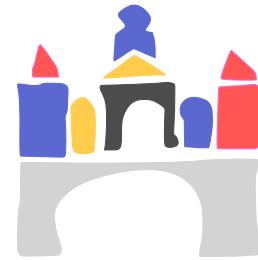


Introducción a la Minería de Datos



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 esta charla se encuentra en el siguiente enlace:

<https://github.com/alvarag/BIEMineriaDeDatos>

- Descargar el fichero “data.zip” y descomprimirlo en el ordenador.
- Descargar el fichero “weka.jar”.

¿Quiénes somos?

- Mario Juez Gil.
- José Francisco Díez Pastor.
- Álvar Arnaiz González.
- 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?

- La inteligencia artificial se dedica a la creación de sistemas informáticos con un comportamiento inteligente.
- La minería de datos consiste en la creación de sistemas que aprenden por sí mismos o que extraen conocimiento de los datos.
- Otros nombres muy de moda:
 - Data mining.
 - Machine Learning.
 - Big Data (con matrices).

¿Quiénes usan Minería de Datos?

- Google: Buscador, Google Photos (reconocimiento de imágenes), Youtube (reconocimiento de voz).
- Amazon, Netflix: Sistemas de recomendación.
- Apple: Siri (reconocimiento de voz), Watch (reconocimiento de actividades).
- Banca (predicción de fraude, predicción de riesgos...)
- Industria manufacturera, medicina, medioambientales...

Esquema general



Datos

```
@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
```

Instancias o ejemplos = 6

Atributos 4: Uno nominal (categórico),
uno numérico (real) y
uno binario (dos posibles valores).

La clase es binaria (por lo tanto, es un problema de clasificación).

Tipos de tareas

- Regresión
- Clasificación
- *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.
 - Casos prácticos.

Regresión

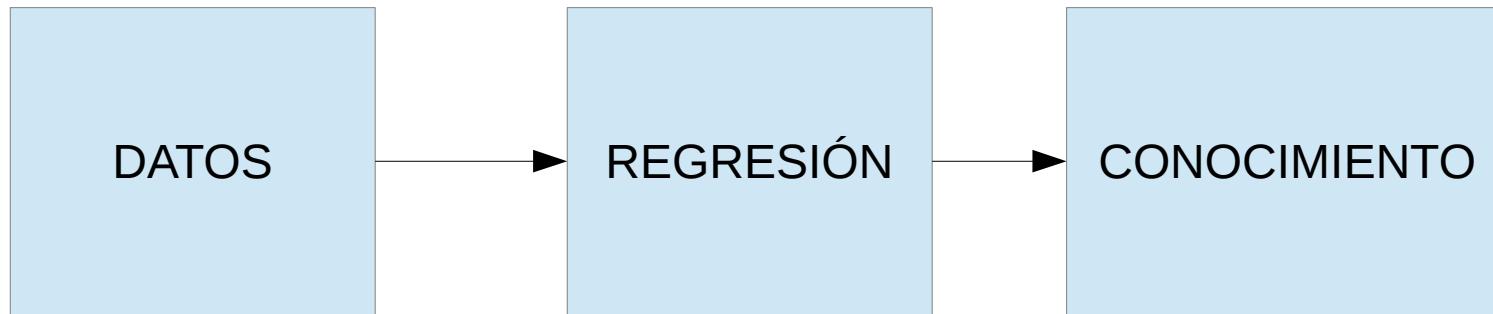
- Tenemos datos: un conjunto de ejemplos.
- Cada ejemplo se compone de diversas variables independientes (atributos) y una variable dependiente de tipo numérico (y).
- Se quiere hallar la función que relacione los atributos de entrada con la variable dependiente (con el menor error posible).

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

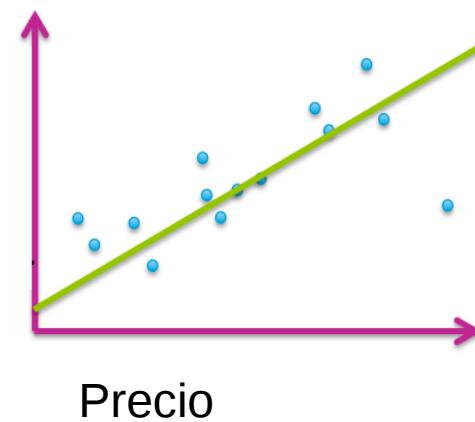
Regresión

- 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



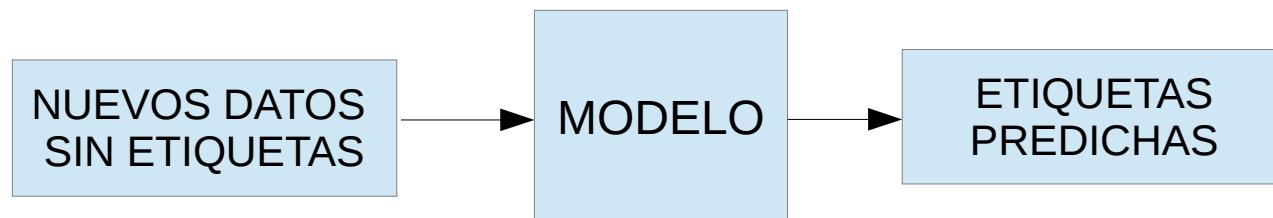
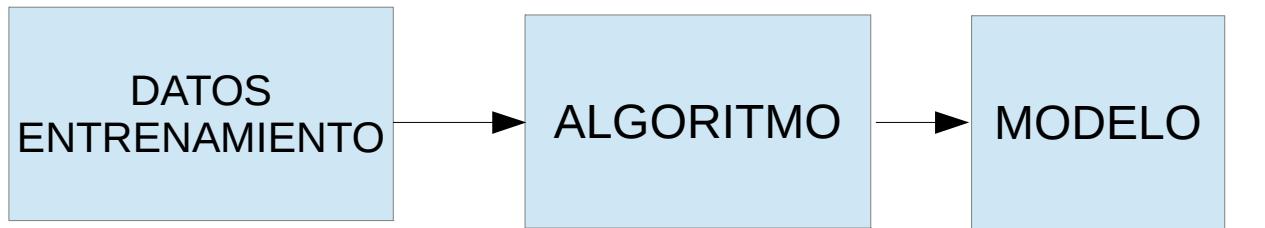
Características
de las casas



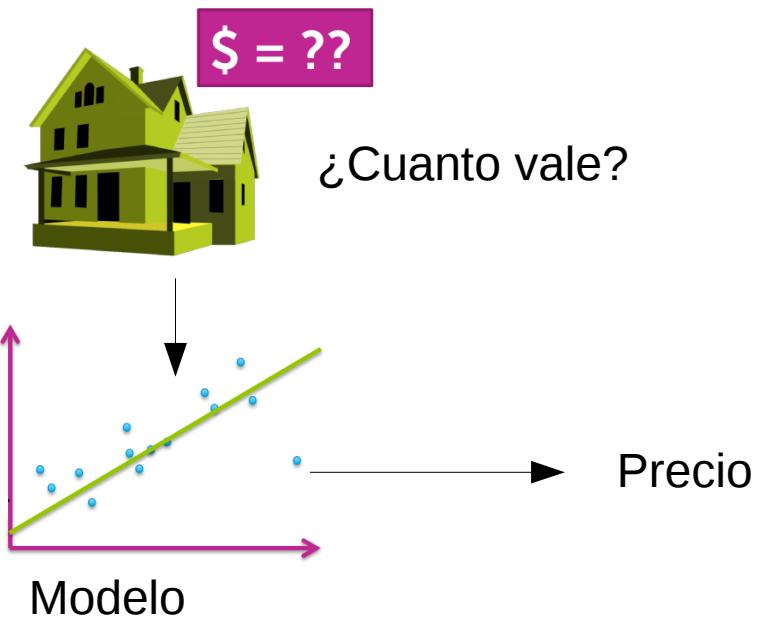
¿Cuánto vale?



Esquema general



Características de las casas



Regresión

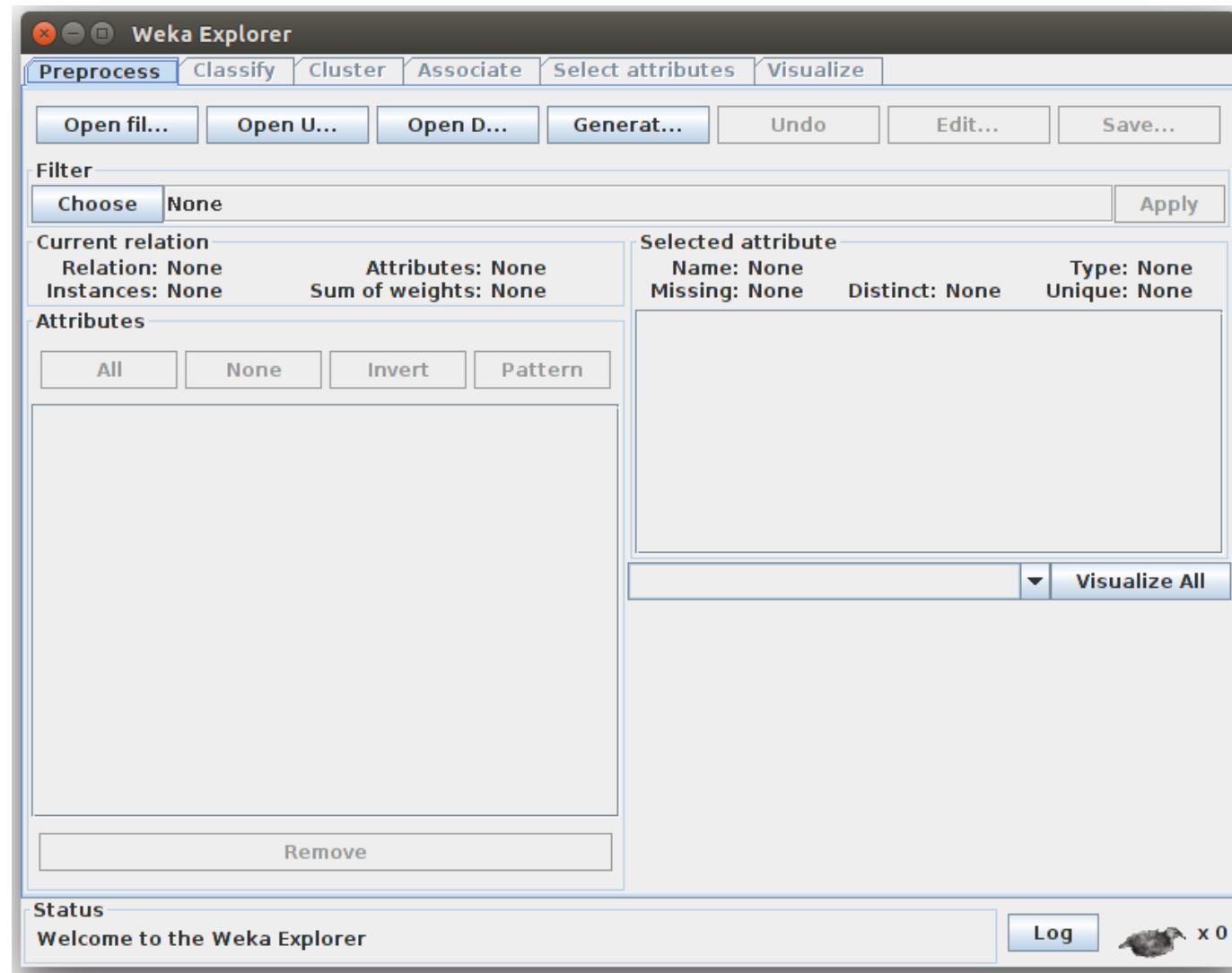
- Vamos a predecir el precio de coches.
- Vamos a utilizar algoritmos de Weka.

Regresión

- Hacer click en el botón “Explorer”.



Regresión

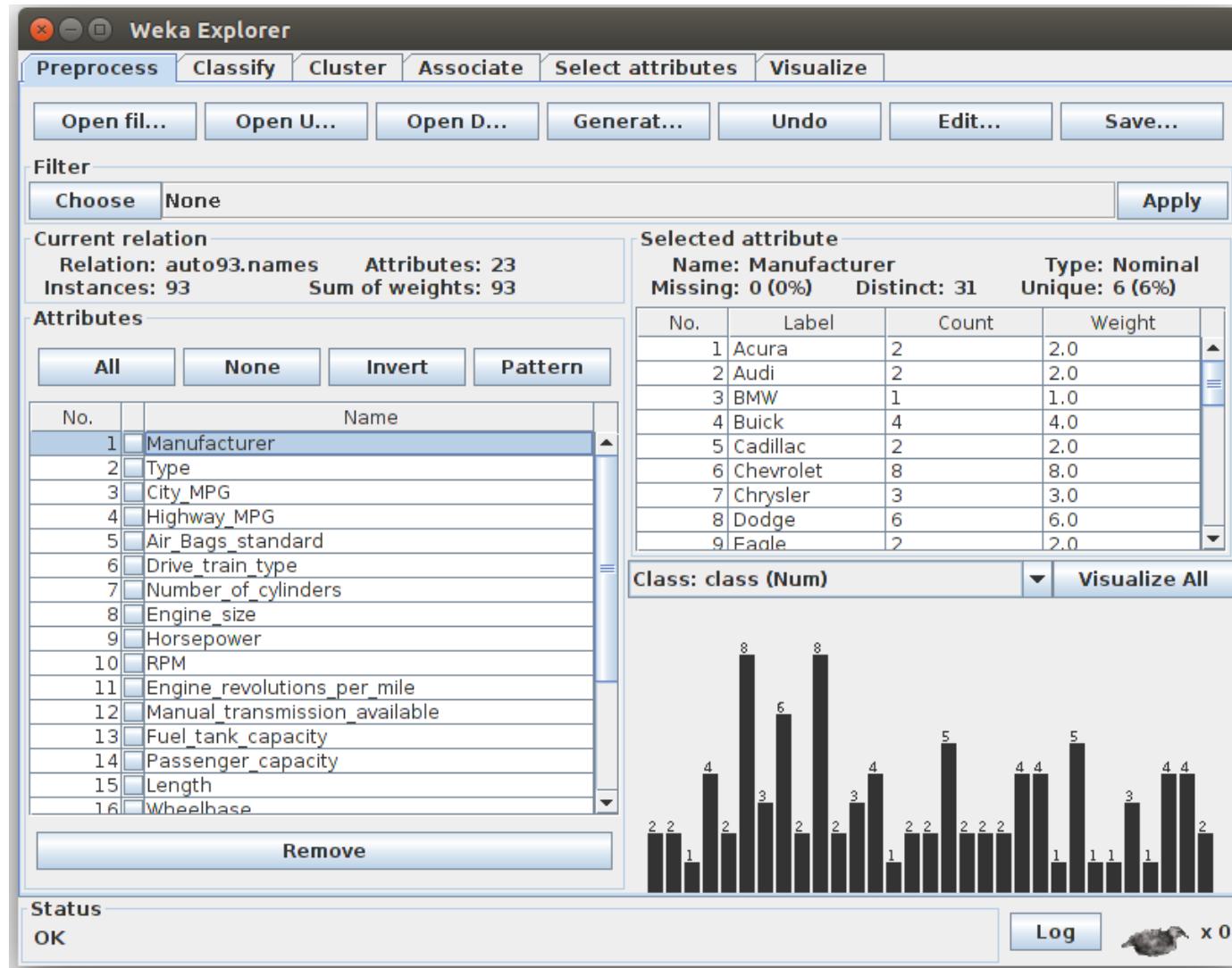


Regresión

- 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

- El algoritmo de clasificación y regresión más simple.
- 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.

k -vecinos más cercanos

- Conjunto de datos de 2 atributos: género y altura. Se predice el peso. Con $k = 2$.

Hombre, 160, 60

Mujer, 150, 45

Mujer, 170, 62

Hombre, 165, 65



Normalización. Restar 150, dividir entre 20

Hombre, 0.5, 60

Mujer, 0, 45

Mujer, 1, 62

Hombre, 0.75, 70

Atributos nominales: Distancia 0 si el valor es igual 1 si es diferente.

Atributos numéricos: La diferencia.

k -vecinos más cercanos

- Llega un caso que queremos predecir

Hombre, 162, ?

Normalizado vale

Hombre, 0.6, ?



Distancias

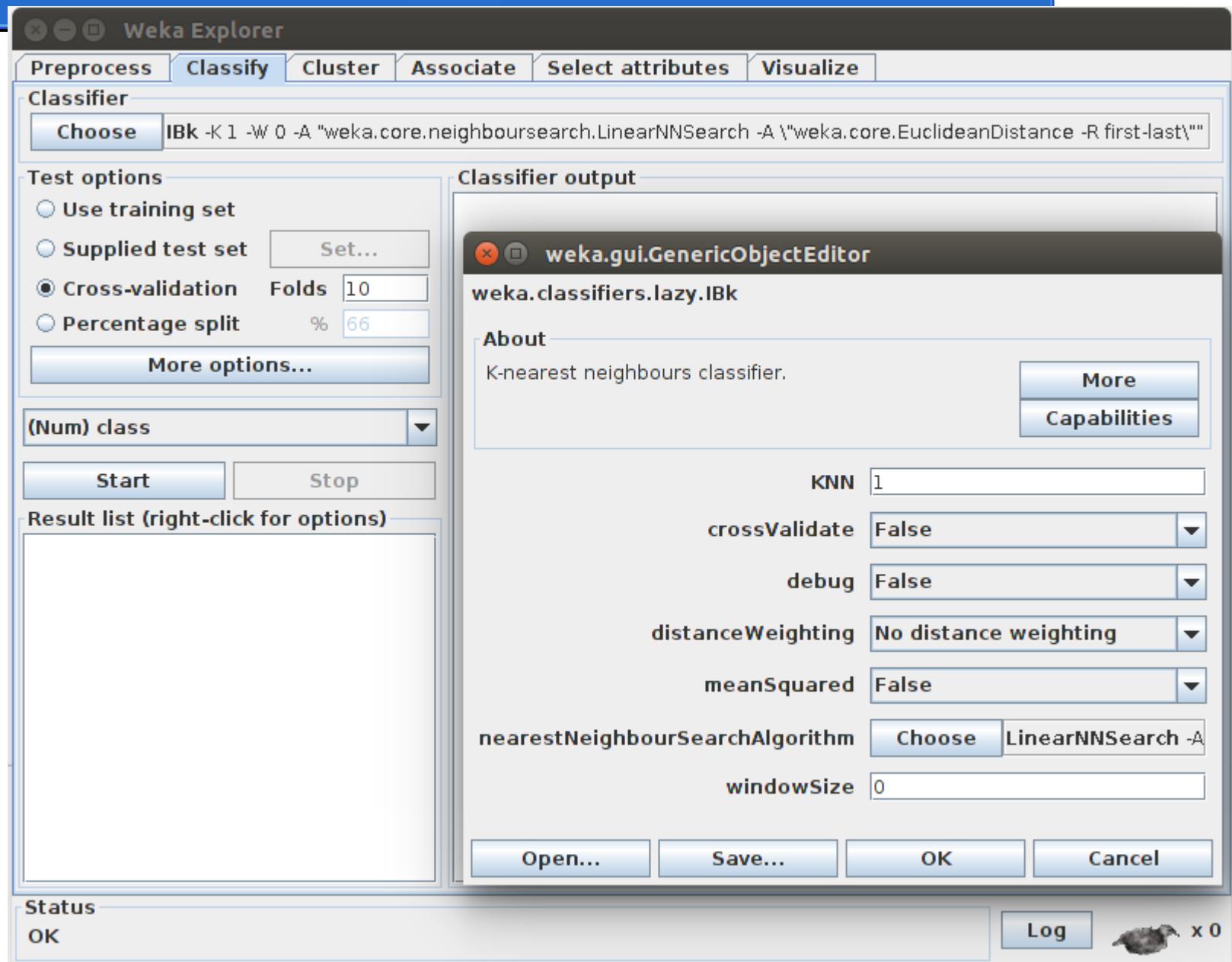
Hombre, 0.5, 60	0.1	←
Mujer, 0, 45	1.15	
Mujer, 1, 62	1.12	
Hombre, 0.75, 70	0.15	←

Se predice la media entre 60 y 70 = 65

k-vecinos más cercanos

- Vamos a *classify, choose*.
- Elegimos *lazy*, elegimos *Ibk*.
- (*Distance weighting* hace la media ponderada por distancia)
- *More options*. En *output predictions* ponemos *plain text*.
- Hacer click en *start*.

k-vecinos más cercanos



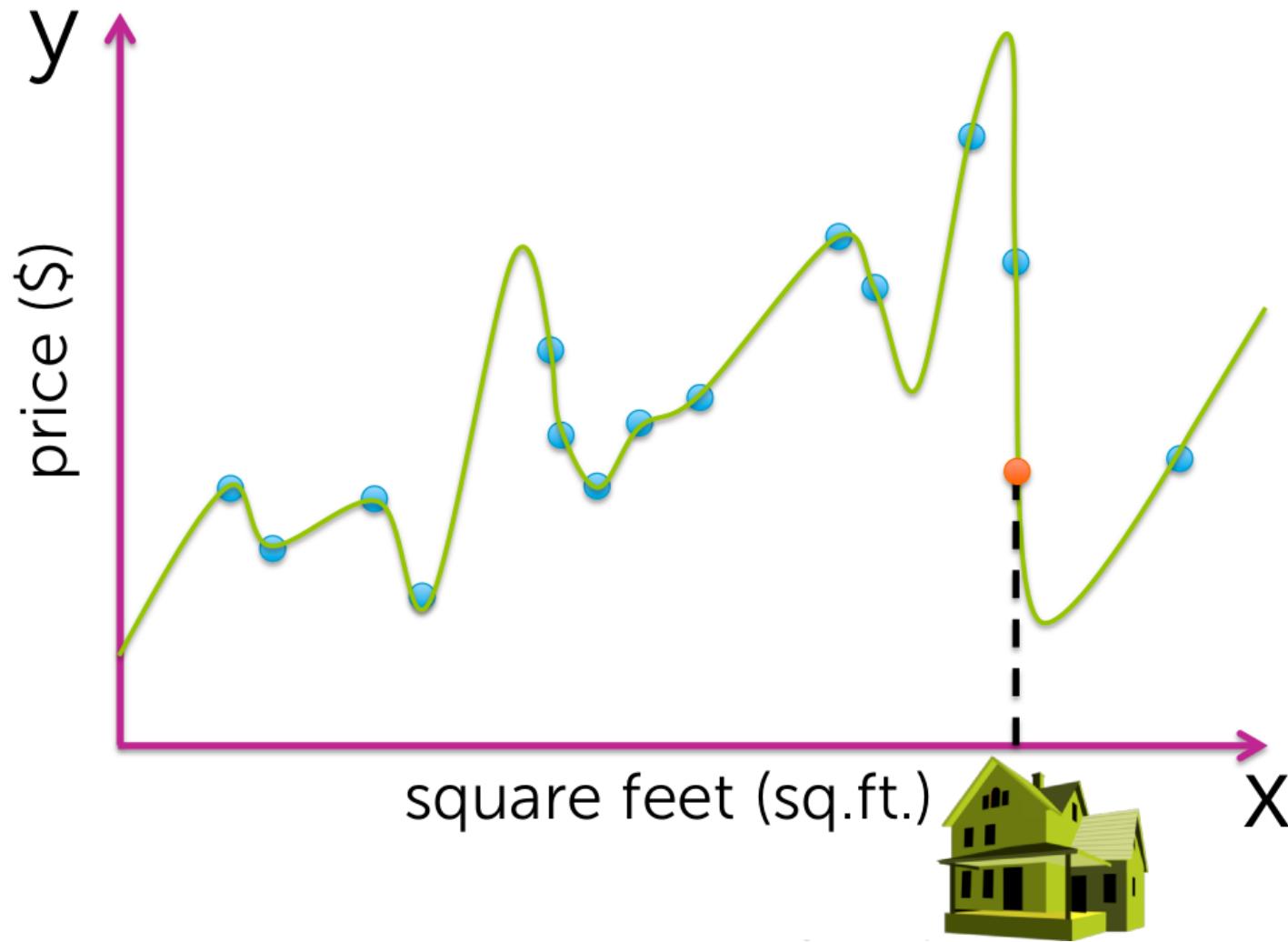
k -vecinos más cercanos

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

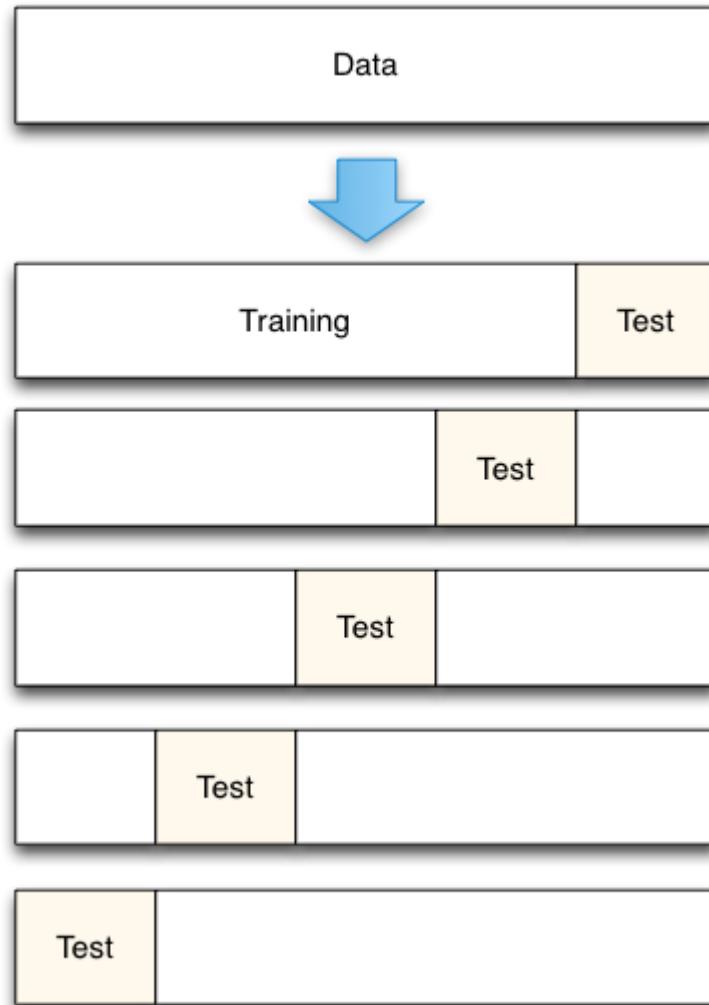
¿Cómo evaluar los errores justamente?

- 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. Generaliza bien.
- Solución: Dividir el conjunto de datos en entrenamiento y test.

Mala generalización



Validación cruzada



El error del algoritmo es la media de los errores de cada una de las particiones. En este dibujo son 5. En Weka eran 10.

Regresión lineal

- Se quiere aproximar el valor a predecir (y) mediante combinación lineal de los atributos (que ahora solo pueden ser numéricos).
- Ejemplo predecir la nota a partir de la media de horas de estudio y el número de faltas.
- Nota = $X_1 \cdot \text{horas} + X_2 \cdot \text{faltas} + X_3$
- 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)
- 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

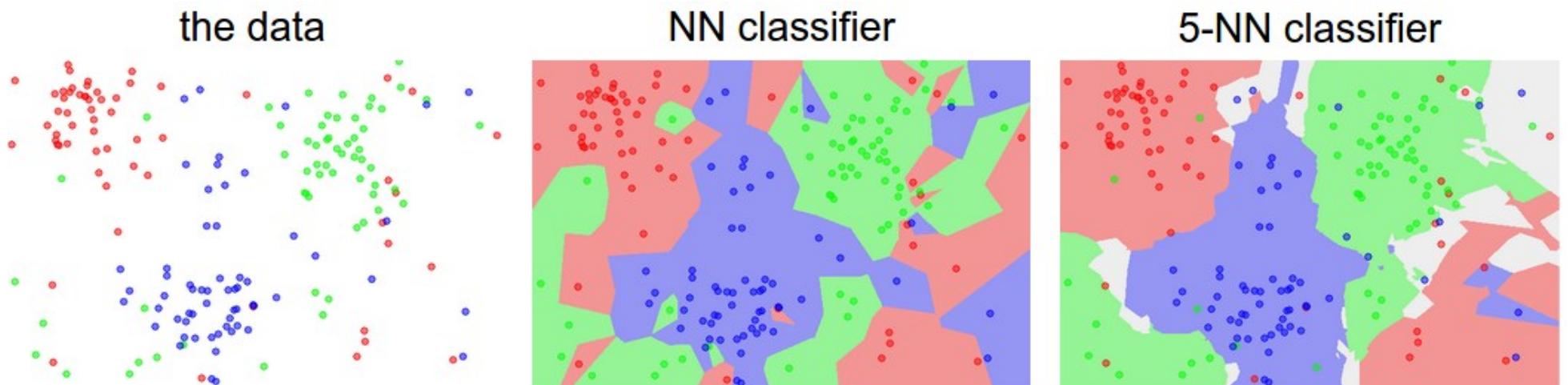
- Predecir el diagnóstico de un paciente.
 - Atributos: Valores de análisis de sangre, de orina etc. Clase: Tiene Lupus Si/No
- Clasificador de SPAM.
 - Atributos: Frecuencias de determinadas palabras en el email. Clase: Es SPAM Si/No
- OCR (Optical character recognition)
 - Los valores de los píxeles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

Clasificación

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.

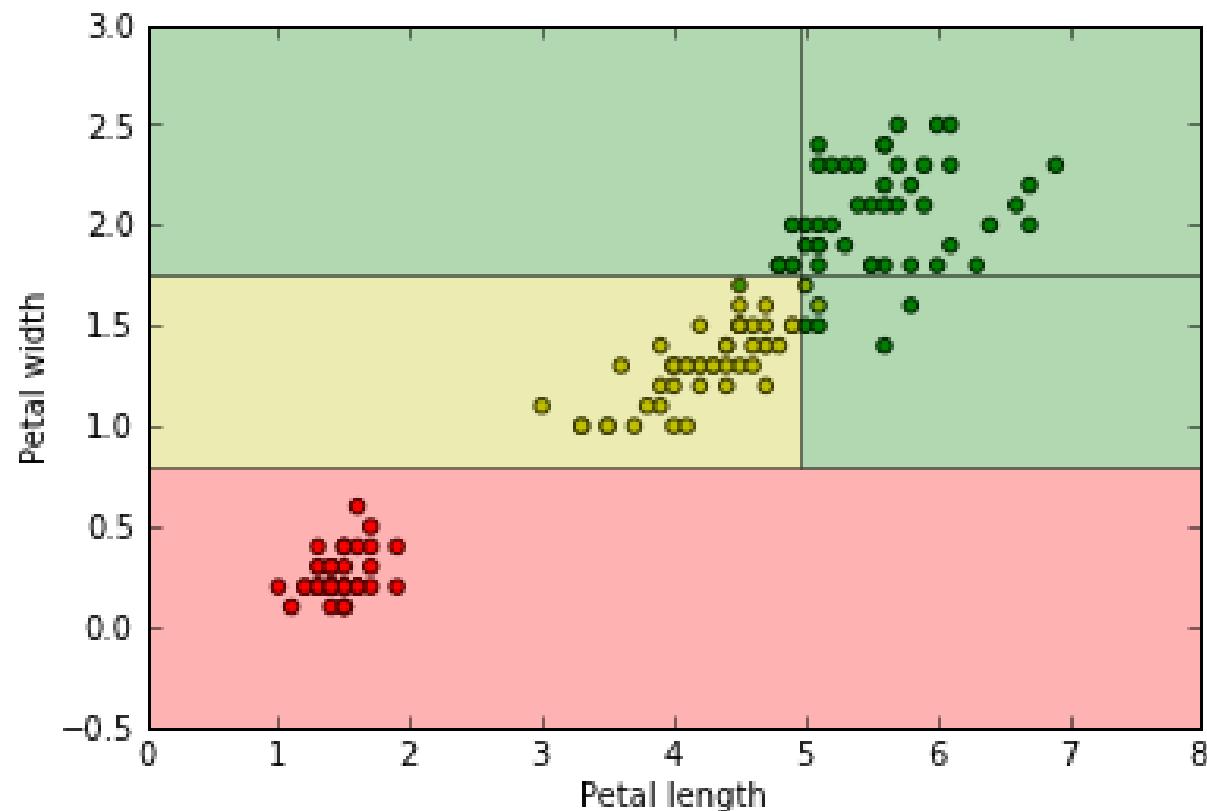


Árboles de decisión

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

- Los árboles de decisión partitionan los datos recursivamente
- Un proceso recursivo es un proceso que se reutiliza/llama/invoca a sí mismo.

Para entender recursividad. Nada que ver con los árboles

Fibonacci (n)

Si $n < 2$

Devuelve n

Si no

Devuelve Fibonacci (n-1) + Fibonacci (n-2)

0,1,1,2,3,5,8,13,21,34 ...

Árboles de decisión

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

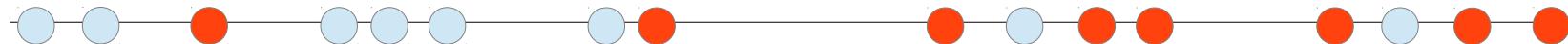
Árboles de decisión

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

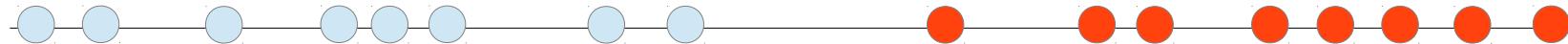
Este atributo es malo



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
- Le damos a start.

Árboles de decisión



Iris Setosa



Iris Virginica

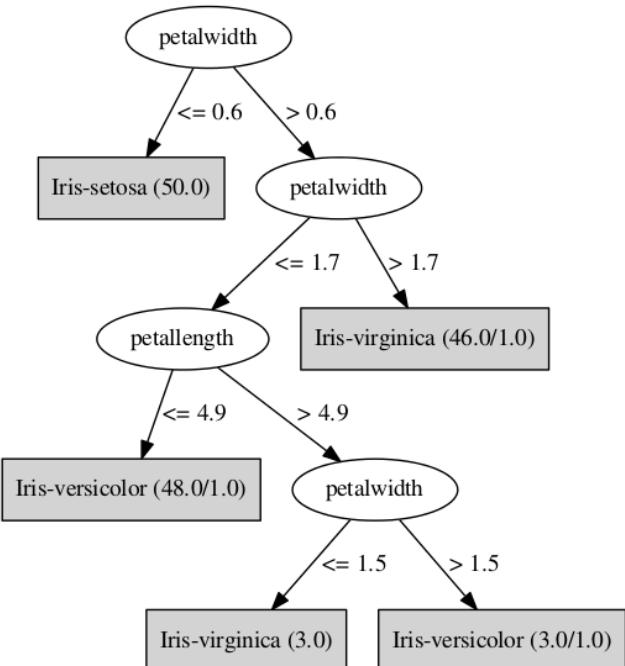


Iris Versicolor



```
@RELATION iris  
@ATTRIBUTE sepalength REAL  
@ATTRIBUTE sepawidth 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
```



Casos de Uso

- Reconocimiento de matrículas. Proyecto de bachillerato de excelencia.

Tutor: José Francisco Diez e Indalecio Ceballos. Alumna: Raquel Palacin.

- Detectar la matrícula es un problema de clasificación de dos clases: matrícula Si, matrícula No. Los atributos se calculan a partir de los píxeles de la imagen.
 - Reconocer la matrícula es un problema de clasificación donde hay tantos números como letras y dígitos.

Casos de Uso

Se entrena un clasificador con imágenes de matrículas (positivos) y con imágenes que no son matrículas (negativos).

Positivos

VI·8538·Y	AB·8538·V	A·1794·EN	AL·3434·AK	AV·8209·I
BA·0032·AG	IB·5745·DT	B·4819·XG	BU·9509·Z	CC·9834·U
CA·6701·BT	CS·6737·AW	CR·5805·Z	CO·2155·AY	C·1397·CK
CU·7433·K	GI·8826·BT	GR·1874·AZ	GU·0333·J	SS·6261·BK
H·4575·AB	HU·9747·P	J·4209·AG	LE·8797·AJ	L·4069·AJ
LO·1001·V	LU·6360·X	M·6814·ZX	MA·8932·DF	MU·9921·CK
NA·7541·BD	OU·8069·X	O·0610·CK	P·0849·L	GC·7889·CM
PO·0126·BU	SA·8295·V	TF·1308·CD	S·6756·AS	SG·4434·J
SE·7129·DW	SO·3367·G	T·1425·BG	TE·7164·I	T0·6753·AG
V·1257·HJ	VA·8654·AL	BI·9894·CV	ZA·1665·L	Z·5428·BT
			CE·1131·H	ML·1931·F

Negativos

Imágenes de tamaño N x M
extraídas aleatoriamente



Extractor de características (Proyecciones Horizontales y verticales)



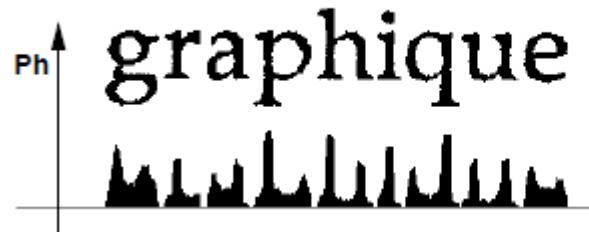
Conjunto de datos de entrenamiento



Clasificador
entrenado

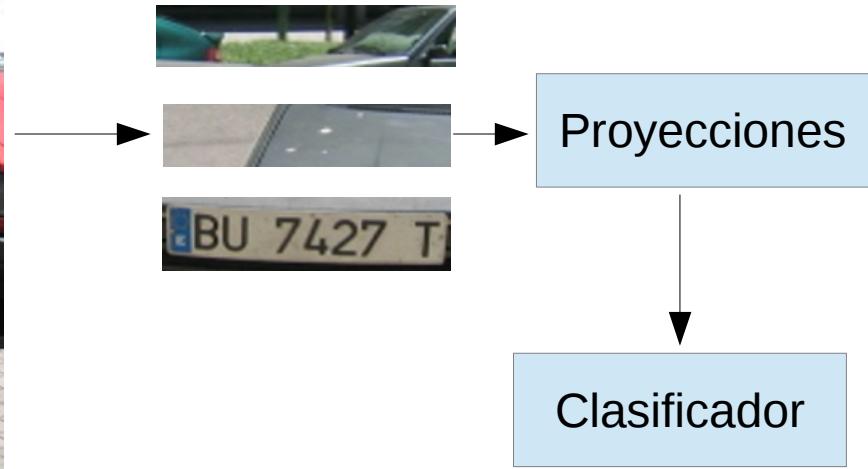
Casos de Uso

- Un clasificador trabaja con vectores de características, generalmente vectores de números. Una técnica muy usada en el reconocimiento de caracteres son las proyecciones horizontales y verticales.
 - Por cada columna, sumamos el número de píxeles en negro.
 - Por cada fila, lo mismo.



Casos de Uso

Se evalúa cada una de las posibles “ventanas de la imagen”, haciendo pasar esa ventana por el clasificador entrenado previamente.



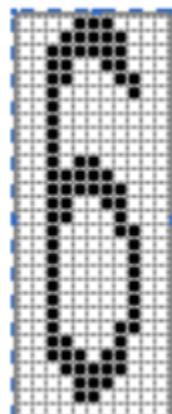
Casos de Uso

Se combinan todos los positivos del clasificador y se descartan los positivos aislados.

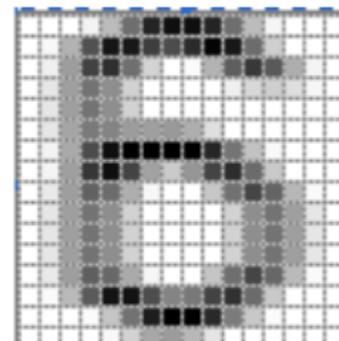


Casos de Uso

- Reconocimiento de matrículas.
 - La clase es el carácter que representa cada imagen.
 - Los atributos es un "downsampling" de la imagen. La imagen se reescala a 16x16.
 - 256 atributos. 0 si no hay carácter 255 si lo hay.
 - Ej 0,0,0,0,1,1,1,0 , Clase6



=>

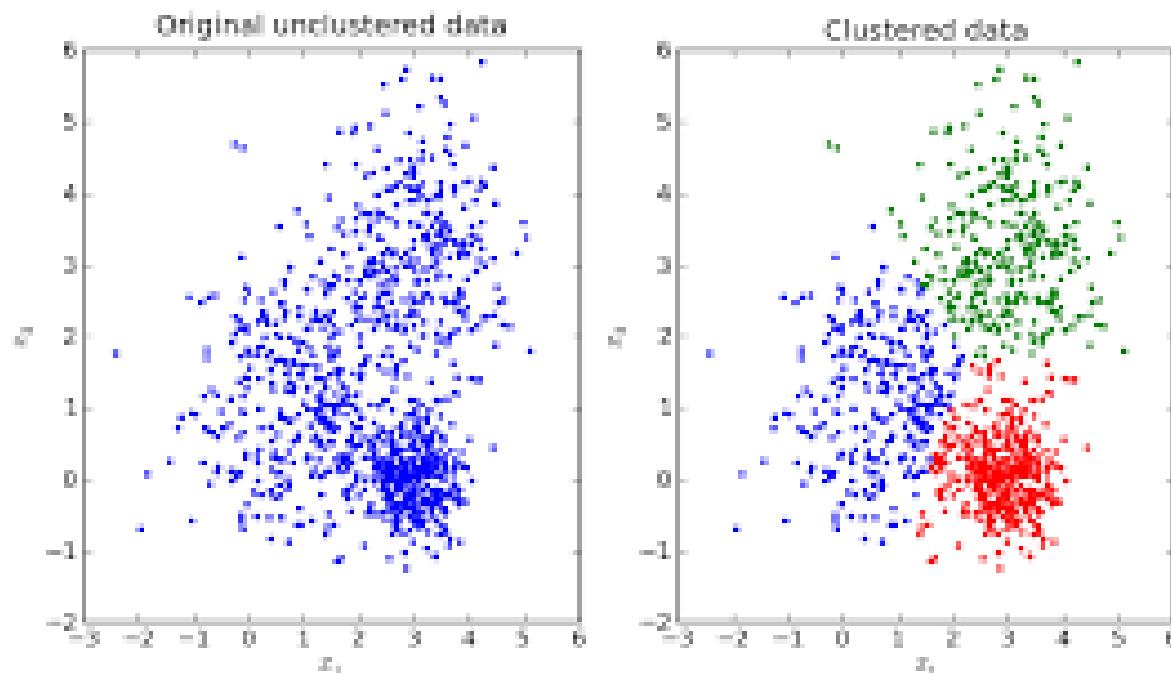


Casos de Uso

- Reconocimiento de matrículas.
 - Abrimos el MatriculasMatrizBinaria.arff.
 - Probamos con vecinos más cercanos o con un árbol de decisión.

Clustering

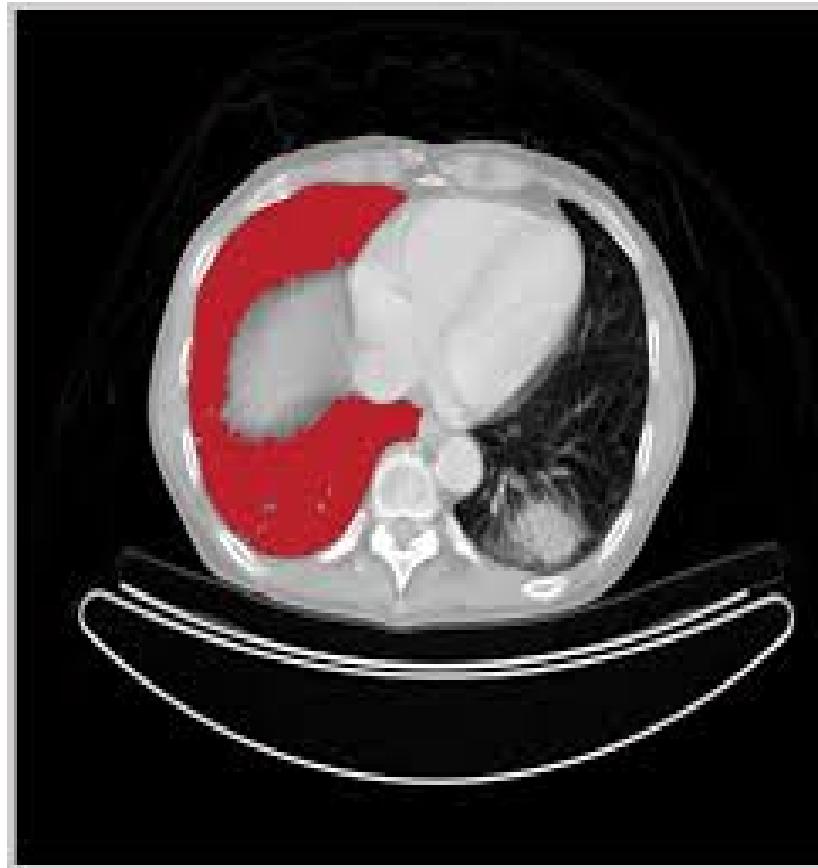
- El clustering o agrupación no tiene en cuenta la clase, ya que se desconoce. Consiste en agrupar los ejemplos en grupos que tengan características parecidas.



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).

Clustering



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>

Casos de Uso

- Reconocimiento de setas. Proyecto final de máster.

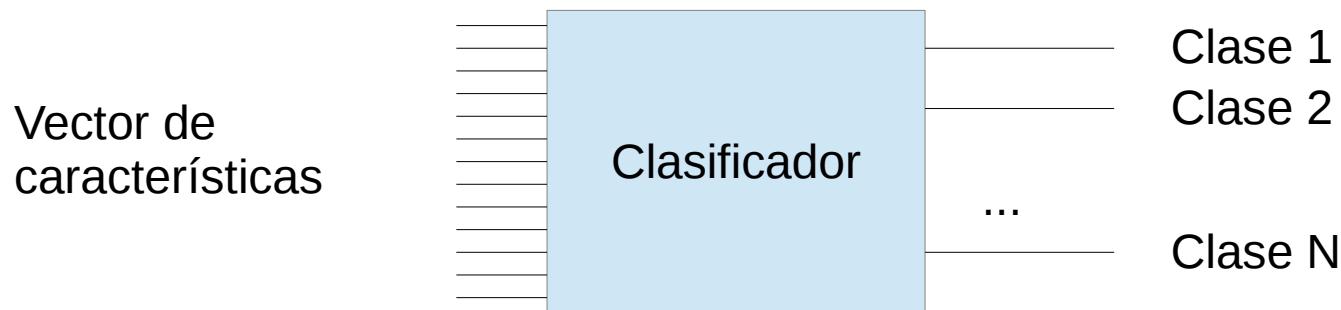
Tutores: José Francisco Díez y Raúl Marticorena Sanchez

Alumno: Iñaki Arroyo Nebreda.

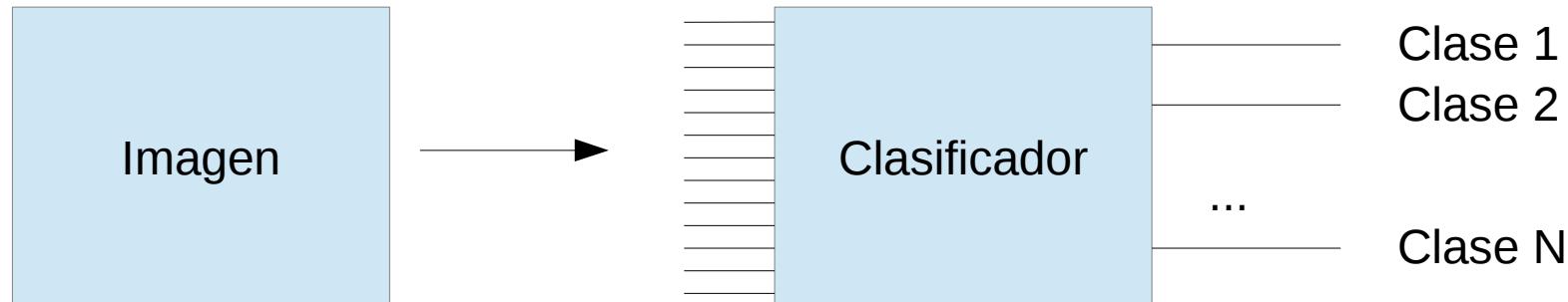
- Se procesan las imágenes con el algoritmo *bag of words*, basado en k-means para obtener un conjunto de datos a partir de un conjunto de imágenes.

Casos de Uso

- El reconocimiento de objetos es el problema de clasificar objetos en un conjunto de categorías definidas.



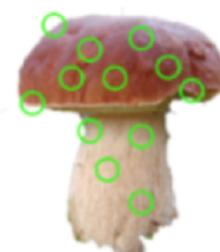
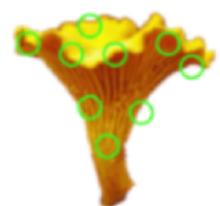
- El mayor problema en el reconocimiento de objetos es extraer las características con las que poder trabajar.



Casos de Uso

Imágenes
segmentadas

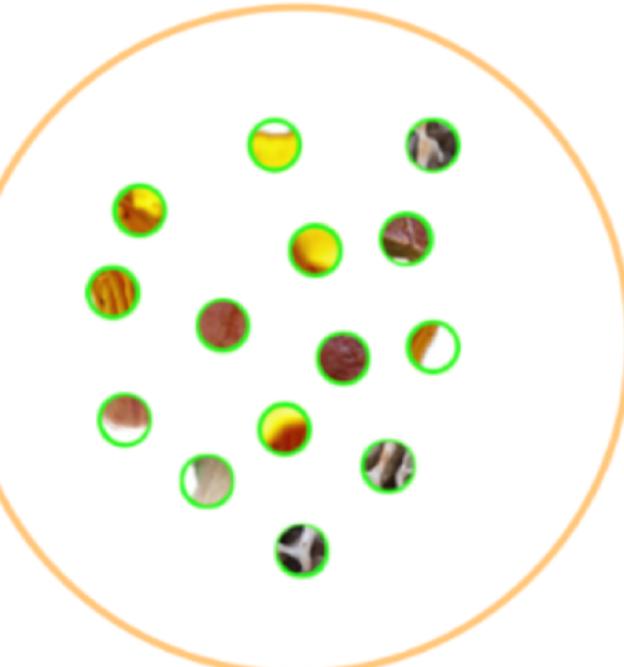
Colección de
palabras clave



SURF

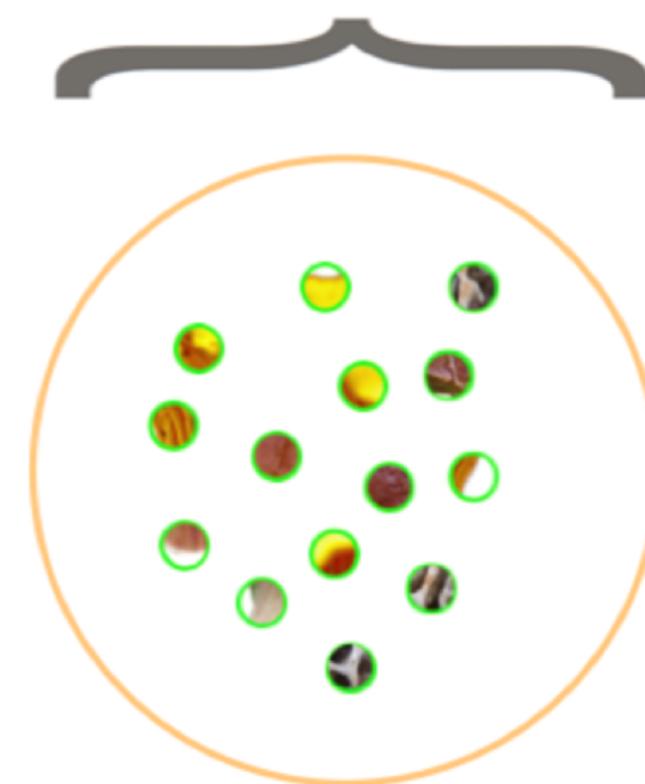
SURF

SURF

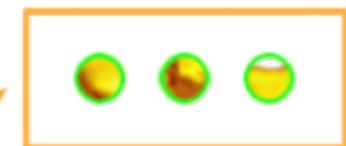


Casos de Uso

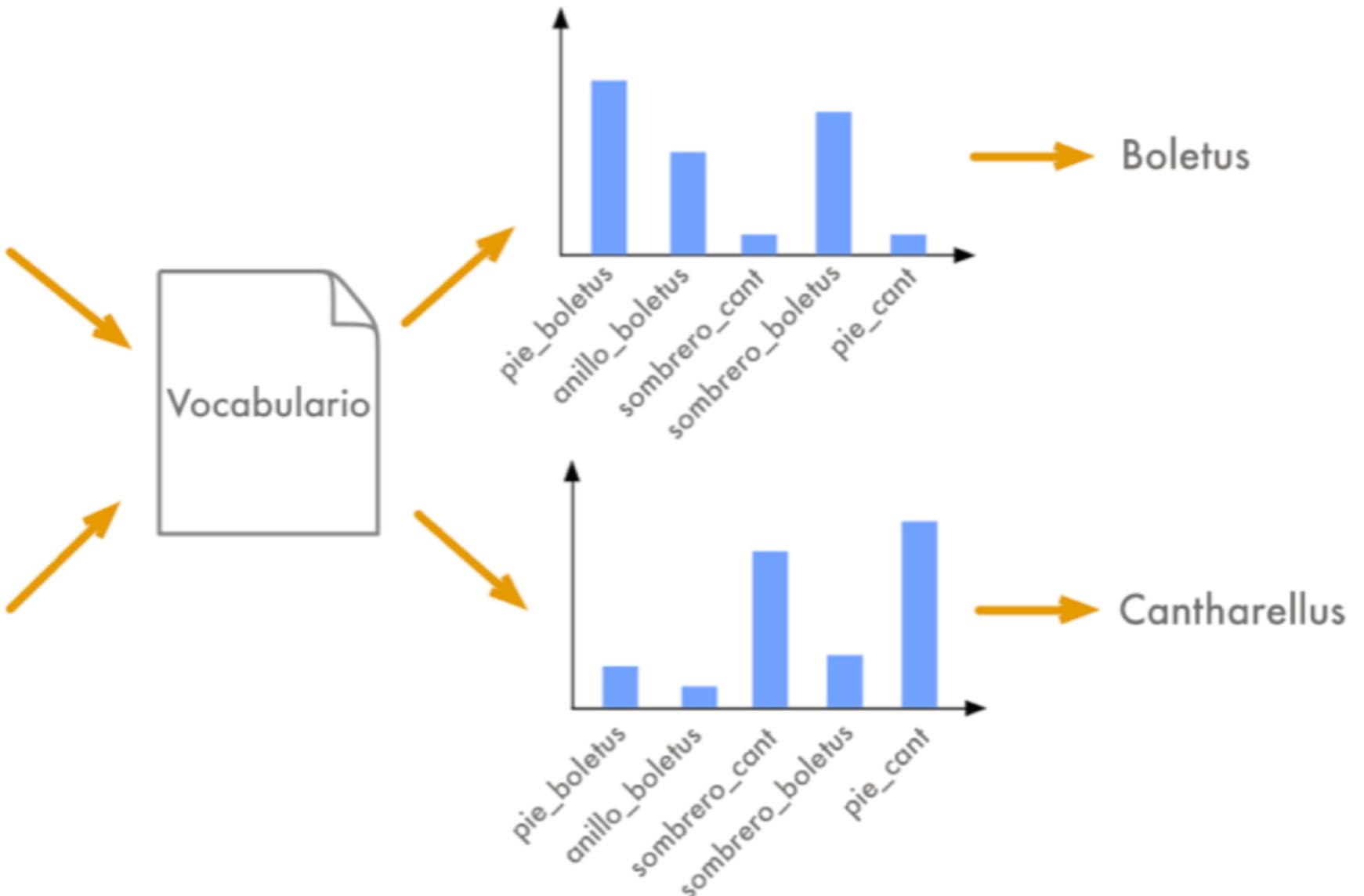
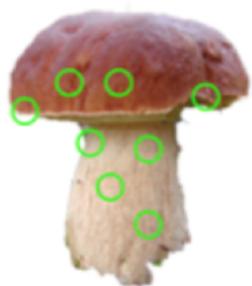
Conjunto de
palabras clave



Vocabulario



Casos de Uso



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.
 - Eje: 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.
 - Eje. 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

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.



Deep Learning

Features for machine learning

Images



Image



Vision features

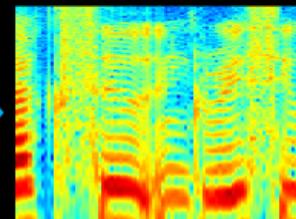


Detection

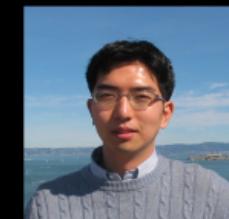
Audio



Audio



Audio features

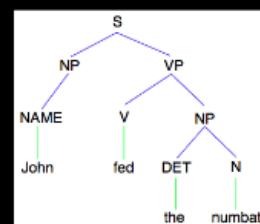


Speaker ID

Text



Text



Text features

Web search

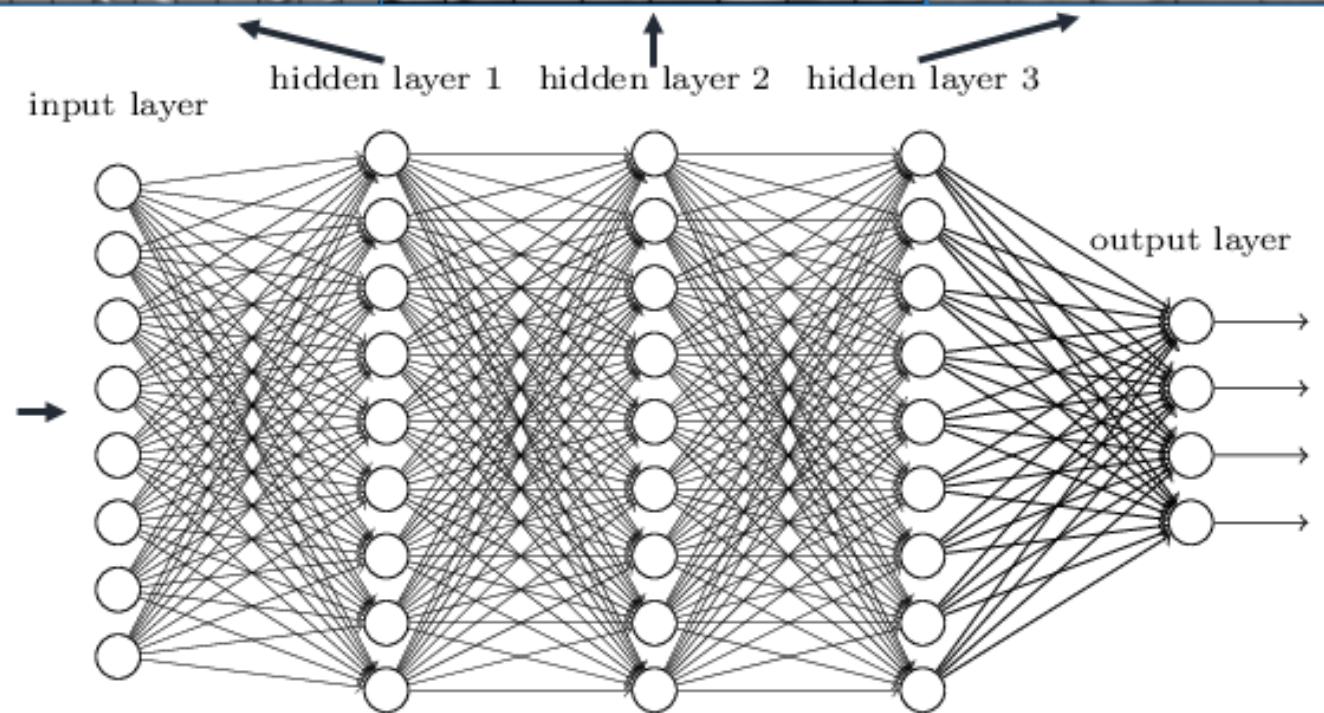
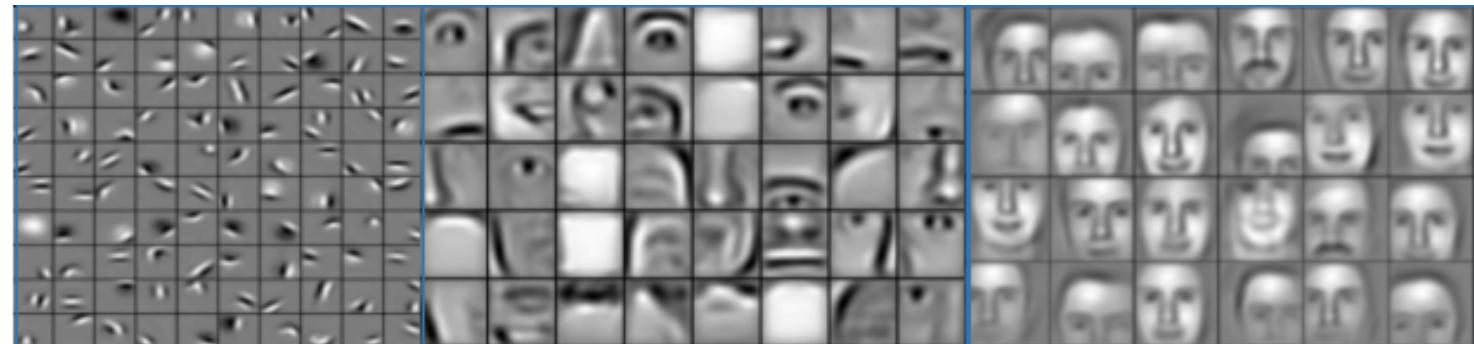
...

Deep Learning

- Trabajar con datos que no están estructurados en atributos-clase es muy complicado.
- Imágenes, video, audio, textos tienen cada uno distintas técnicas para ser procesados.
- Pero el cerebro humano procesa todo utilizando un solo algoritmo (En un ciego el cortex auditivo "aprende a ver").
- Solución: construir algoritmos de aprendizaje que imitan el cerebro.

Deep Learning

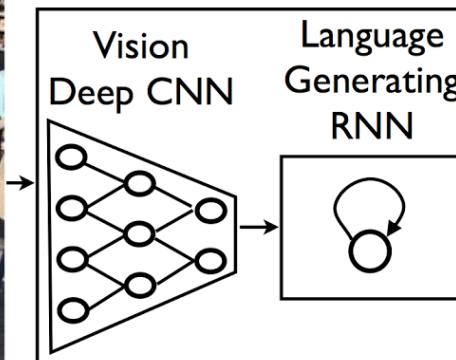
Deep neural networks learn hierarchical feature representations



Caso de Uso

- Reconocimiento de escenas. Proyecto Final de Grado.

Tutores: José Francisco Díez y César García Osorio
Autor Bryan Reinoso.



A group of people shopping at an outdoor market.
There are many vegetables at the fruit stand.

Caso de Uso

- Reconocimiento de objetos. Proyecto de bachillerato de excelencia.

Tutores: José Francisco Diez y Miguel Angel Conde

Autora: Esperanza Montes.

Caso de Uso

- Composición/Armonización de canciones.



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

Caso de Uso

- Creaciones artísticas



- Acceder:

<https://deeplearning4j.org/deepdreamgenerator>

Comentarios finales



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 NewsService | 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.