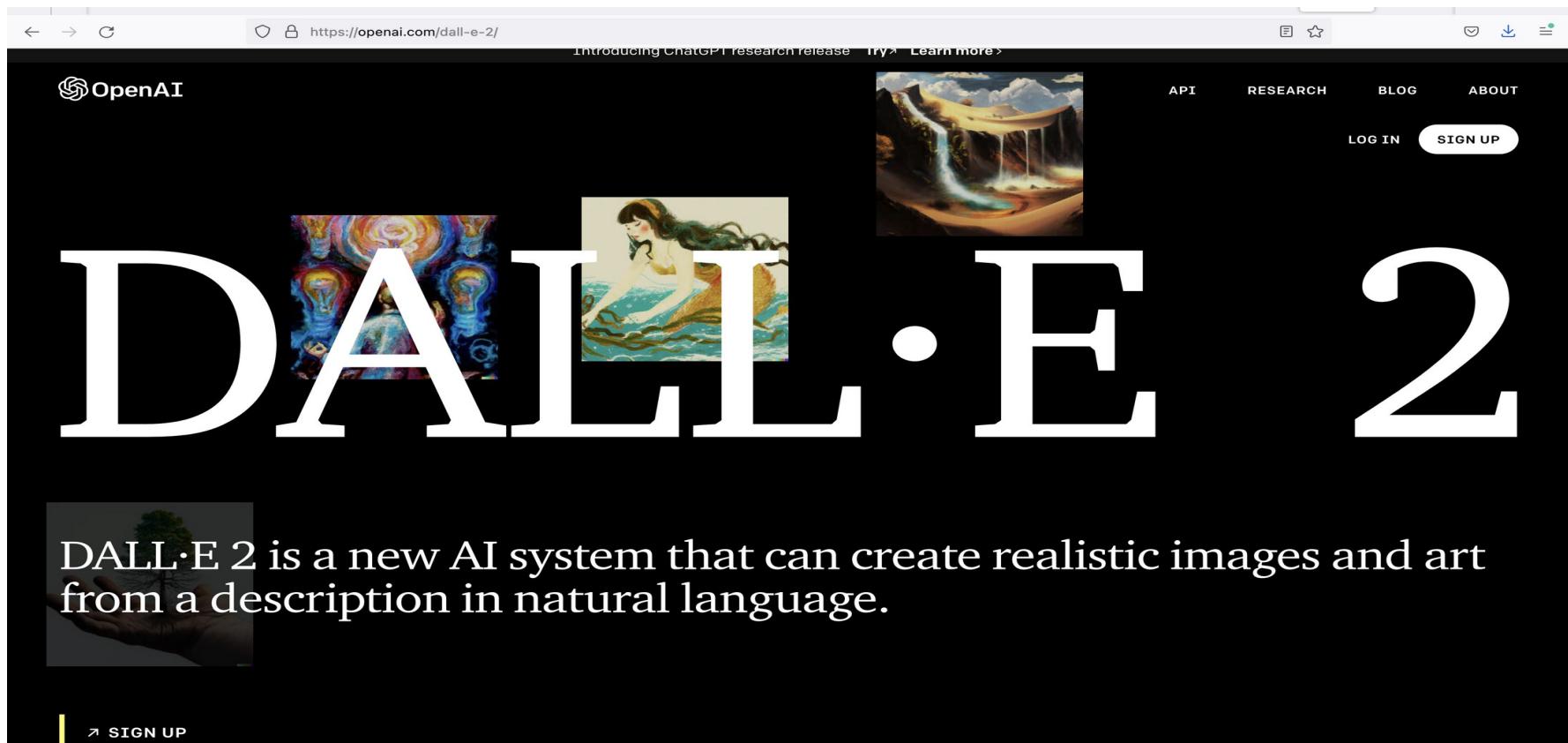
A robot taking a selfie while falling on a red parachute, digital art.

Generate



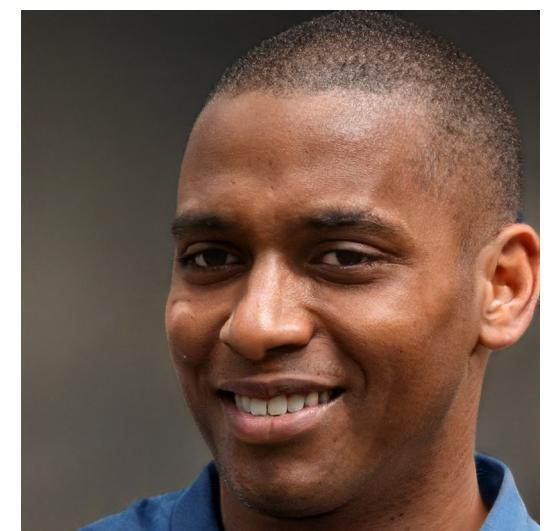
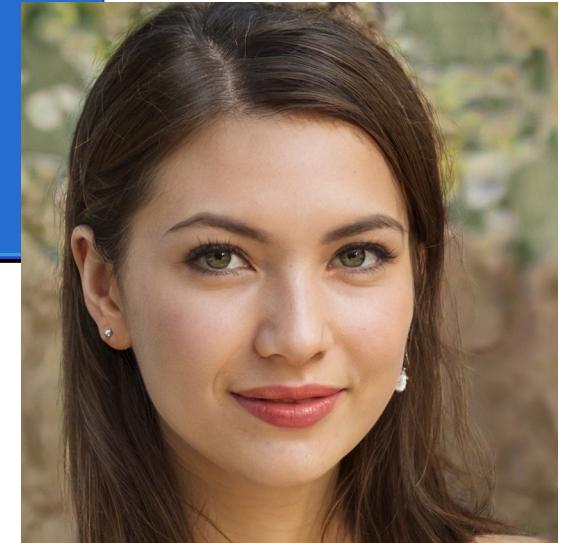
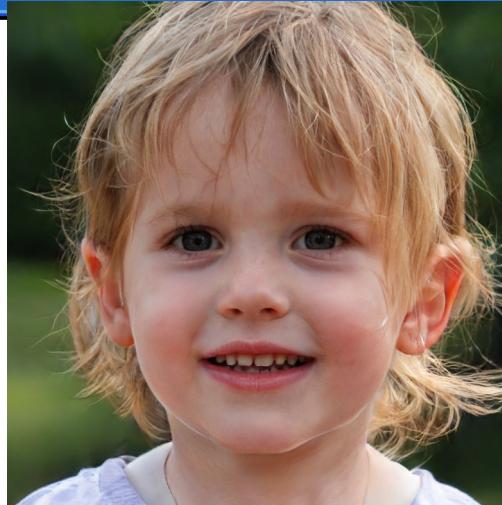
- Stable Diffusion: <https://stability.ai/blog/stable-diffusion-public-release>

# Caso de Uso



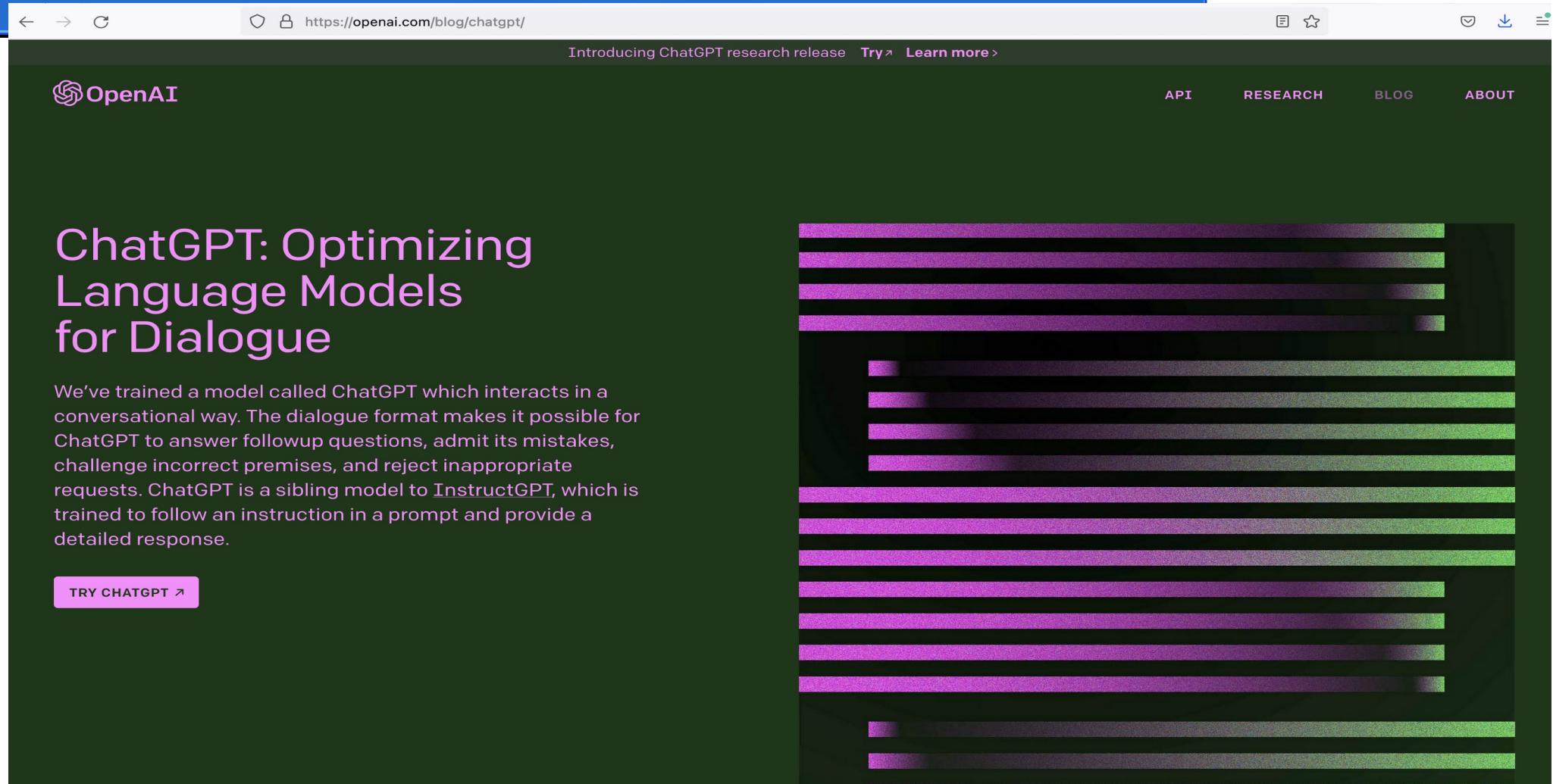
- Dalle 2: <https://openai.com/dall-e-2/>

# Caso de Uso



- <https://thispersondoesnotexist.com/>

# Caso de Uso

A screenshot of a web browser displaying the "Introducing ChatGPT research release" page from openai.com. The page has a dark background with a grid of colored bars on the right. The main content area features the OpenAI logo, a title about ChatGPT, and a paragraph describing its conversational capabilities. A "TRY CHATGPT" button is visible at the bottom left.

Introducing ChatGPT research release [Try ↗](#) [Learn more >](#)

**OpenAI**

**ChatGPT: Optimizing Language Models for Dialogue**

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests. ChatGPT is a sibling model to InstructGPT, which is trained to follow an instruction in a prompt and provide a detailed response.

[TRY CHATGPT ↗](#)

- ChatGPT: <https://openai.com/blog/chatgpt/>

# Comentarios finales

≡ **IT**WORLD Today's playlist ▾



Social media icons: Twitter, LinkedIn, Facebook, YouTube, RSS, Search.

Home > Cloud Computing

## Google reports strong profit, says it's 'rethinking everything' around machine learning

Google's products will use that form of AI even more in the future



By James Niccolai

IDG News Service | October 22, 2015

FOLLOW

---

### MORE GOOD READS



Understanding Google's Alphabet structure (think, alpha bet!)

---

Google restructures, naming parent company Alphabet

*Machine learning is a core, transformative way by which we're rethinking everything we're doing. We're thoughtfully applying it across all our products, be it search, ads, YouTube or Play*

Sundar Pichai, CEO, Google

# Comentarios finales

- 1985 Backpropagation (redes neuronales)
- 1993 C4.5 (J48 árboles de decisión)
- 1994 Bagging.
- 1997 Boosting.
- 1997 Deep Blue gana a Kasparov.
- 2004 DARPA Grand Challenge.
- 2009 Google Car.
- 2011 Watson gana al Jeopardy
- 2015 Google, Facebook y Baidu superan la inteligencia humana en imagenet
- 2016 AlphaGo vs Lee Sedol (9 de marzo).
- ...
- Robots, nanotecnología, asistentes virtuales, medicina preventiva ...

# Comentarios finales

- La minería de datos y la inteligencia artificial están revolucionando el mundo y aún están empezando.
- Cada año hay más datos, más capacidad de cómputo, más y mejores algoritmos.
- Cada año se resuelven nuevos problemas usando minería de datos.