

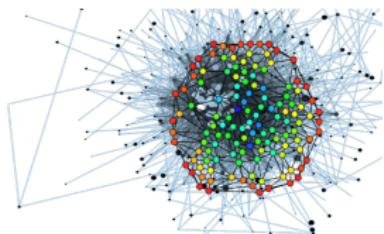
Introducción a la ciencia de datos y el aprendizaje automático (parte 2)

Javier Sánchez-Monedero

Departamento de Métodos Cuantitativos
Universidad Loyola Andalucía
2 de noviembre de 2017



Índice



Modelos de aprendizaje automático para clasificación

Tipos de clasificación

Modelos populares de clasificación

Cuestionario y ejercicios

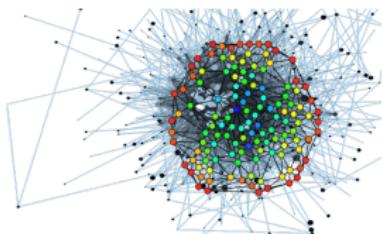
Big Data

Deep Learning y AI

Conclusiones



Índice



Modelos de aprendizaje automático para clasificación

Tipos de clasificación

Modelos populares de clasificación

Cuestionario y ejercicios

Big Data

Deep Learning y AI

Conclusiones



Clasificación binaria

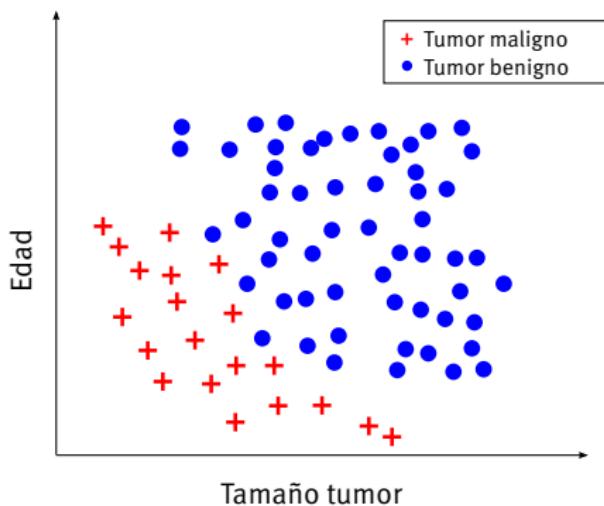


Figura: Ejemplo de problema de **clasificación** ¿Podrías estimar un diagnóstico basado en el tamaño del tumor y la edad del paciente?

Clasificación multi-clase

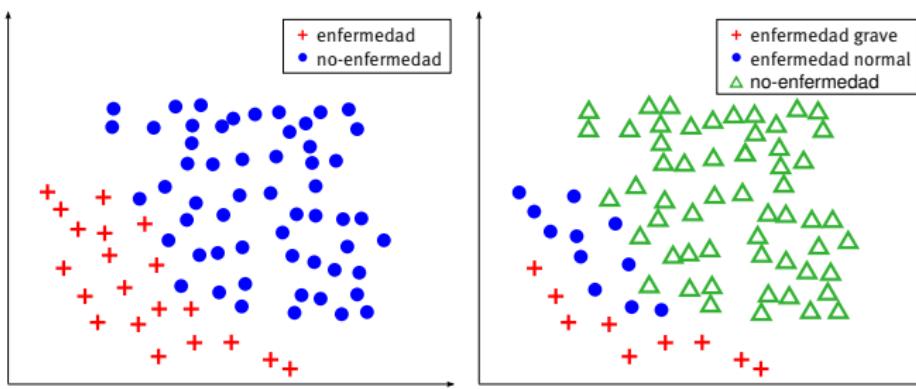


Figura: Un ejemplo de **clasificación binaria** (figura a la izquierda) frente a la **clasificación multi-clase** (figura de la derecha). En el primer caso hay dos estados para un patrón: enfermo o no enfermo. Sin embargo, un experto que se apoye en técnicas de aprendizaje automático puede demandar grados de clasificación más finos, en cuyo caso podría afrontarse el problema como clasificación multi-clase.

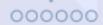
Otros tipos de clasificación

Además de la clasificación binaria y multi-clase, existen otros tipos:

- **Clasificación ordinal** (también llamada regresión ordinal).
- Clasificación **multi-etiqueta**.
- Clasificación **semi-supervisada**.

Formulación matemática

- Disponemos de un **espacio de entrada** \mathcal{X} compuesto por patrones etiquetados con $\mathcal{C} = \{C_1, C_2, \dots, C_Q\}$ donde Q es el número de clases.
- Cada **patrón** se representa un por **vector de características** de dimensión K , $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^K$ y una etiqueta de clase $y \in \mathcal{C}$.
- El objetivo es aprender una función ϕ que relacione los datos del espacio de entrada \mathcal{X} al conjunto finito \mathcal{C} .
- El conjunto de patrones de entrenamiento \mathbf{T} está compuesto de N patrones
$$\mathbf{T} = \{(\mathbf{X}, \mathbf{Y}) = (\mathbf{x}_i, y_i) : \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{C} (i = 1, \dots, N)\},$$
 con
$$\mathbf{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,K}).$$



Modelo linear vs no-linear

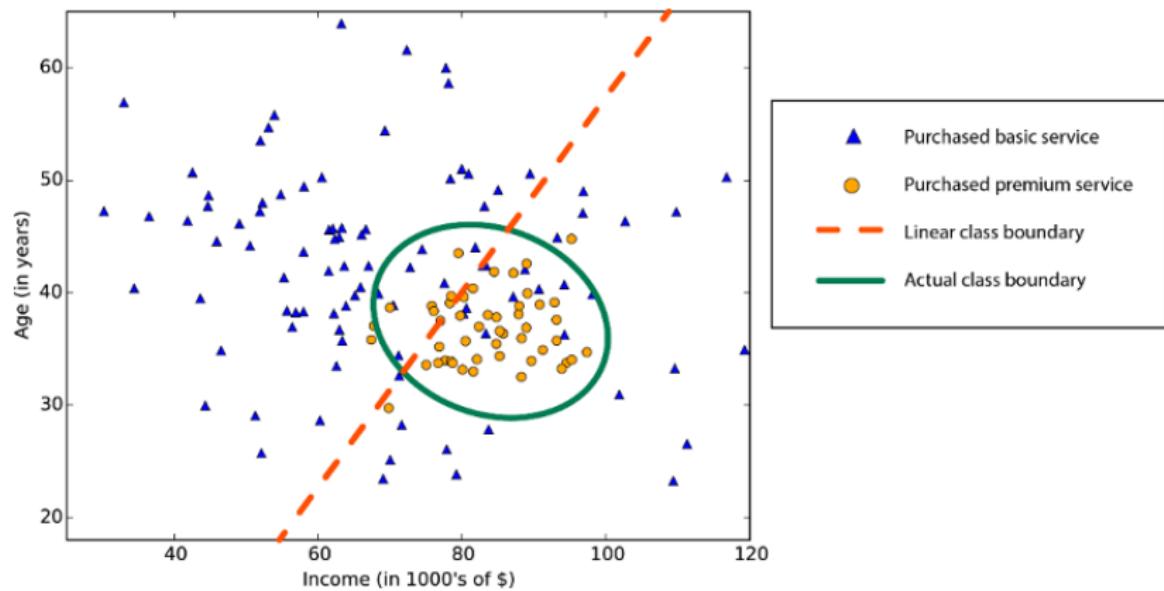


Figura: Fuente [How to choose algorithms for Microsoft Azure Machine Learning](#)

Regresión logística

La **regresión logística** es un **método estadístico**, aunque muchos paquetes de aprendizaje automático la incluyen con variantes. En un **modelo lineal** de clasificación binaria

$$p_i = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_{1,i} + \dots + \beta_K x_{K,i})}} \quad (1)$$

- **Ventajas:** probabilístico, no tiene hiper-parámetros y es interpretable.
 - **Desventajas:** modelo lineal.



k-vecinos cercanos

El método ***k*-vecinos cercanos** o *k*-NN (*K nearest neighbors*) es uno de los más sencillos. Se basa en estimar la probabilidad de pertenencia de un patrón x a una clase $F(x/C_j)$ en base a los k patrones más cercanos que le rodean.

- **Ventajas:** probabilístico, no lineal, tiene variantes que mejoran el rendimiento y robustez a patrones anómalos.
- **Desventajas:** altamente dependiente de k , no interpretable.

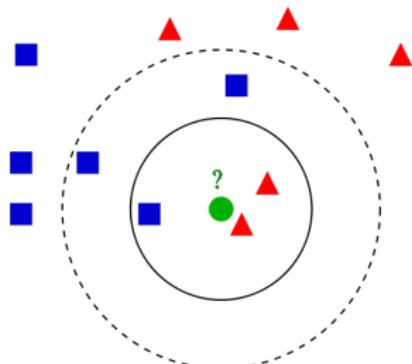
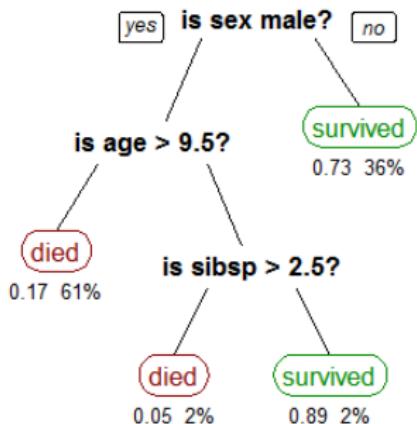


Figura: Fuente <https://commons.wikimedia.org/wiki/File:KnnClassification.svg>

Árboles de decisión I

El algoritmo de aprendizaje construye un **árbol de decisión** donde las hojas son las categorías y las ramas representan valores de características o conjuntos de características. C4.5 es la implementación más extendida. Ej. probabilidad de sobrevivir al accidente del Titanic.



Fuente

https://commons.wikimedia.org/wiki/File:CART_tree_titanic_survivors.png

Árboles de decisión II

- **Ventajas:** probabilístico, no lineal, interpretable, no sensible a hiper-parámetros
- **Desventajas:** puede sobreentrenar, pueden ser muy complejos y perder Interpretabilidad.

Clasificador bayesiano ingenuo I

Clasificador bayesiano ingenuo o **Naive Bayes** es un clasificador probabilístico fundamentado en el **teorema de Bayes** y algunas hipótesis simplificadoras adicionales (ingenuo o **naive**).

- Asume que la presencia o ausencia de una característica particular no está relacionada con la presencia o ausencia de cualquier otra característica, dada la clase variable.
- En otras palabras, los valores de las variables dependientes no están influidos por el resto de variables.
- Esta simplificación facilita enormemente el entrenamiento.
- Son muy eficientes en algunos problemas. Por ejemplo muchos filtros de correo basura utilizan un clasificador naive Bayes.



Clasificador bayesiano ingenuo II

Es un modelo de probabilidad condicionada, donde la probabilidad de pertenencia a la clase C_k del patrón $\mathbf{x} = (x_1, \dots, x_n)$ es:

$$p(C_k | x_1, \dots, x_n) \quad (2)$$

Que reformulado bajo el teorema de Bayes:

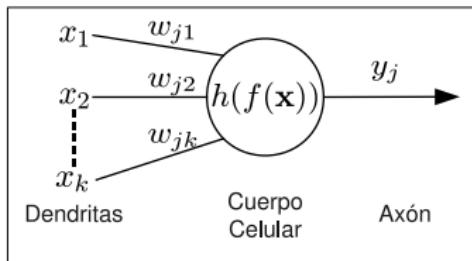
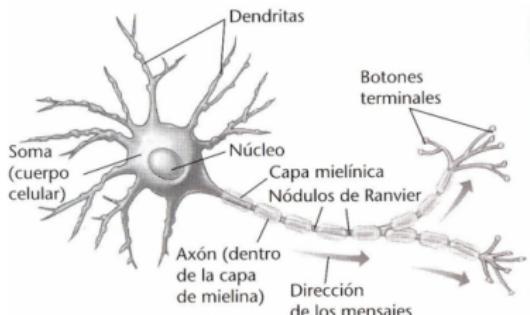
$$p(C_k | \mathbf{x}) = \frac{p(C_k) p(\mathbf{x} | C_k)}{p(\mathbf{x})} \quad (3)$$

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}} \quad (4)$$

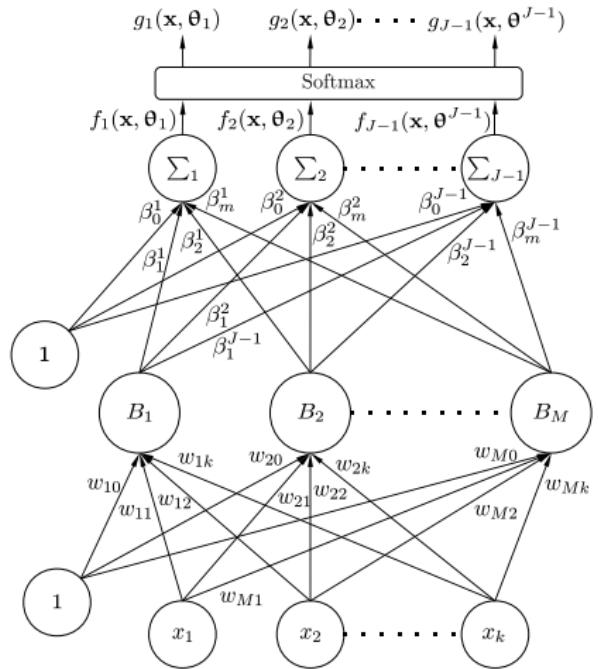


Concepto de Red Neuronal Artificial

- Técnica de modelado fundamentada en la **emulación de los sistemas nerviosos biológicos**.



- Combina una gran cantidad de elementos simples de procesado (**neuronas**), altamente interconectados y agrupados en **capas**.
 - Una red neuronal es una **relación funcional matemática** entre unas variables de entrada y unas variables de salida.



Para *J* clases,

$$f_j(\mathbf{x}, \theta_j) = \beta_0^j + \sum_{i=1}^M \beta_i^j B_i(\mathbf{x}, \mathbf{w}_i)$$

para $1 \leq j \leq J - 1$

$$g_j(\mathbf{x}, \theta_j) = \frac{e^{f_j(\mathbf{x}, \theta_j)}}{\sum_{i=1}^m e^{f_i(\mathbf{x}, \theta_i)}}$$



Características de las RNA I

- Las RNA son **aproximadores universales**, es decir, pueden aprender cualquier función matemática aumentando su capacidad, que depende de la complejidad (número de capas y número de conexiones).
- Hay **muchos tipos de arquitecturas** de RNA, aquí sólo hemos mostrado las redes *single layer feedforward*.
- Otros tipos de RNA: redes recurrentes, redes convolucionales, etc.
- Aunque nunca han perdido uso, recientemente vuelven a la actualidad científica sobre todo en el campo del **aprendizaje profundo** (*deep learning*). Por ejemplo, el motor de búsqueda de imágenes de Google funciona con RNA para aprender conceptos etiquetados en imágenes¹.

Características de las RNA II

- **Ventajas:** probabilístico, no lineal, aproximador universal, versatilidad (reconocimiento del habla, textos...)
- **Desventajas:** a menudo sobreentrenan, el coste de entrenar RNA es alto con los métodos más populares, no son interpretables

¹Inceptionism: Going Deeper into Neural Networks



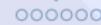
Máquinas de vector soporte

La idea básica de las **máquinas de vector soporte** (SVM, *support vector machines*) es simple: encontrar el **hiper-plano que define la máxima separación entre patrones de dos clases diferentes**:

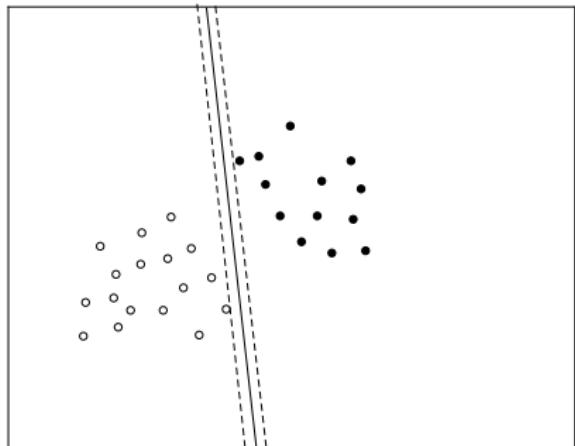
$$f(\mathbf{x}) = \hat{y} = \text{sgn} (\langle \mathbf{w} \cdot \phi(\mathbf{x}) \rangle + b),$$

donde $\hat{y} = +1$ si \mathbf{x} corresponde a la clase positiva y $\hat{y} = -1$ en otro caso.

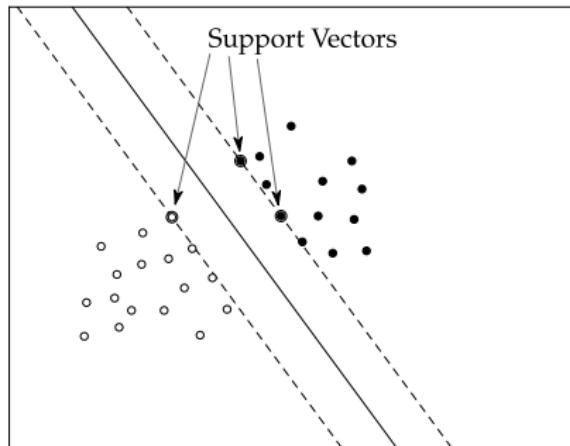
El modelo correspondiente para el caso no lineal y de margen blando es más complejo, y comprender el proceso de cálculo del hiper-plano óptimo requiere de habilidades matemáticas en optimización.



Máximo hiper-plano separador

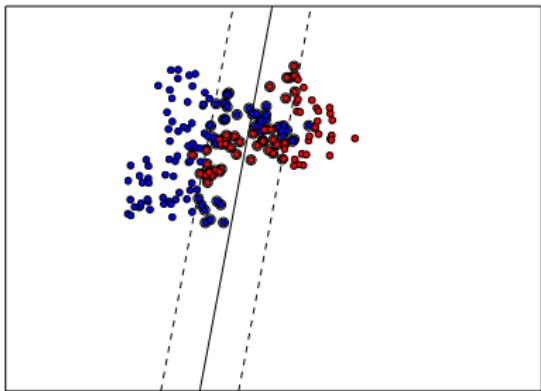
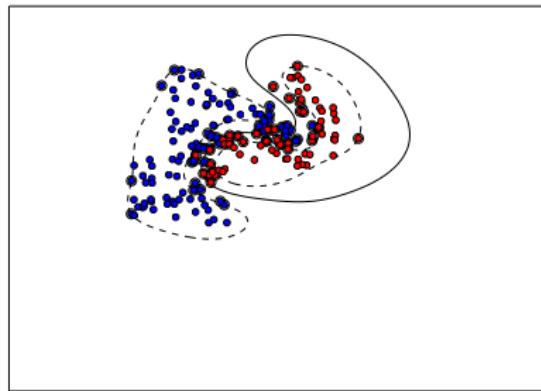


Small margin



Large margin

Ejemplo de SVM lineal y no lineal

Linear: $u^T v$ RBF: $\exp(-\gamma|u-v|^2)$ Poly: $(\gamma u^T v + r)^d$ Linear: $u^T v$ RBF: $\exp(-\gamma|u-v|^2)$ Poly: $(\gamma u^T v + r)^d$



Características de los SVM I

- Para **pocos patrones** ($< 2,000 - 10,000$ dependiendo de la dimensionalidad) los modelos no lineales son muy potentes, previo ajuste de los hiper-parámetros de coste y ancho del kernel gausiano.
- Para bases de datos mayores, el modelo lineal es bastante competitivo (implementación liblinear).
- **Ventajas:** no lineal, más robusto al problema de la alta dimensionalidad, *sparsity*, etc.
- **Desventajas:** a menudo sobre-entrenan si no se ajustan bien los hiper-parámetros, no son interpretables, ineficientes para entrenar grandes volúmenes²

²Si bien hay propuestas para la resolución distribuida del problema de optimización



SVM en Weka

- La implementación de SVM que viene por defecto en Weka (SMO) no es muy adecuada.
- La mejor implementación de SVM es LibSVM³, que se puede utilizar dentro de Weka, además de poder usarse dentro de muchos entornos de programación científica.
- Para instalar LibSVM en Weka debemos utilizar el gestor de paquetes.

³ <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>



Cuestionario

¿Qué parámetro controla la complejidad del modelo en una red neuronal artificial? (puede haber más de una respuesta)

1. El número de neuronas en la capa o capas ocultas
2. Las funciones que procesan los datos en cada neurona
3. El número de capas de la red.

Cuestionario

En una red neuronal artificial, ¿qué determina el número de neuronas de entrada?:

1. La complejidad del problema.
2. El número de clases del problema.
3. El número de variables del problema.

¿Qué principal desventaja tiene el algoritmo k-NN?

1. Es un modelo lineal.
2. Su rendimiento depende del correcto ajuste del parámetro k .



Algoritmo k-NN (IBk en Weka):

- En el entorno *Explorer* utiliza el algoritmo IBk con una validación cruzada 7-fold. Visualiza la clasificación realizada (gráfica y matriz de confusión). Prueba 3 valores distintos de k . Recuerda que k tiene que ser un número impar. Incluye ambas en la memoria.

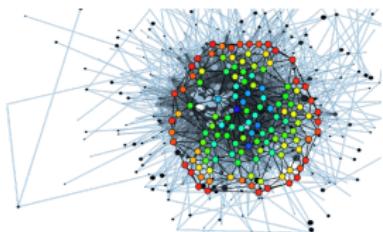
Ejercicio

Algoritmo *Logistic* y *SimpleLogistic* en Weka:

- En el entorno *Explorer* utiliza los algoritmos de clasificación *Logistic* y *SimpleLogistic* con alguna de las bases de datos de ejemplo de Weka. Utiliza un 5-fold y analiza los modelos obtenidos (variables más influyentes, variables descartadas, etc.).



Índice



Modelos de aprendizaje automático para clasificación

Tipos de clasificación

Modelos populares de clasificación

Cuestionario y ejercicios

Big Data

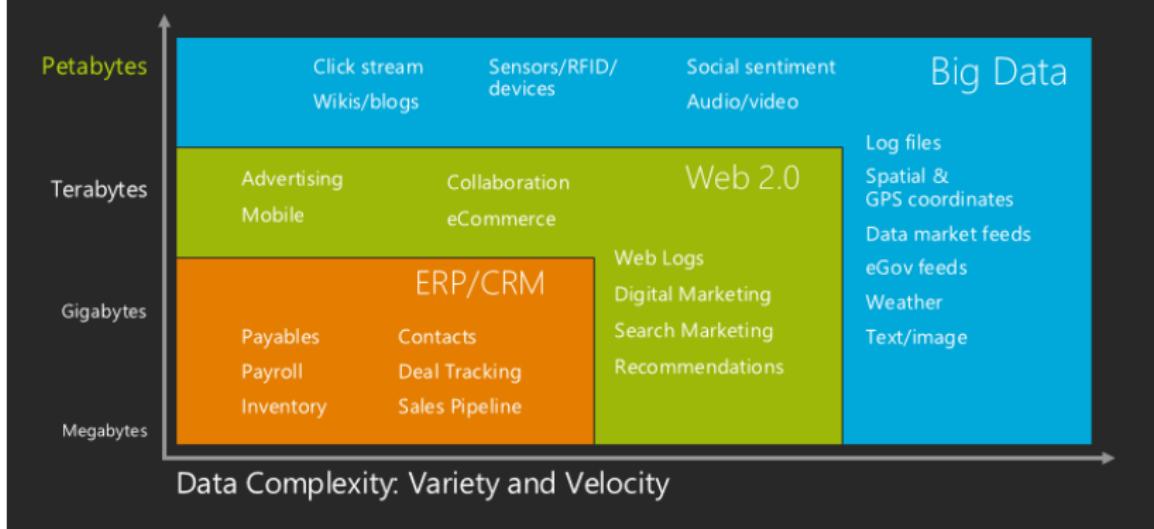
Deep Learning y AI

Conclusiones



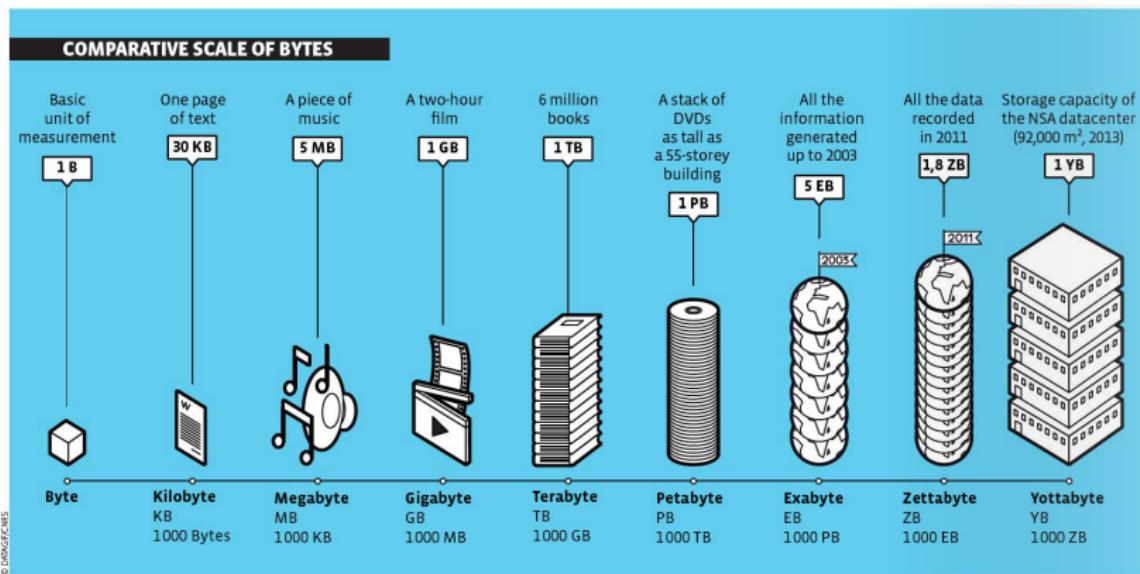
¿Qué es big data?

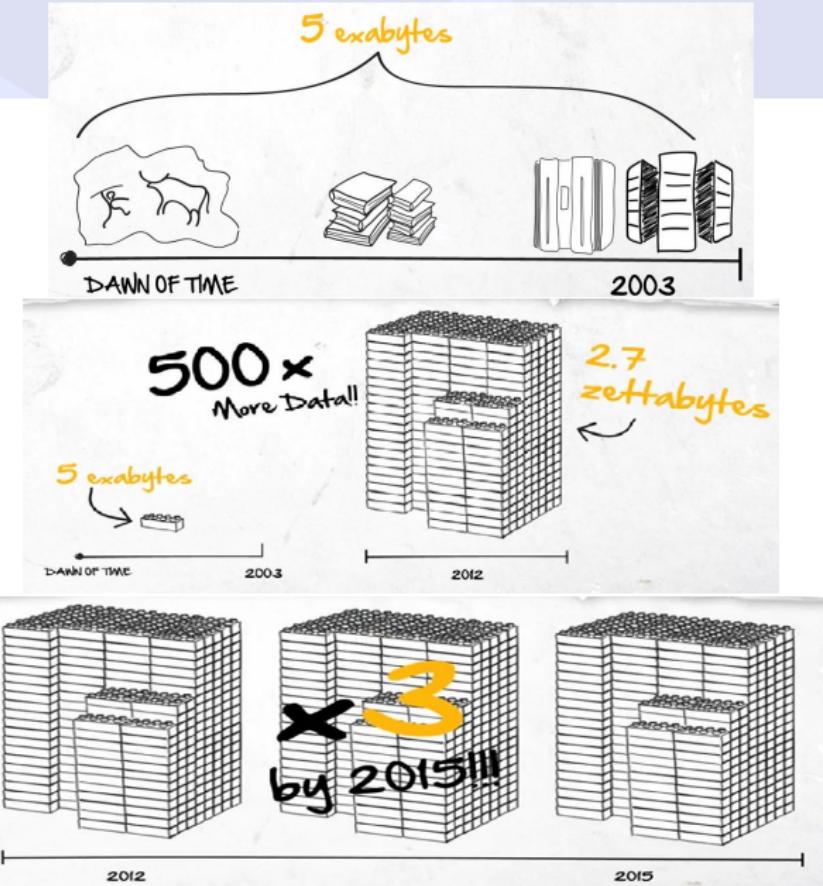
WHAT IS BIG DATA?





Medidas de volumen de la información





Crecimiento en la generación y captación de datos



- Problemas que supone el crecimiento exponencial de los datos:
 - ▶ Desconocimiento acerca de **dónde** y **cómo** adquirir los datos.
 - ▶ Falta de **formación** y capacitación para su procesamiento y para extraer valor.
 - ▶ **Presupuesto** ⇒ Falta de dinero para hacer inversiones.
 - ▶ La información llega tarde, después de los resultados ⇒ necesidad de las empresas por llevar a cabo una analítica empresarial **a tiempo real** que permita tomar las decisiones al instante.
 - ▶ **Seguridad** y privacidad.

¿Por qué big data?

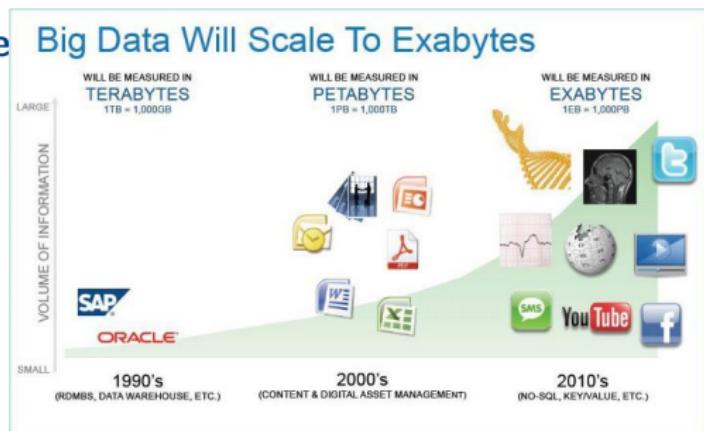


Bill Gates supuestamente dijo en 1981: “**640K ought to be enough for anybody.**”



¿Qué es big data?

- No hay una definición estándar.
- *Big data* es una colección de datos grande, complejos, **muy difícil de procesar a través de herramientas de gestión y procesamiento de datos tradicionales.**
- *Big data* son datos cuyo volumen, diversidad y complejidad requieren **nueva arquitectura, técnicas, algoritmos y análisis** para gestionar y **extraer valor y conocimiento oculto** en ellos ...





Las tres Vs del big data



Netflix: el origen de Kaggle



- Competición de Netflix (2006-2011) para obtener mejores sistemas de recomendación de películas y contenido multimedia.
- El premio fue de **1M\$**.
- Supuso un gran avance para los **sistemas de recomendación**.





Netflix: la solución no se implantó

La solución diseñada por el equipo ganador del **1M\$ no llegó a ser puesta en producción** por dos motivos:

- **Eficiencia**: la mejora en la precisión no justificaba el coste computacional y la dificultad de los ingenieros para poner en producción los algoritmos de aprendizaje automático.
- El **modelo de negocio** de Netflix pasaba del alquiler de DVDs a la emisión por Internet, lo que generaba datos muy distintos a los usados en el concurso.

<http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html>



MapReduce: motivación



- ¿Cómo almacenamos los datos para realizar la computación?
 - ▶ **No tenemos suficiente RAM** para almacenar los datos.
 - ▶ El acceso a disco es lento.
- Solución: divide y vencerás.

No muevas los datos al computador, mueve la computación a los datos.

MapReduce: motivación

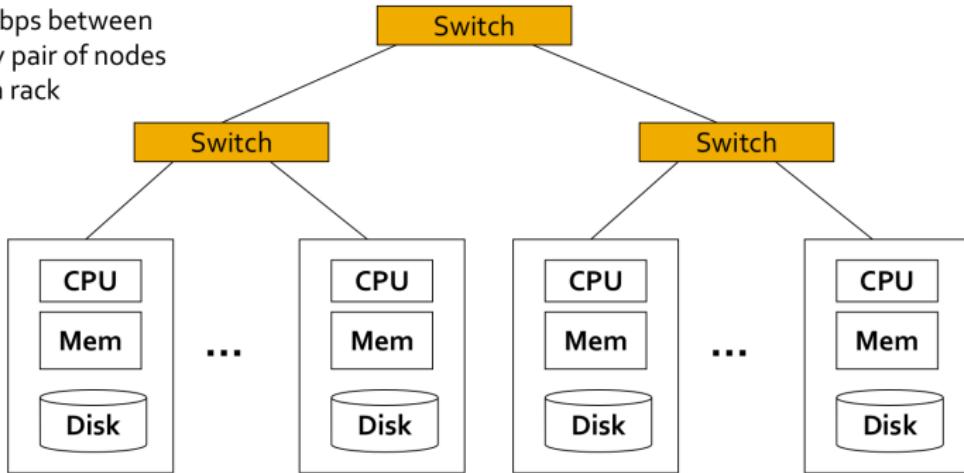
- **Programación distribuida:** retos.
 - ▶ *¿Cómo* distribuimos la computación?.
 - ▶ La computación paralela/distribuida es **engorrosa**.
- Google:
 - ▶ +20 billones de páginas web \times 20KB = **+400TB**
 - ▶ Un ordenador lee 30 – 35MB/seg del disco \Rightarrow **4 meses** para leer la web.
 - ▶ **≈1.000 discos duros** para almacenarla.
- Palabra clave: *commodity hardware* \Rightarrow *hardware asequible*.



MapReduce: motivación

2-10 Gbps backbone between racks

1 Gbps between
any pair of nodes
in a rack



Google tiene 2.376.640 servidores (2013, estimado)

<https://plus.google.com/+JamesPearn/posts/VaQu9sNxJuY>

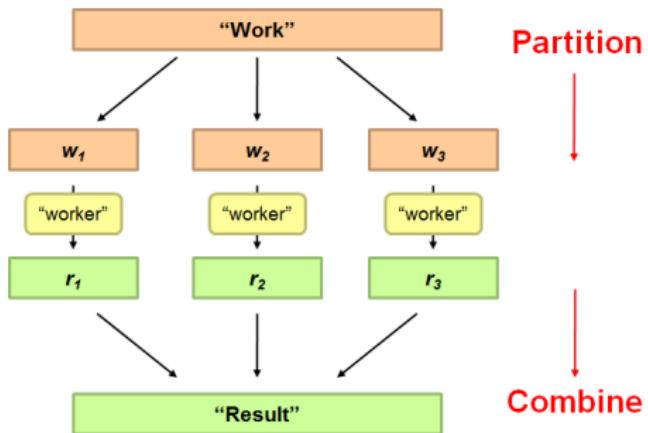
MapReduce: motivación





MapReduce: motivación

- Divide y vencerás:



MapReduce: motivación

- Divide y vencerás (retos de paralelización):
 - ▶ ¿Cómo asignamos unidades de trabajo a los **workers (trabajadores)**?
 - ▶ ¿Qué pasa si tenemos **más trabajos que trabajadores**?
 - ▶ ¿Qué pasa si los trabajadores tienen que compartir **resultados parciales**?
 - ▶ ¿Cómo agregamos **resultados parciales**?
 - ▶ ¿Cómo **sabemos** si todos los trabajadores han **terminado**?
 - ▶ ¿Qué pasan cuando **algunos trabajadores caen**?



MapReduce: concepto

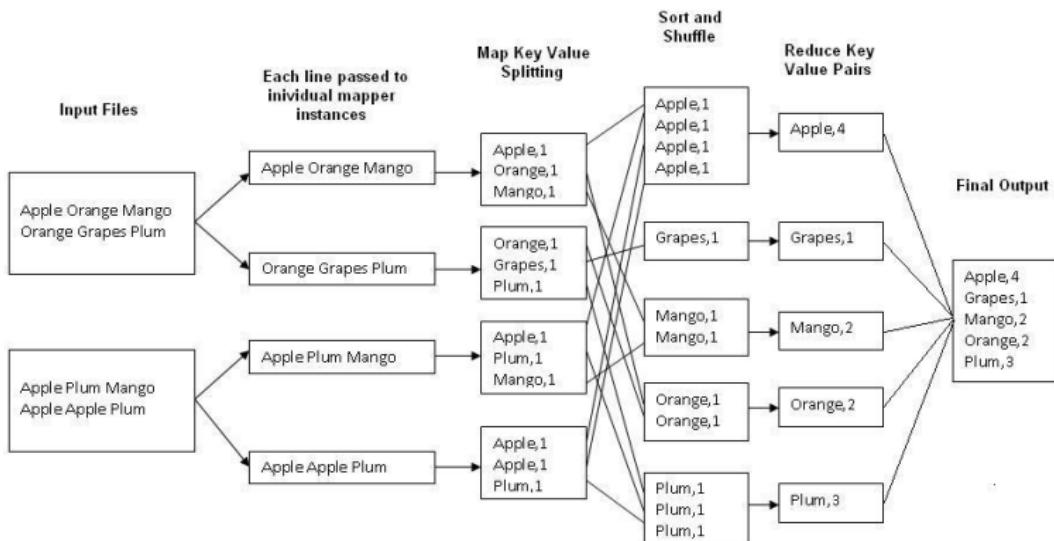
- **Tarea:** contar el número de veces que aparece cada palabra dentro de un fichero de texto muy grande.
- **Aplicación:** analizar ficheros *log* de un servidor web para encontrar los términos más populares (<http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html>).



MapReduce: concepto

- El paradigma *MapReduce* se basa en pares **<clave,valor>** y tiene tres fases:
 1. **Map**: Iterar de forma secuencial sobre un conjunto elevado de registros y extraer información de cada uno de ellos. A cada resultado intermedio asignarle una **clave**.
 2. Agrupar los resultados intermedios por claves (**shuffle**).
 3. **Reduce**: fusionar los resultados anteriores de acuerdo a su clave y generar una salida final.
- Se proporciona una abstracción funcional de las operaciones *Map* y *Reduce*, que serán las específicas de cada problema.

MapReduce: concepto





Herramientas de ciencia de datos

Generation	1 ^a Generación	2 ^a Generación
Ejemplos	KNIME, SAS, R, Weka, SPSS, KEEL	Mahout, Pentaho, Cascading
Escalabilidad	Vertical	Horizontal (over Hadoop)
Algoritmos disponibles	Huge collection of algorithms	Small subset: sequential logistic regression, linear SVMs, Stochastic Gradient Descent, k-means clustering, Random forest, etc.
Algoritmos No disponibles	Practically nothing	Vast no.: Kernel SVMs, Multivariate Logistic Regression, Conjugate Gradient Descent, ALS, etc.
Tolerancia a Fallos	Single point of failure	Most tools are FT, as they are built on top of Hadoop

Figura: Lenguajes y herramientas

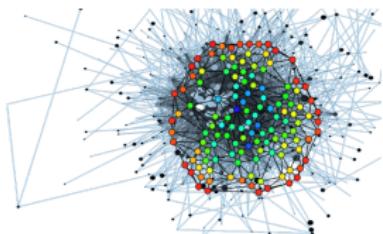


Conclusiones

- El término *Big Data* **suele confundirse** con el término ciencia de datos, analista de datos o minería de datos entre otros.
- Cuando hablamos de *Big Data*, nos referimos a **volúmenes de datos que por su tamaño (número de patrones y/o número de variables) no pueden ser procesados en una única máquina**. Además, estos datos pueden estar no estructurados al provenir de orígenes heterogéneos.
- Para hacer *Big Data* no es imprescindible usar aprendizaje automático, a veces se utilizan tecnologías de Big Data para extraer descriptores estadísticos de grandes volúmenes.
- Debido al volumen y la velocidad, se utilizan modelos lineales mayoritariamente.
- Algunos paradigmas de ensambles como *Random Forest* encajan muy bien en el paradigma *MapReduce* y aportan modelos no lineales.



Índice



Modelos de aprendizaje automático para clasificación

Tipos de clasificación

Modelos populares de clasificación

Cuestionario y ejercicios

Big Data

Deep Learning y AI

Conclusiones

Deep Learning



Maximally accurate

Maximally specific

espresso

2.23192

coffee

2.19914

beverage

1.93214

liquid

1.89367

fluid

1.85519

Aprendizaje profundo I

En general, los algoritmos de aprendizaje automático trabajan sobre variables que describen cada patrón de los datos (estatura, peso, porcentaje de píxeles rojos, velocidad, sexo...).

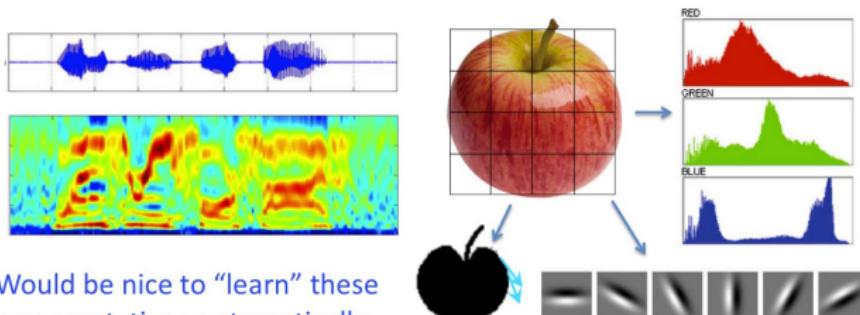
Sobre todo al procesar imágenes, vídeo, secuencias de datos (voz, texto...)... tradicionalmente ha habido expertos que realizaban *ingeniería de características* o **feature engineering**.



Ingeniería de características

Features in Machine Learning

- 1-line summary of ML: $y = \text{sgn} (\mathbf{w}^T \mathbf{x} + b)$
 - SVM can learn very effective weights \mathbf{w}
 - ... if you use the right representation \mathbf{x}



Copyright © 2014 Victor Lavrenko



Representaciones jerárquicas

Modelos compuestos y jerarquía de representaciones (concreto→abstracto)

- **Vision**: píxel, motivo, parte, objeto
 - **Texto**: carácter, palabra, frase, oración
 - **Voz**: audio, banda, fonema, palabra

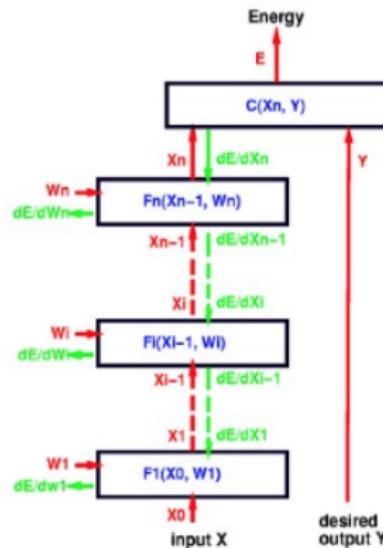


Figura: Fuente: Yann LeCun, ICML '13 tutorial



Aprendizaje end-to-end

Aprendizaje de principio a fin
(End-to-End).

- **Back-propagation:** ajusta todos los parámetros del modelo simultáneamente para optimizar el error de salida de la tarea.

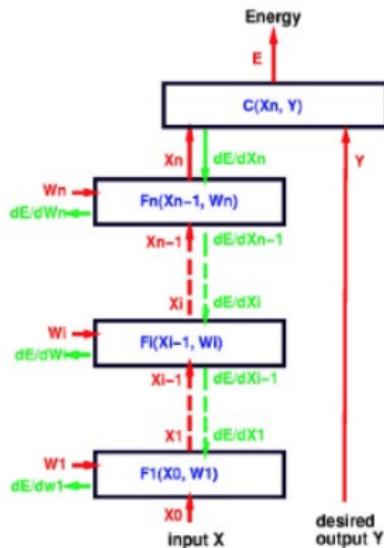
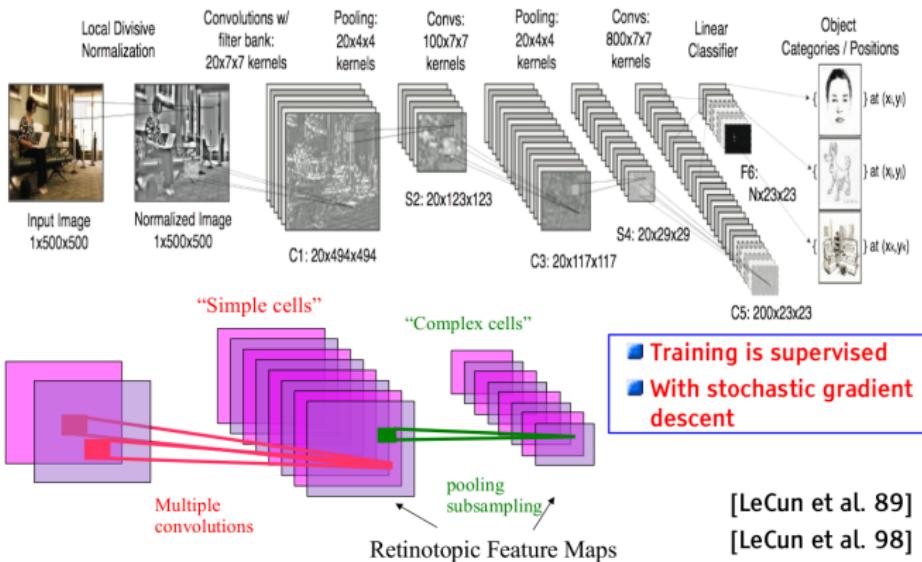


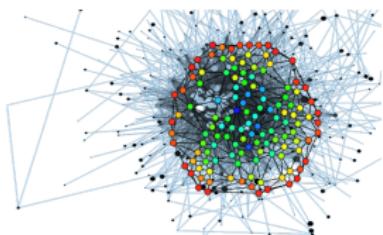
Figura: Fuente: Yann LeCun, ICML '13 tutorial

Redes convolucionales profundas





Índice



Modelos de aprendizaje automático para clasificación

Tipos de clasificación

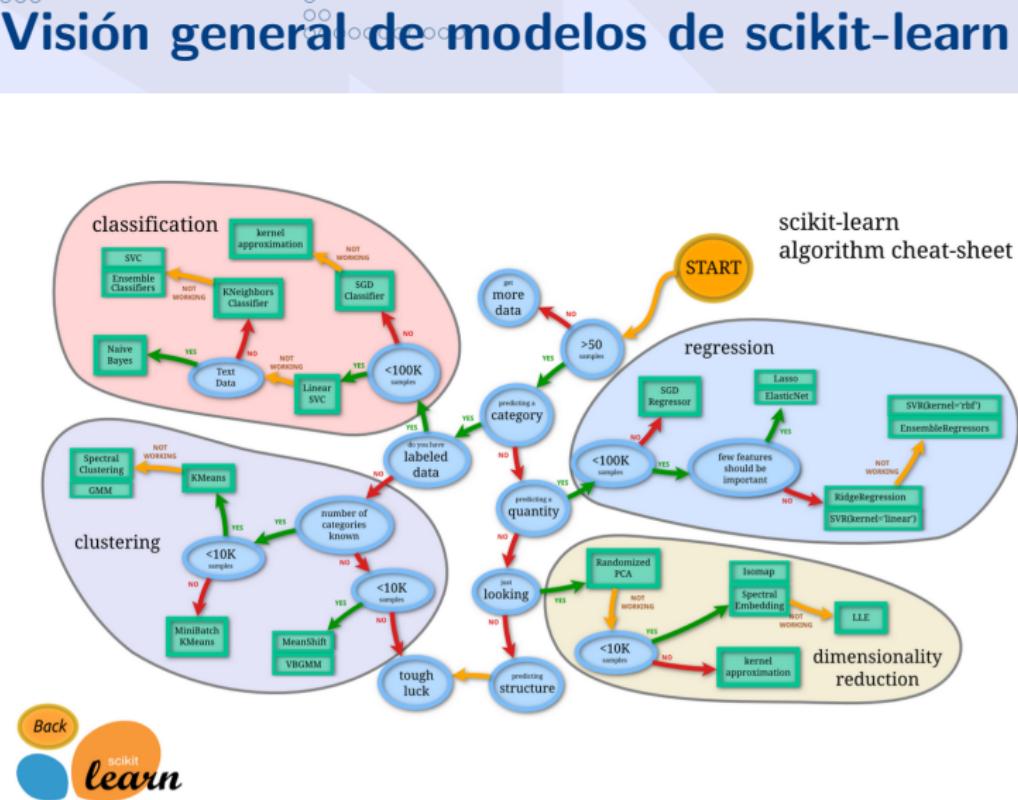
Modelos populares de clasificación

Cuestionario y ejercicios

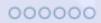
Big Data

Deep Learning y AI

Conclusiones



scikit-learn.org: Choosing the right estimator



Conclusiones ciencia de datos

- Introducción a la ciencia de datos, minería de datos y aprendizaje automático.
- Vista general de técnicas de minería de datos.
- Principales modelos de clasificación, aunque faltan modelos, como los grafos probabilísticos.
- Hemos identificado algunas ventajas e inconvenientes de modelos de clasificación.
- Hemos destacado la importancia de una buena selección de métricas de evaluación
- Destacamos la importancia de ajustar propiamente los parámetros de los modelos y algoritmos.



Conclusiones clasificación I

Hemos identificado una serie de objetivos contrapuestos que se cumplen por lo general:

- Simplicidad del modelo vs precisión
- Interpretabilidad del modelo vs precisión
- Escalabilidad del modelo vs precisión
- Velocidad del modelo vs precisión

La **eficiencia** puede condicionar la aplicabilidad del algoritmo:

- En 2009 el algoritmo ganador del premio de 1.000.000 \$ Netflix, no se implementó nunca debido al coste computacional.

La disponibilidad de datos limita qué modelos aplicar (aprendizaje profundo).



¿Preguntas?
¡Gracias!

