



Seminario Permanente de Formación en Inteligencia Artificial Aplicada a la Defensa



Clasificación III Casos de estudio

Julián Luengo

Instituto Andaluz de Investigación en Data Science
and Computational Intelligence (DaSCI)

Dpto. Ciencias de la Computación e I.A.

Universidad de Granada

julianlm@decsai.ugr.es

<http://sci2s.ugr.es>



DaSCI



**UNIVERSIDAD
DE GRANADA**

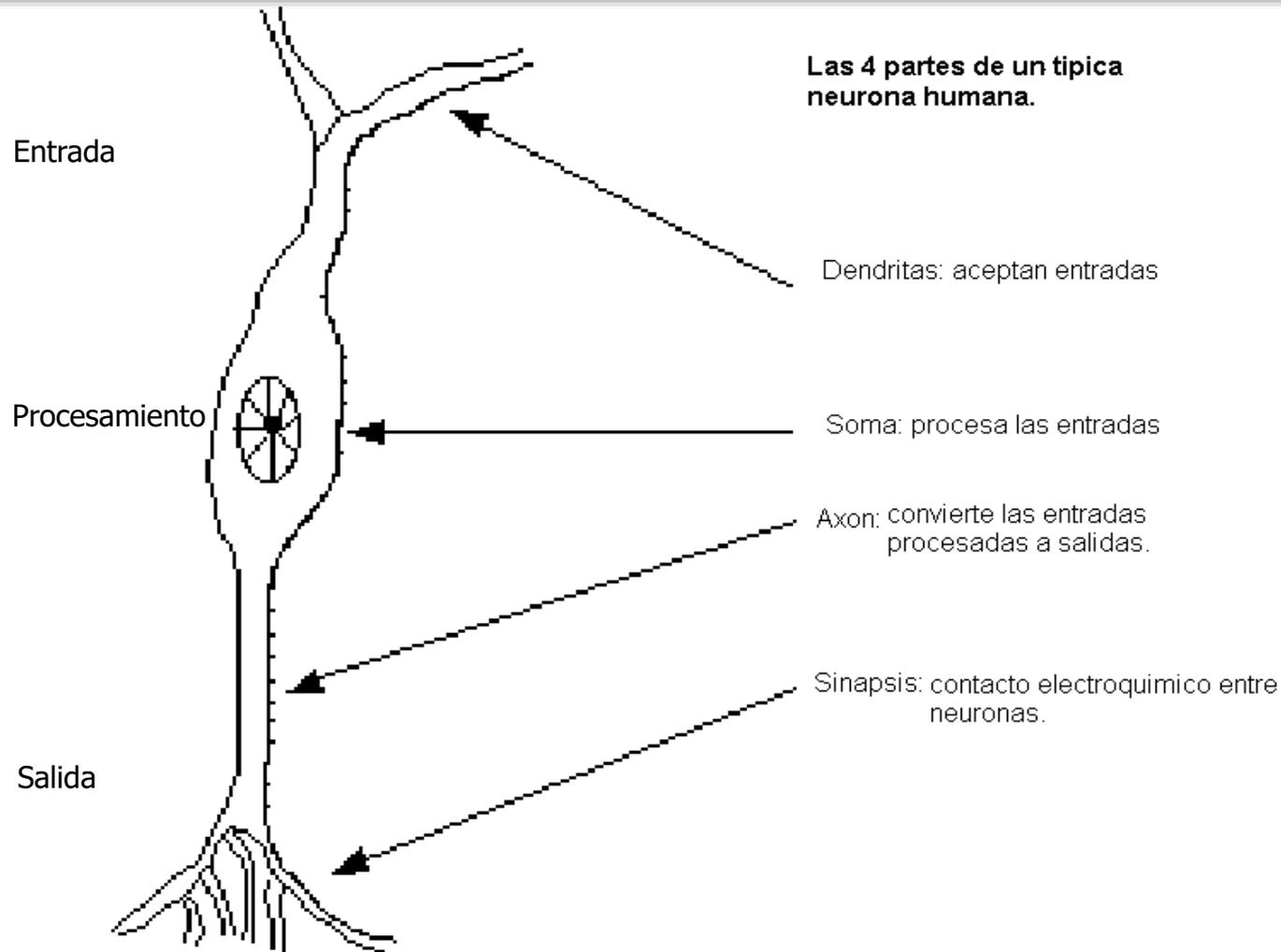
Resumen

1. Redes Neuronales. Fundamentos y Deep Learning
2. Ensemble Learning
3. Calidad de datos. Preprocesamiento de Datos
4. Big Data
5. Comentarios Finales

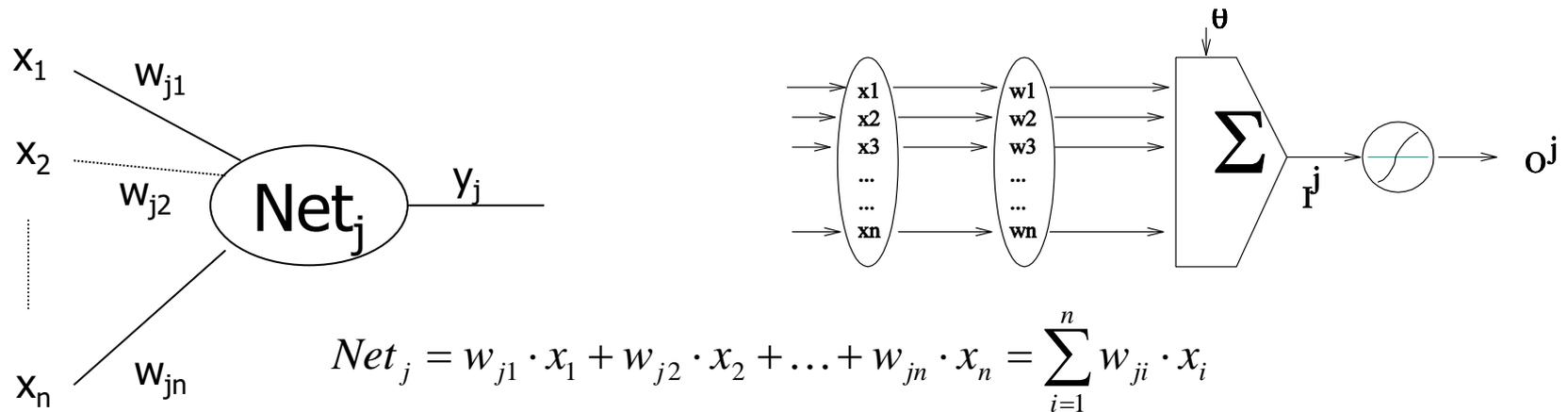
Resumen

- 1. Redes Neuronales. Fundamentos y Deep Learning**
2. Ensemble Learning
3. Calidad de datos. Preprocesamiento de Datos
4. Big Data
5. Comentarios Finales

Redes Neuronales: Las neuronas biológicas



Redes Neuronales: Las neuronas biológicas



■ En definitiva, los componentes básicos de una red neuronal son:

- Vector de pesos (w_{ij}), uno por cada entrada desde a la neurona j
- Un sesgo (θ_j) asociado a la neurona

$$y_j = f(Net_j - \theta_j) = f\left(\sum w_{ji} \cdot x_i - \theta\right)$$

- Los datos de entrada $\{x_1, \dots, x_n\}$ a una neurona N_j se hacen corresponder con un número real utilizando los componentes anteriores

- Una **función de activación** f . Puede ser muy sencilla (función escalón) aunque normalmente es una función sigmoideal

Tipos de funciones de transferencia

- **Función de escalón o**

Haviside. Representa una neurona con sólo dos estados de activación: activada (1) y inhibida (0 ó -1)

$$y_j = H(Net_j - \theta_j) = \begin{cases} 1, & \text{si } Net_j \geq \theta_j \\ -1, & \text{si } Net_j < \theta_j \end{cases}$$

- **Función lineal:** $y_j = Net_j - \theta_j$

- **Función lineal a tramos:**

$$y_j = \begin{cases} 1, & \text{si } Net_j \geq \theta_j + a \\ Net_j - \theta_j, & \text{si } |Net_j - \theta_j| < a \\ -1, & \text{si } Net_j < \theta_j - a \end{cases}$$

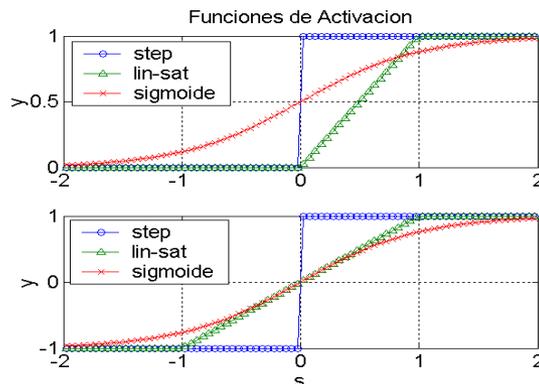
- **Función sigmoidal:**

$$y_j = \frac{1}{1 + e^{-\lambda(Net_j - \theta_j)}}$$

$$y_j = \frac{2}{1 + e^{-\lambda(Net_j - \theta_j)}} - 1$$

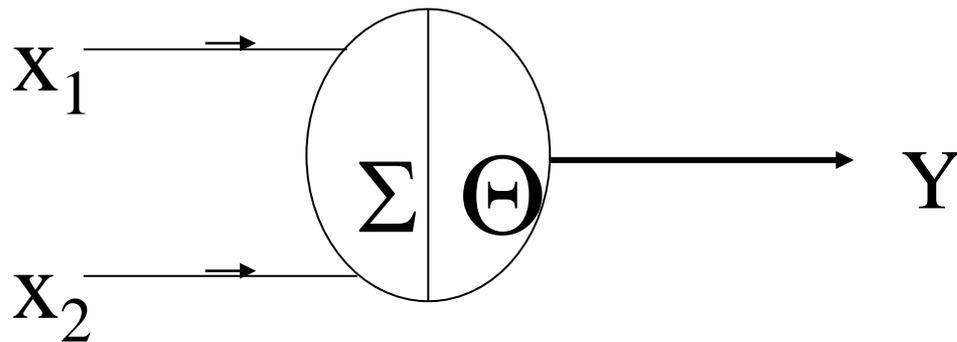
- **Función base radial:**

$$y_j = e^{-\left(\frac{Net_j - \theta_j}{\sigma}\right)^2}$$



Ejemplo de funcionamiento de una neurona

- **Ejemplo:** neurona simple con dos entradas y una salida. Perceptrón.



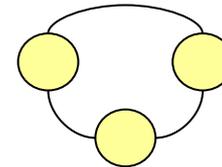
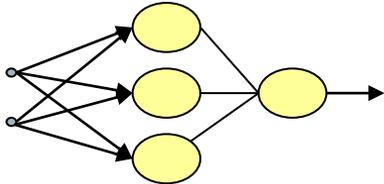
- Intentemos implementar la función OR con la ayuda de esta neurona

<u>X1</u>	<u>X2</u>	<u>Salida</u>
0	0	0
0	1	1
1	0	1
1	1	1

¿Qué es una red neuronal?

Una **red de neuronas artificiales** está caracterizada por su:

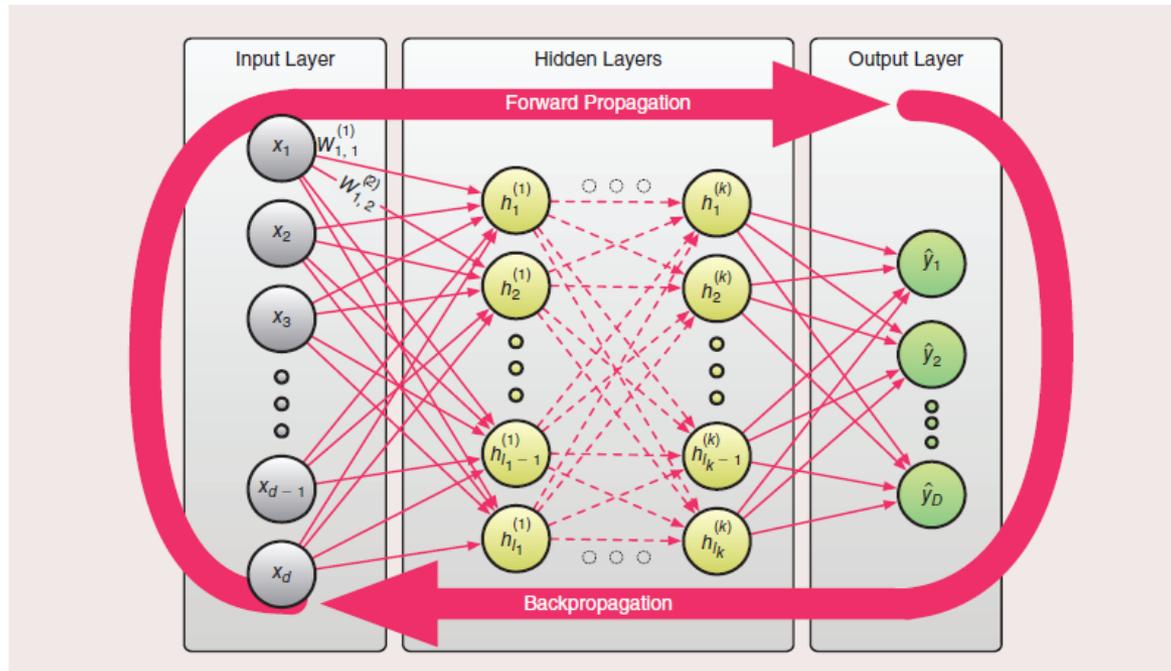
- **Arquitectura:** Estructura o patrón de conexiones entre las unidades de proceso



- **Dinámica de la Computación** que nos expresa el valor que toman las unidades de proceso y que se basa en unas **funciones de activación (o de transferencia)** que especifican como se transforman las señales de entrada de la unidad de proceso en la señal de salida
- **Algoritmo de Entrenamiento o Aprendizaje:** Procedimiento para determinar los pesos de las conexiones. Una característica muy importante de estas redes es su naturaleza **adaptativa**, donde el "**aprendizaje con ejemplos**" sustituye a la "programación" en la resolución de problemas.

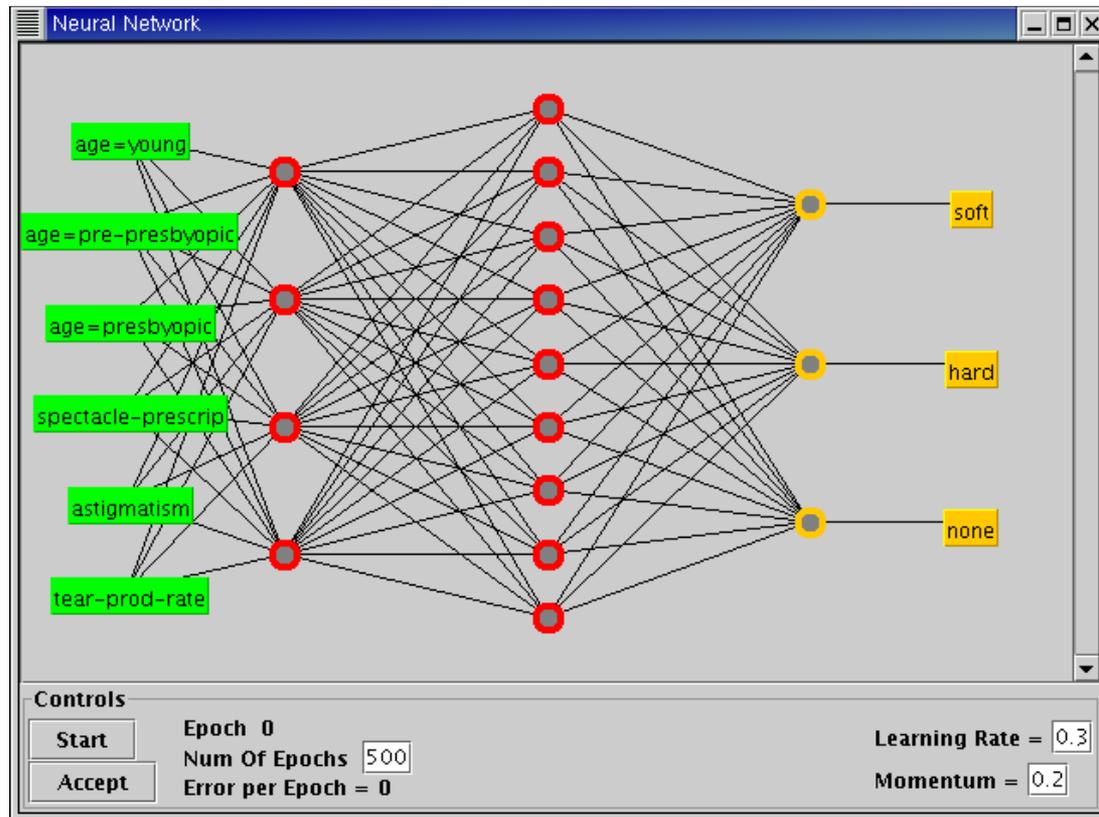
Clasificadores basados en redes neuronales. Multi-Layer Perceptron (MLP)

- Todas las variables numéricas se normalizan $[-1,1]$
 - Algoritmo de *Backpropagation* o retropropagación:
 - Basado en la técnica del gradiente descendiente
 - No permite conexiones hacia atrás (retroalimentación) en la red
- 1 *epoch* = procesamiento de todos los ejemplos de la BD



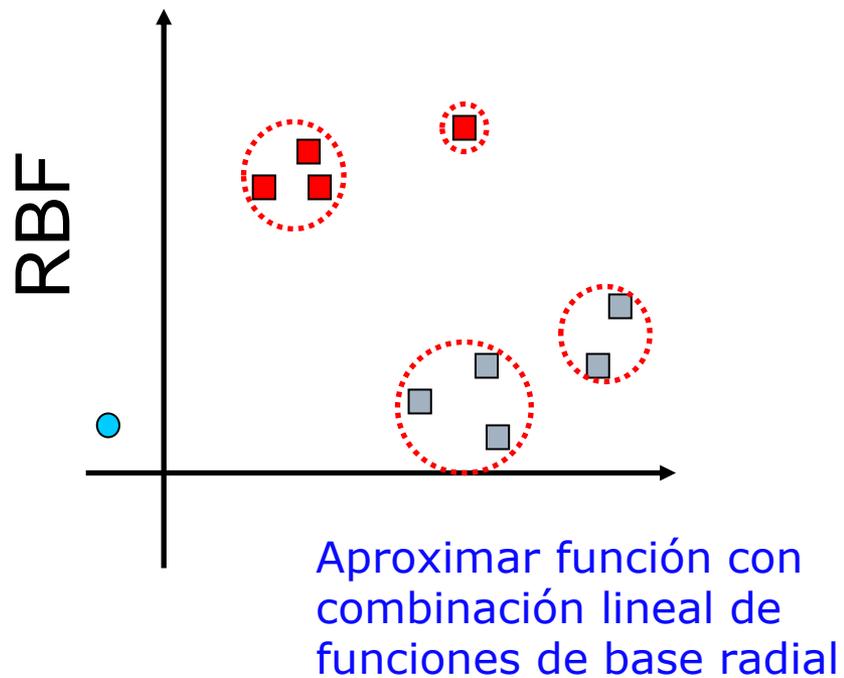
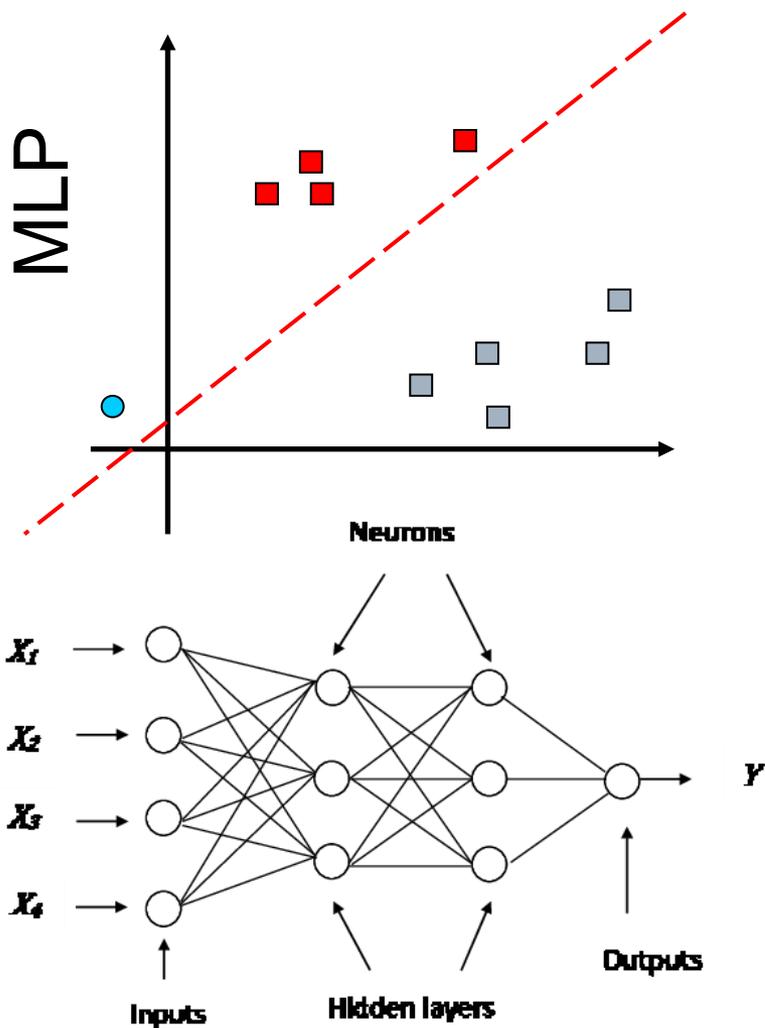
Clasificadores basados en redes neuronales. Multi-Layer Perceptron (MLP)

- Ejemplos con la BD lentes
 - Capa oculta = a, t



<https://playground.tensorflow.org/>

Clasificadores basados en redes neuronales. Redes de Funciones de base Radial (RBF)



$$f(x) = \sum w_j h_j(x)$$

$$h_j(x) = \exp\left(-\frac{(x-c_j)^2}{r_j^2}\right)$$

Where c_j is center of a region,
 r_j is width of the receptive field

Deep Learning



Deep Learning : El aprendizaje profundo es un conjunto de algoritmos que intenta modelar abstracciones de alto nivel en los datos mediante el uso de arquitecturas compuestas de transformación no lineales múltiples.

Nota: Deep Learning introduce el uso de estructuras de aprendizaje que requieren de arquitecturas de procesamiento eficiente y distribuido (GPU, Spark, ...) y muestra resultados importantes en el procesamiento de imágenes, habla, lenguaje natural, ...

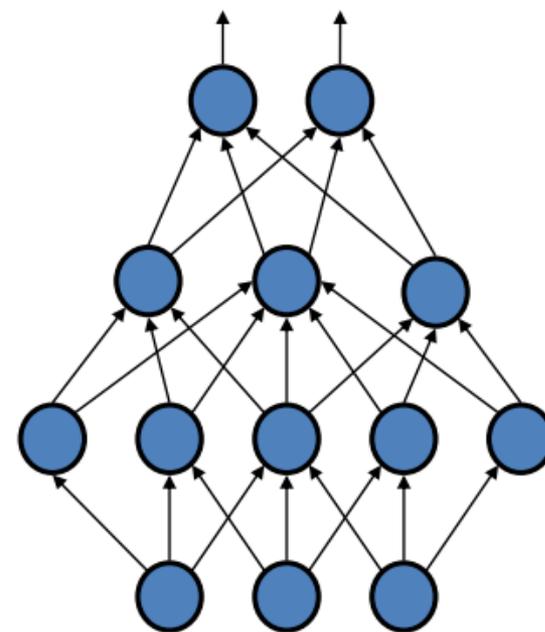


Deep Learning



Problems with Backpropagation

- Multiple hidden Layers
- Get stuck in local optima
 - start weights from random positions
- Slow convergence to optimum
 - large training set needed
- Only use labeled data
 - most data is unlabeled

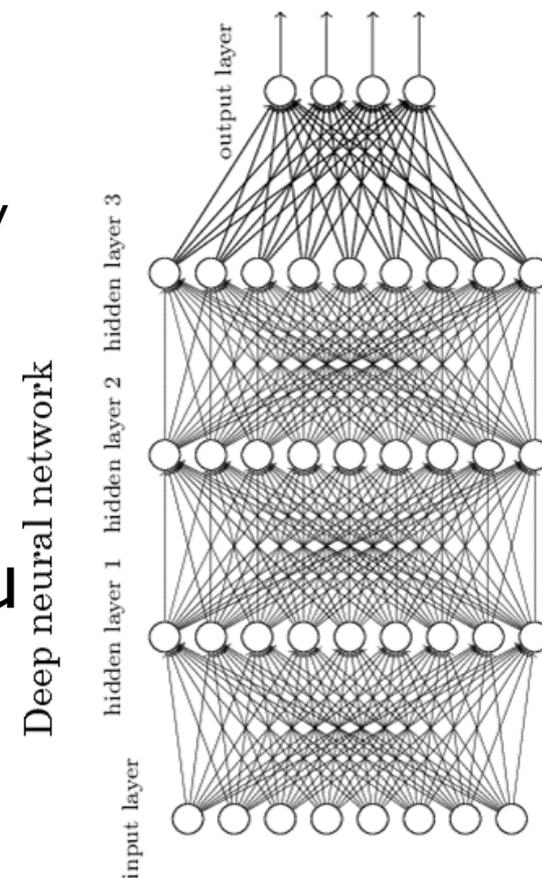


Deep Learning



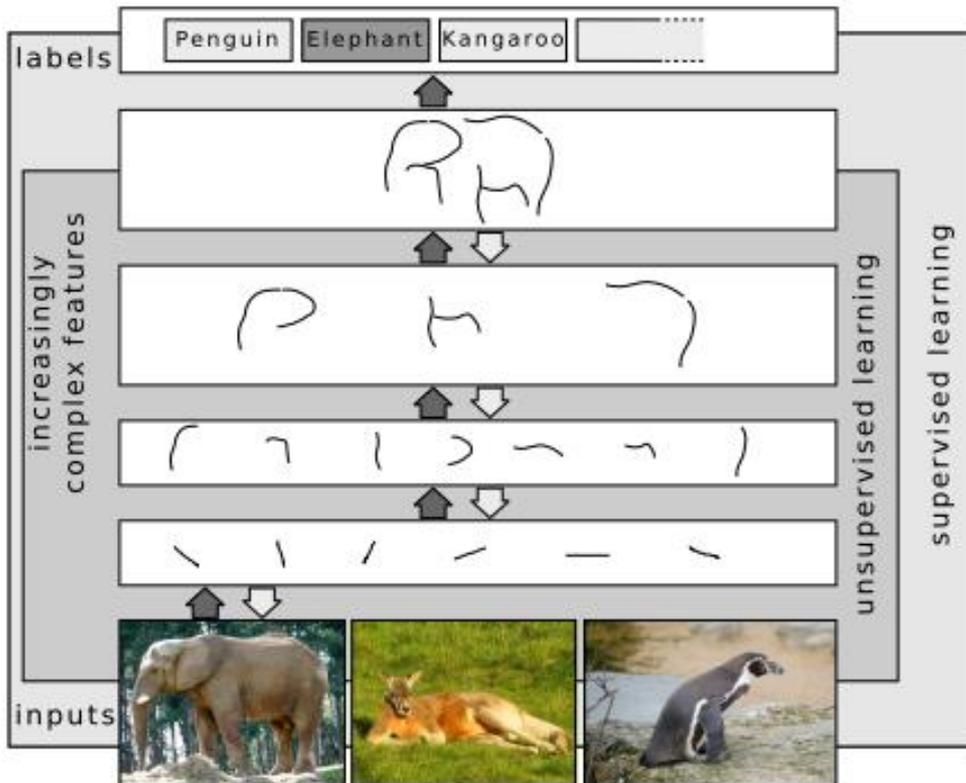
Deep Architecture (Train networks with many layers)

- Multiple hidden layers
- Motivation (why go deep?)
 - Approximate complex decision boundary
 - Fewer computational units same functional mapping
 - Hierarchical Learning
 - Increasingly complex features
 - Work well in different domains
 - Vision, Audio, ...



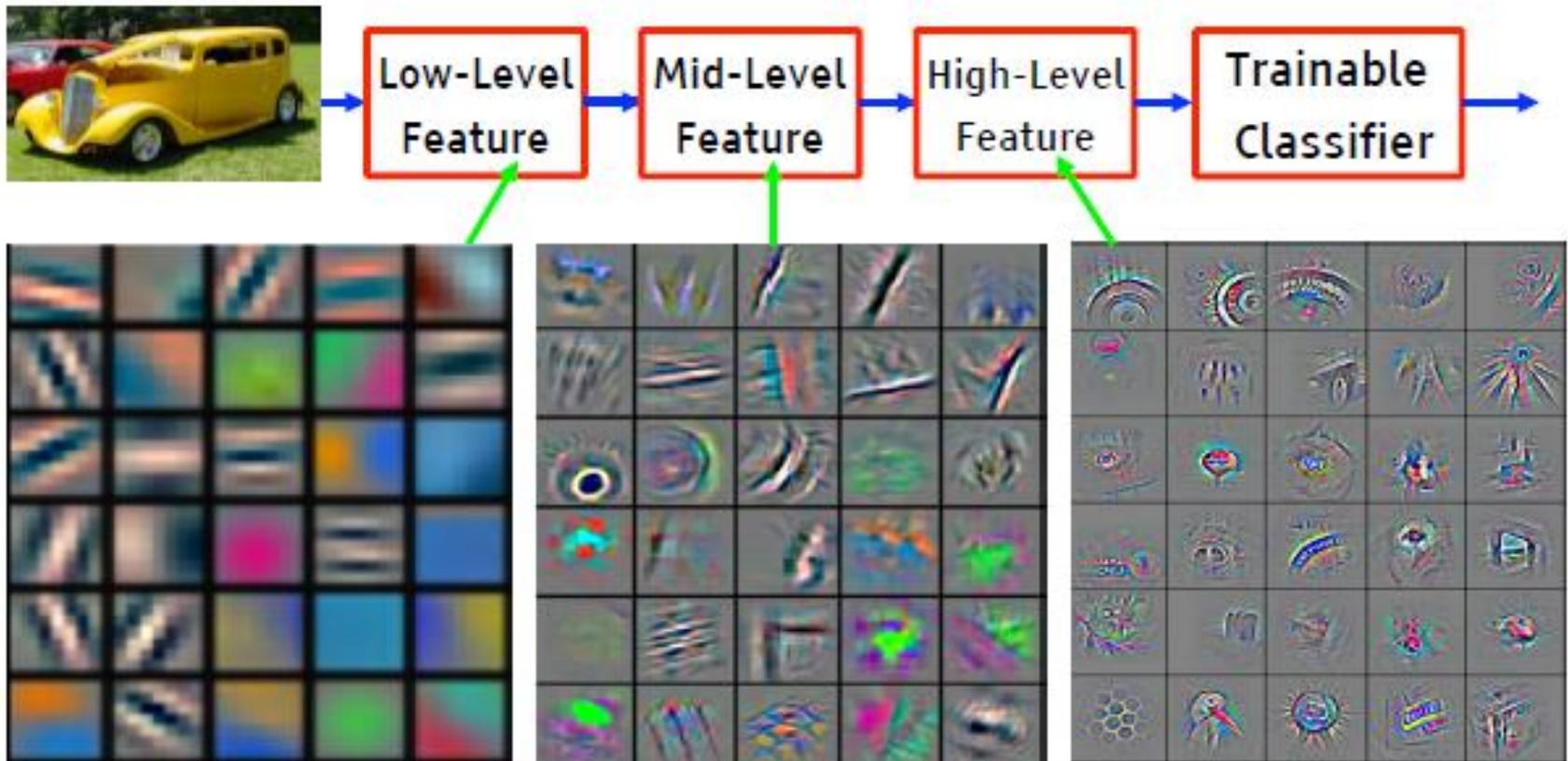


Deep Learning

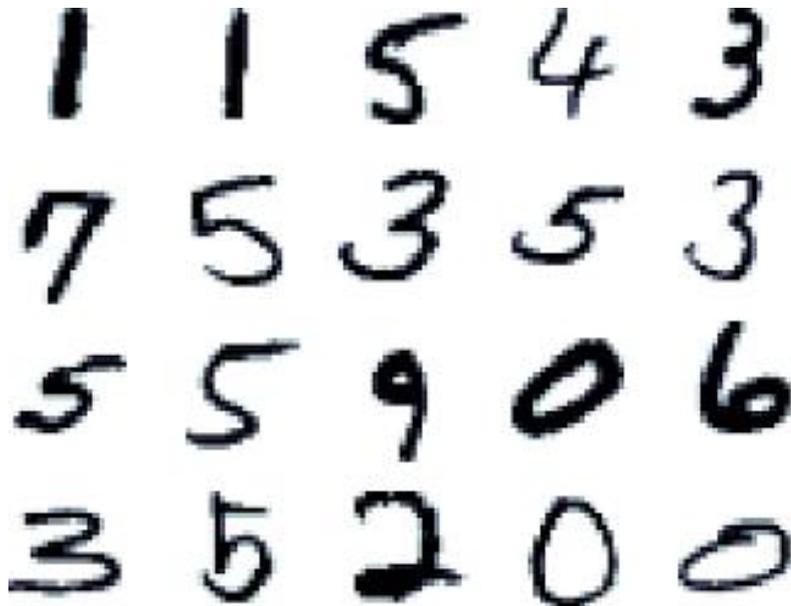


- Genera *features* complejas a partir de simples
- Paso de aprendizaje no supervisado al aprendizaje supervisado

Deep Learning



Deep Learning, reconociendo números a mano



- Desarrollar un reconocedor de dígitos es uno de los **problemas clásicos** de la ciencia de datos.
- Sirve de **benchmark** para probar los nuevos algoritmos. ¡Ningún humano acierta el 100%!
- **Aplicación práctica:** detección de matrículas, conversión de escritura a mano en texto ...

9 6 6 5 4 0 7 4 0 1
3 1 3 4 7 2 7 1 2 1
1 7 4 2 3 5 1 2 4 4



9 6 6 6 4 0 7 4 0 1
3 1 3 4 7 2 7 1 2 1
1 7 4 2 3 5 1 2 4 4

Deep Learning, reconociendo números a mano



Ejemplo de errores que comenten los sistemas: Un modelo de DL con 99.67% de acierto, 33 de 10,000 imágenes de test de error. La etiqueta de arriba a la derecha es la clasificación correcta, según los datos del MNIST, mientras que en la parte inferior a la derecha está la etiqueta de salida:



Deep Learning, reconociendo números a mano



Video que muestra gráficamente la conexión de neuronas en diferentes modelos



Deep Learning, reconociendo números a mano



MNIST-NET10
(10 errores)
Record Mundial

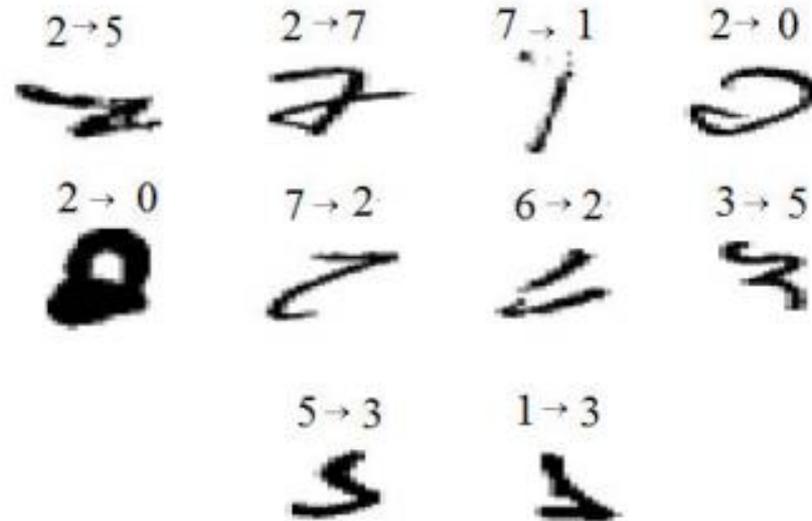
Cornell University the Sim

arXiv.org > cs > arXiv:2001.11486 Search...
Help | Advanc

Computer Science > Machine Learning

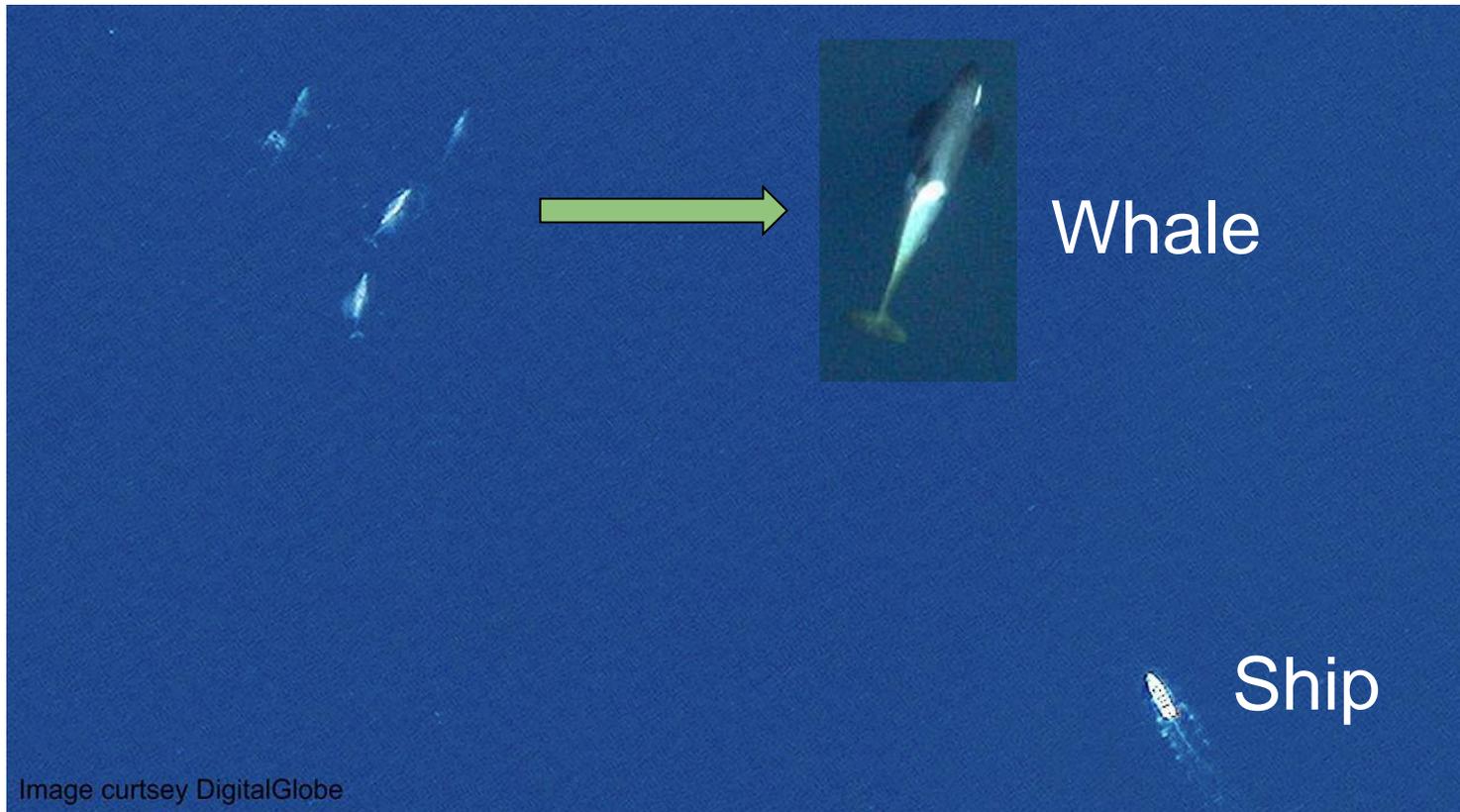
MNIST-NET10: A heterogeneous deep networks fusion based on the degree of certainty to reach 0.1 error rate. Ensembles overview and proposal

S. Tabik, R.F. Alvear-Sandoval, M.M. Ruiz, J.L. Sancho-Gómez, A.R. Figueiras-Vidal, F. Herrera
(Submitted on 30 Jan 2020 (v1), last revised 7 Apr 2020 (this version, v2))

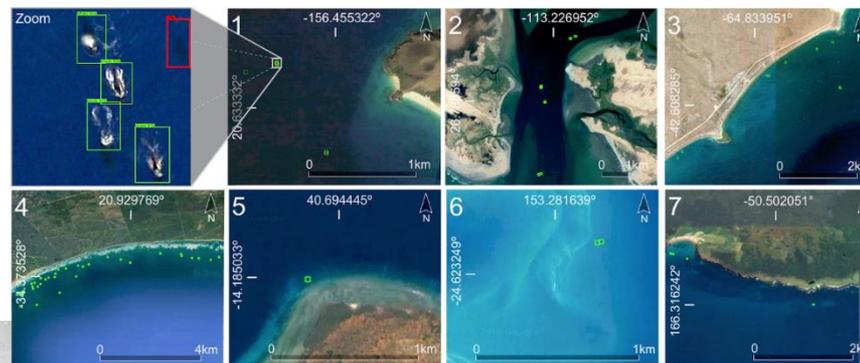
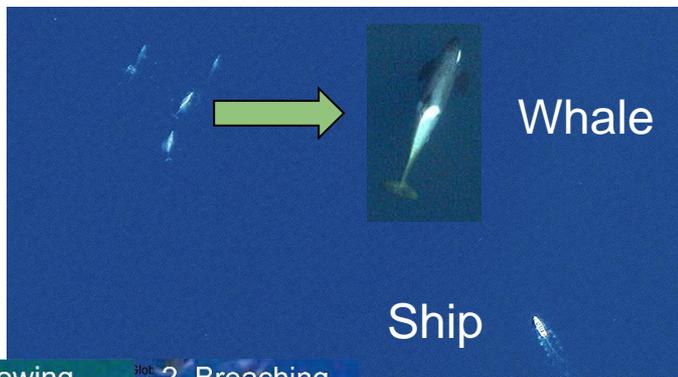


www.cybercontrols.org

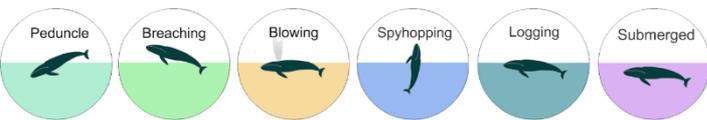
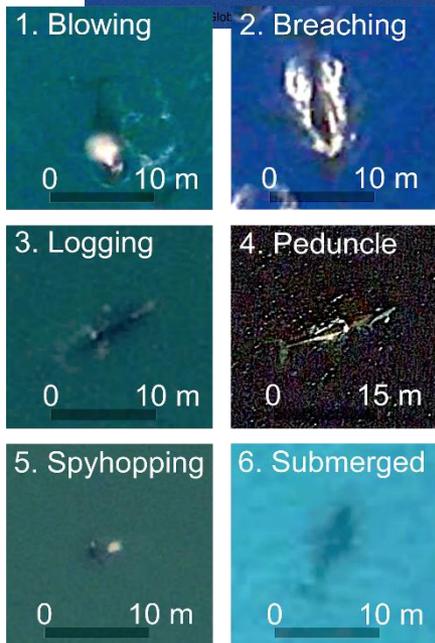
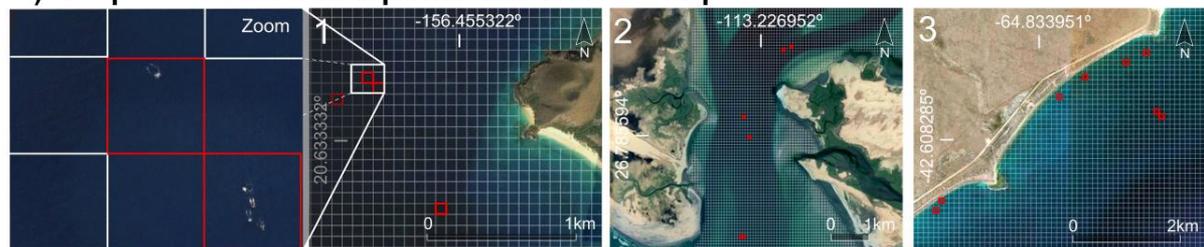
Deep Learning: Detección y conteo de bancos de ballenas



Deep Learning: Detección y conteo de bancos de ballenas



B) Output from the whale presence detection step



Resumen

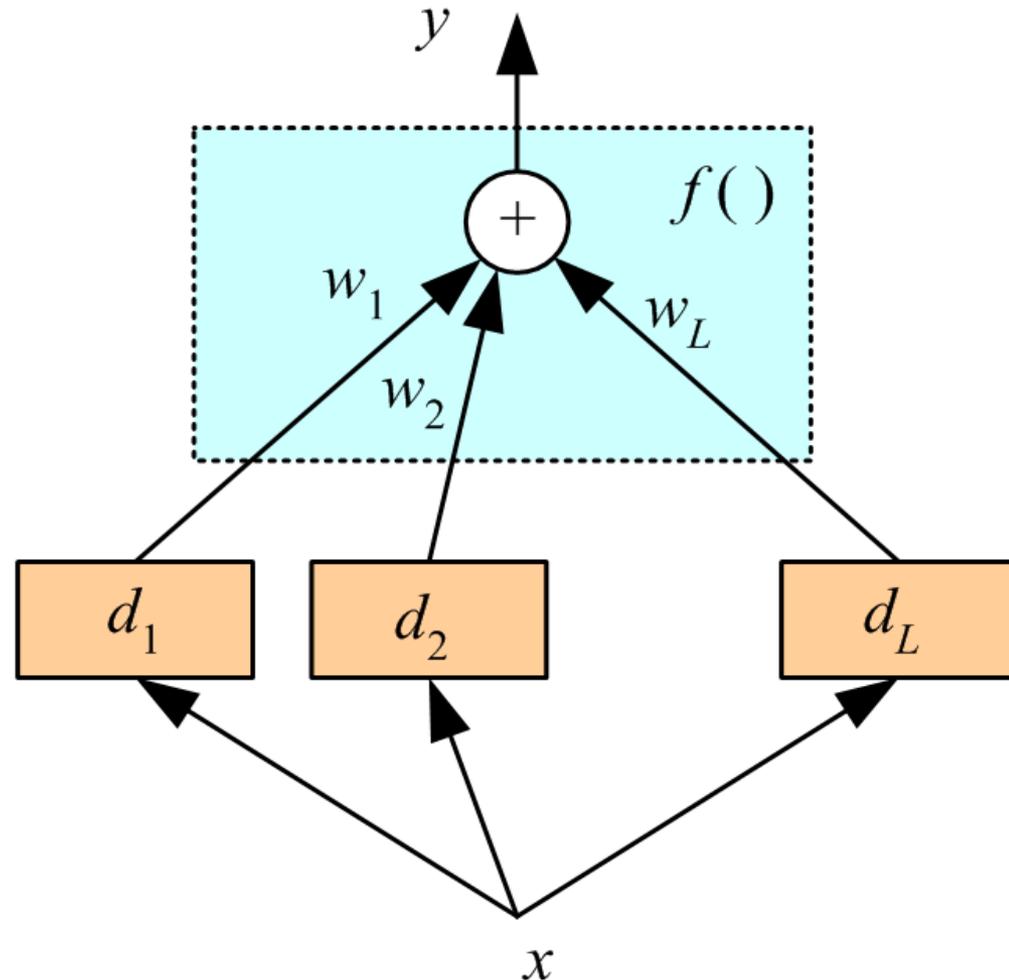
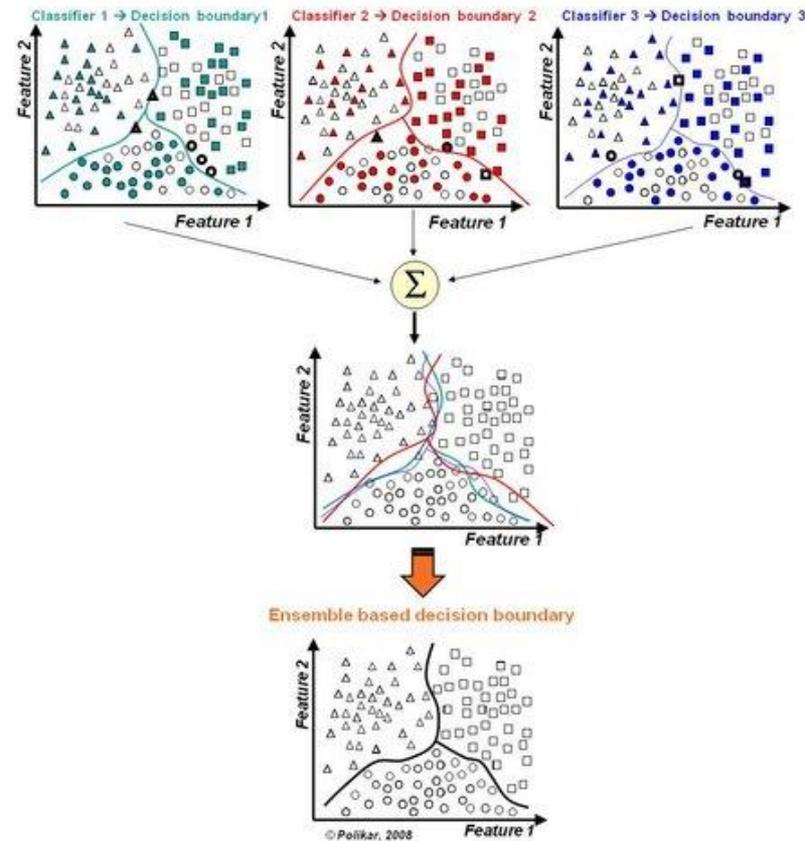
1. Redes Neuronales. Fundamentos y Deep Learning
- 2. Ensemble Learning**
3. Calidad de datos. Preprocesamiento de Datos
4. Big Data
5. Comentarios Finales

Rationale for Ensemble Learning

- No Free Lunch theorem: There is no algorithm that is always the most accurate
- Generate a group of **base-learners** which when combined have higher accuracy
- **Diversity** and **Accuracy** among classifiers are the key points for the success of ensembles.
- Different learners use different
 - Algorithms / Parameters
 - Training sets / Subproblems

Voting

- Linear combination



Voting

Ejemplo: Fusión de 2 modelos de Deep learning mediante un voto asociado al porcentaje de clasificadores exitosos de cada uno de ellos

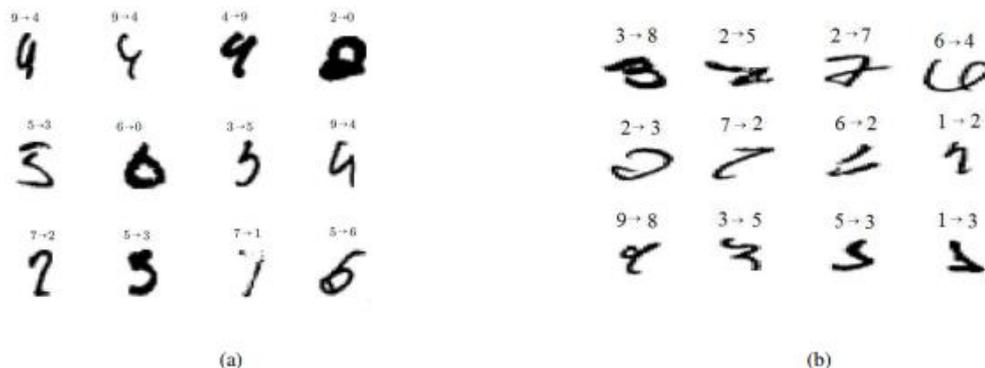


Figure 3: Misclassified test digits (label → results) by (a) a run of ensemble **FS1** (Subsection 4.2) and (b) a run of ensemble **FS2** with CapsNet input (Subsection 4.3).

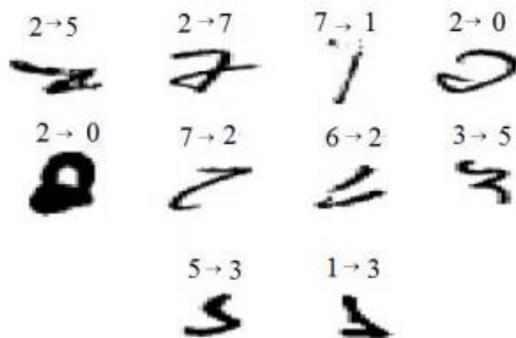
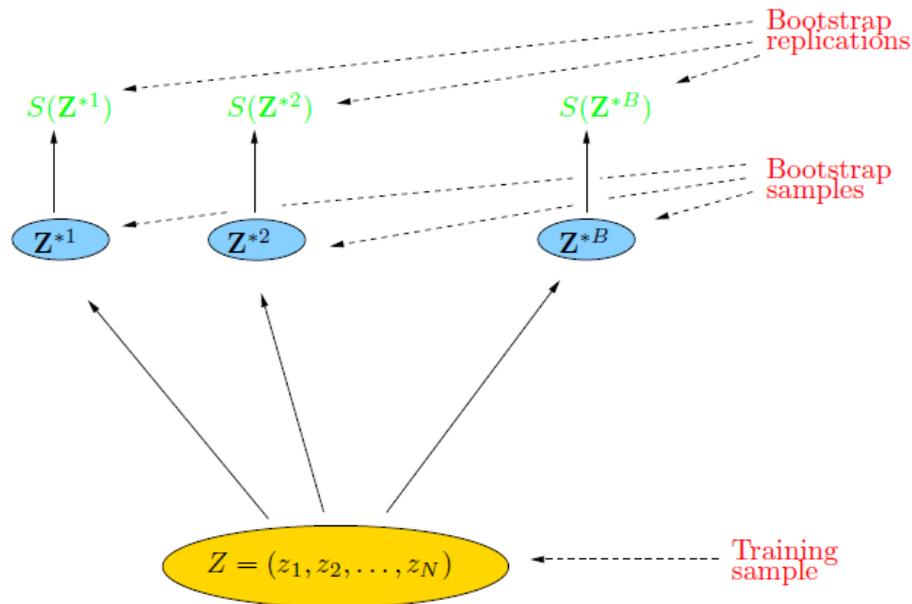


Figure 4: Misclassified test digits (label → result) obtained from fusing **FS1** and **FS2**.

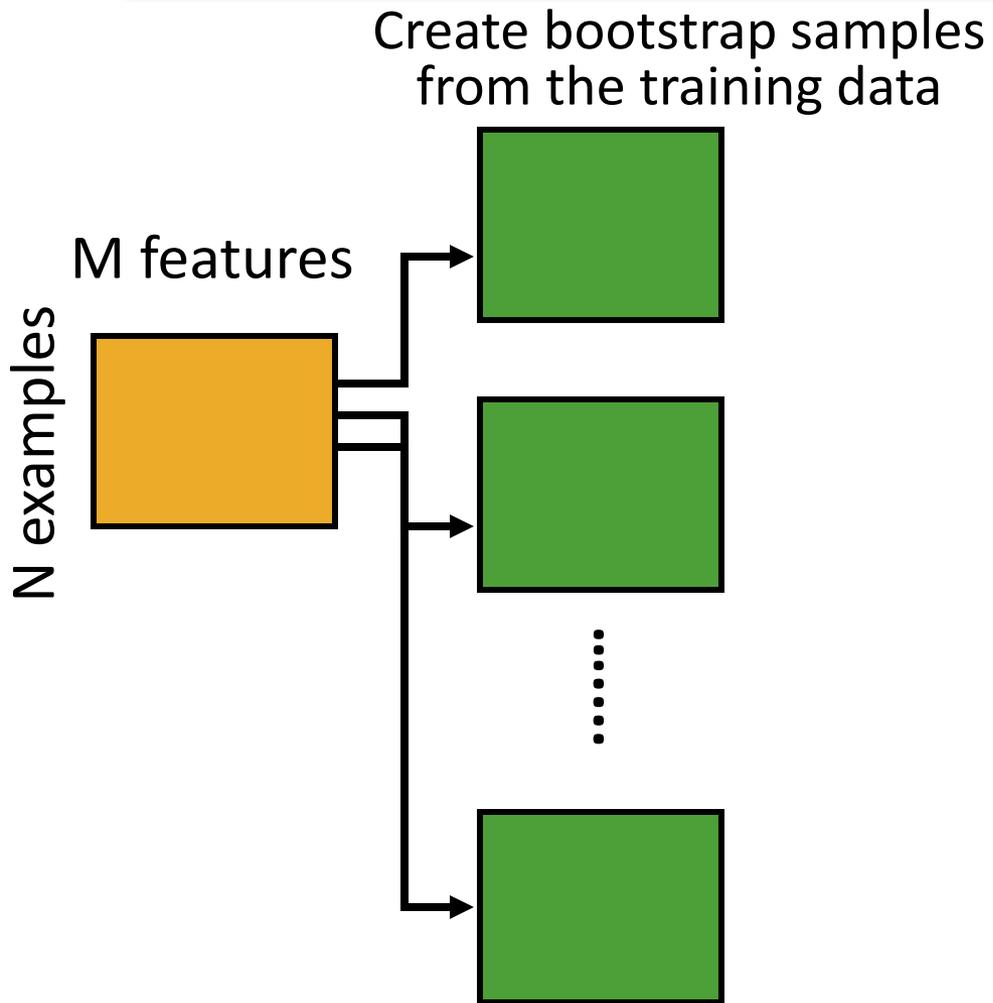
Bootstrap

The basic idea:

randomly draw datasets *with replacement* from the training data, each sample *the same size as the original training set*

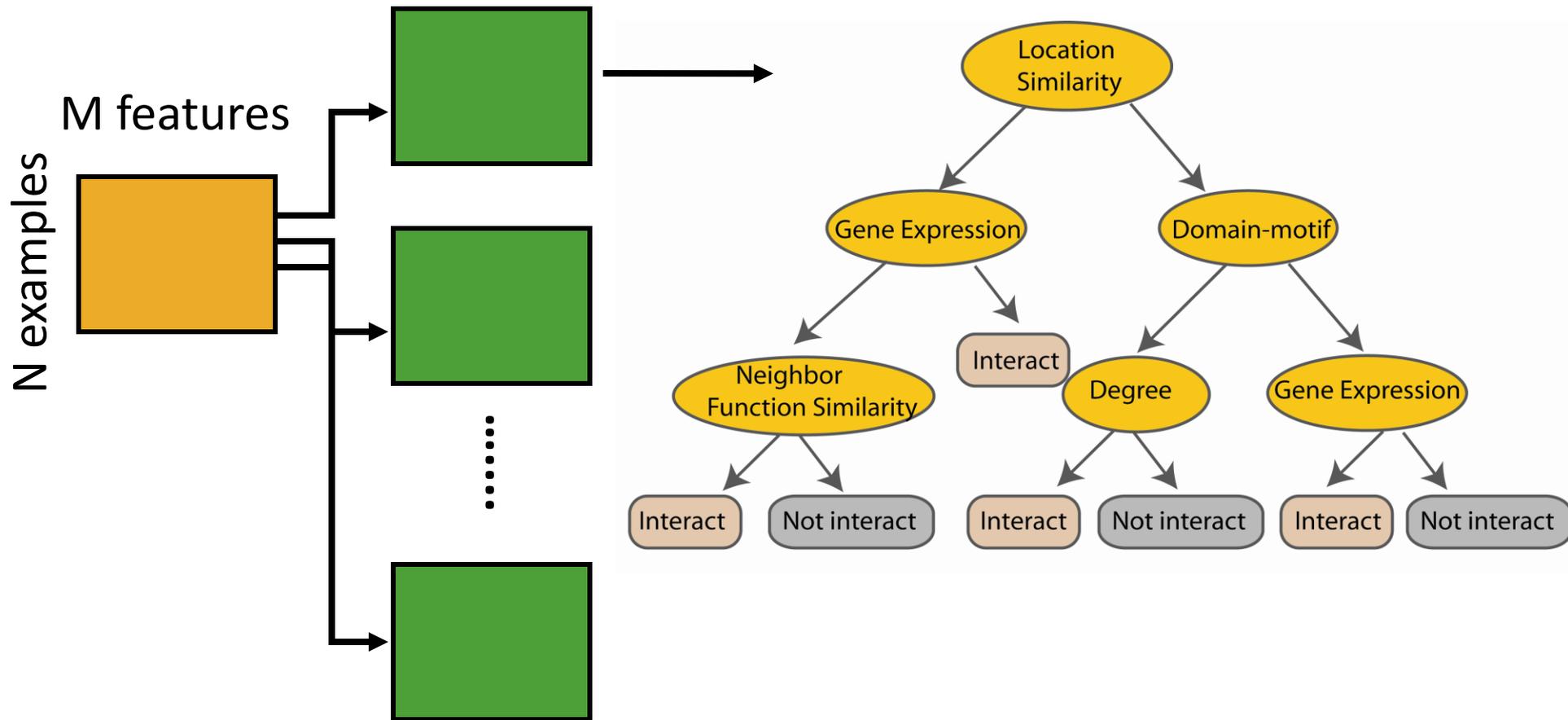


Bagging

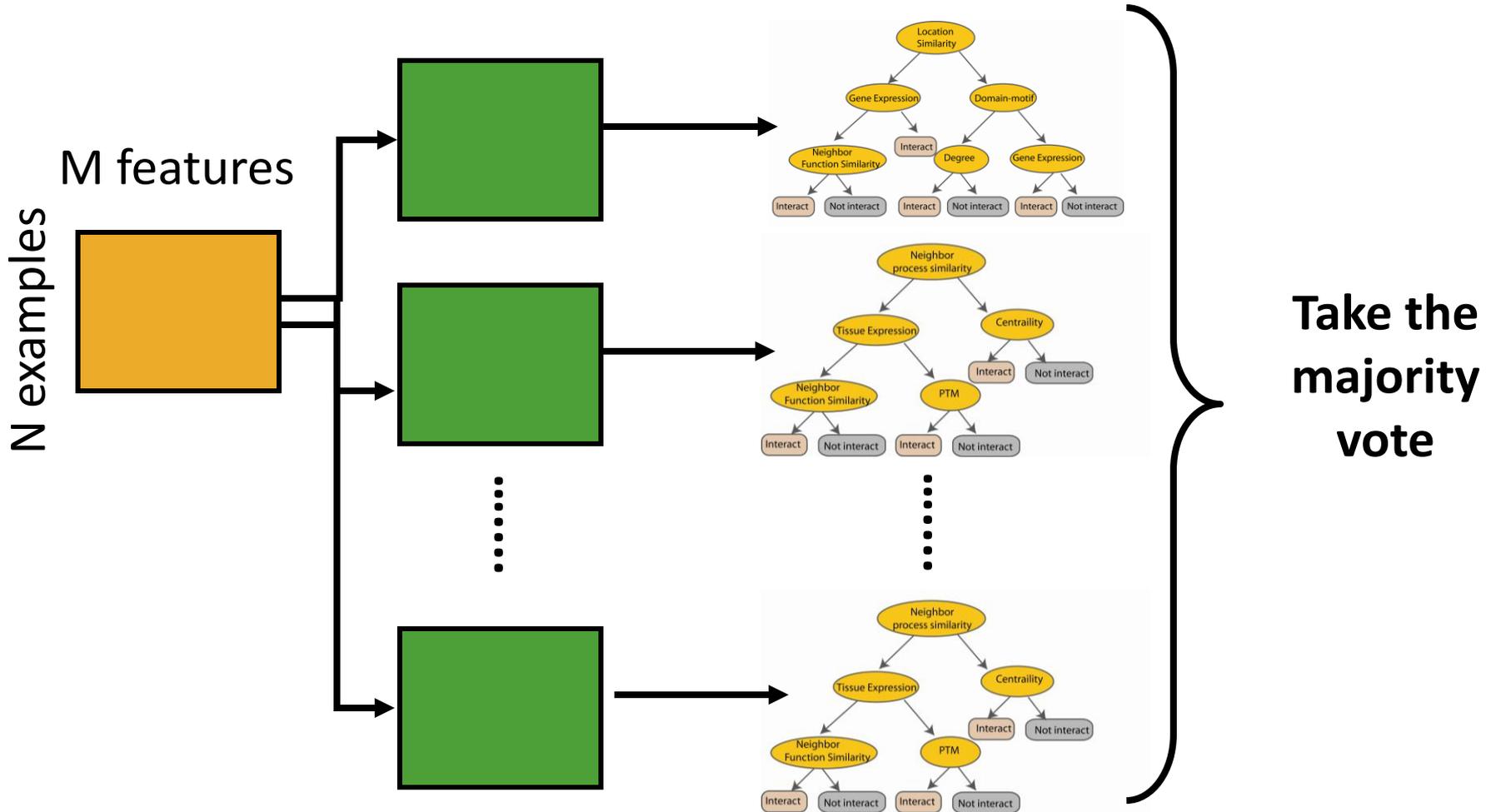


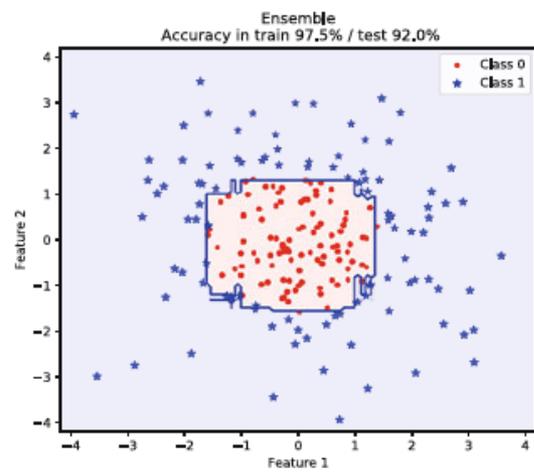
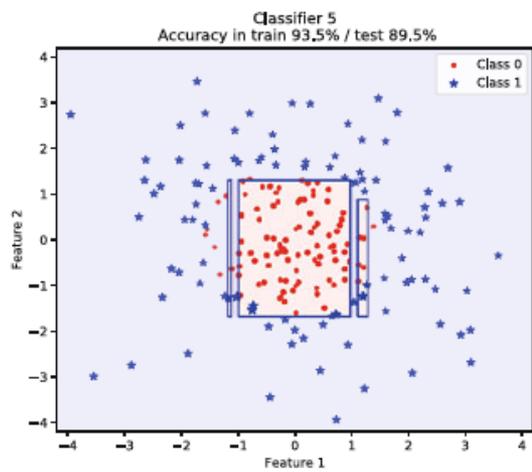
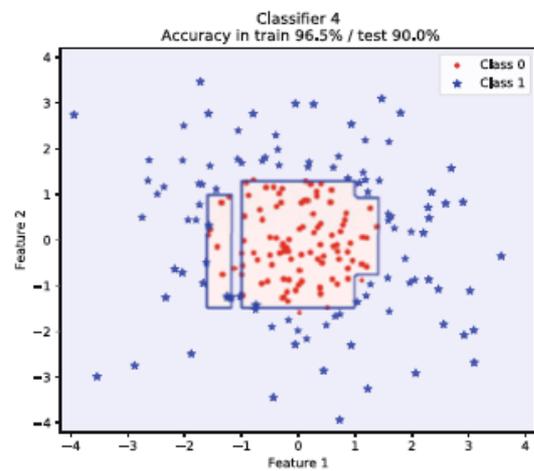
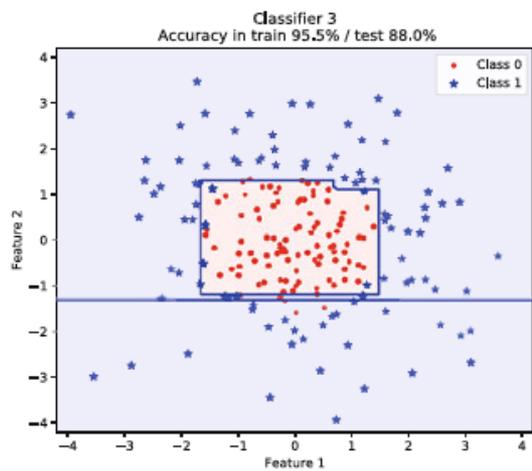
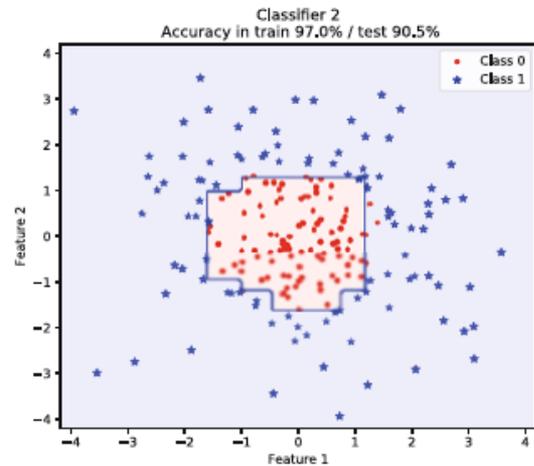
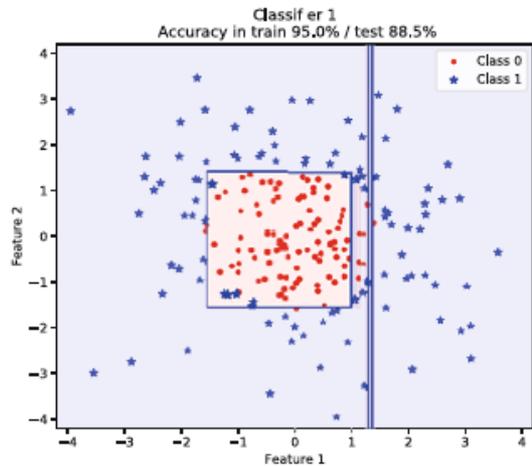
Random Forest Classifier

Construct a decision tree



Random Forest Classifier

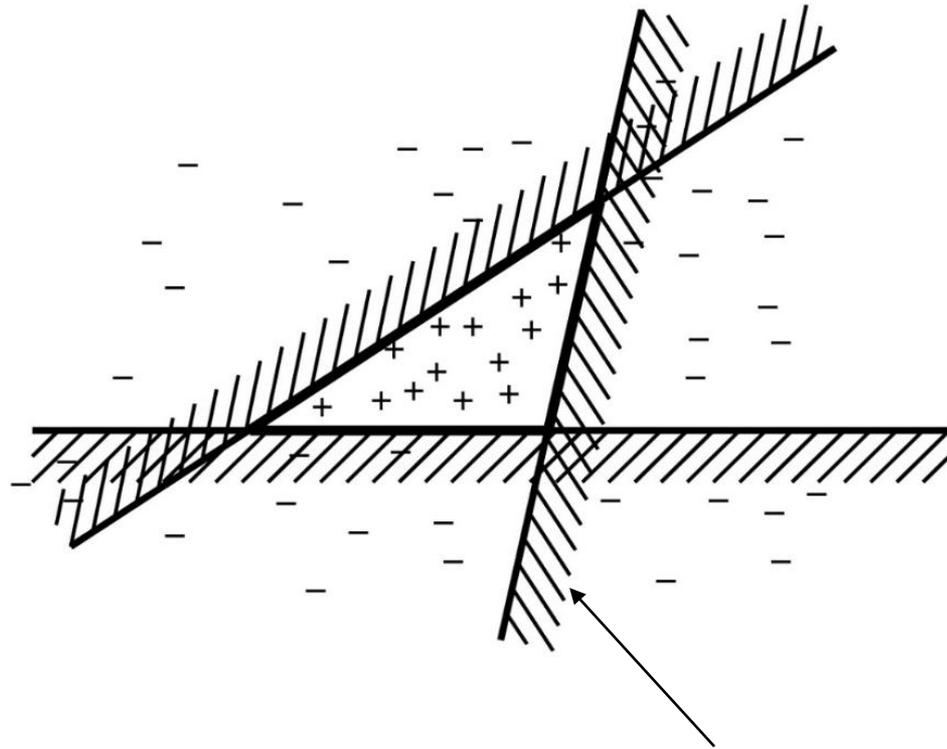




Boosting

- Train classifiers (e.g. decision trees) in a sequence.
- A new classifier should focus on those cases which were incorrectly classified in the last round.
- Combine the classifiers by letting them vote on the final prediction (like bagging).
- Each classifier is “weak” but the ensemble is “strong.”
- AdaBoost is a specific boosting method.

Example

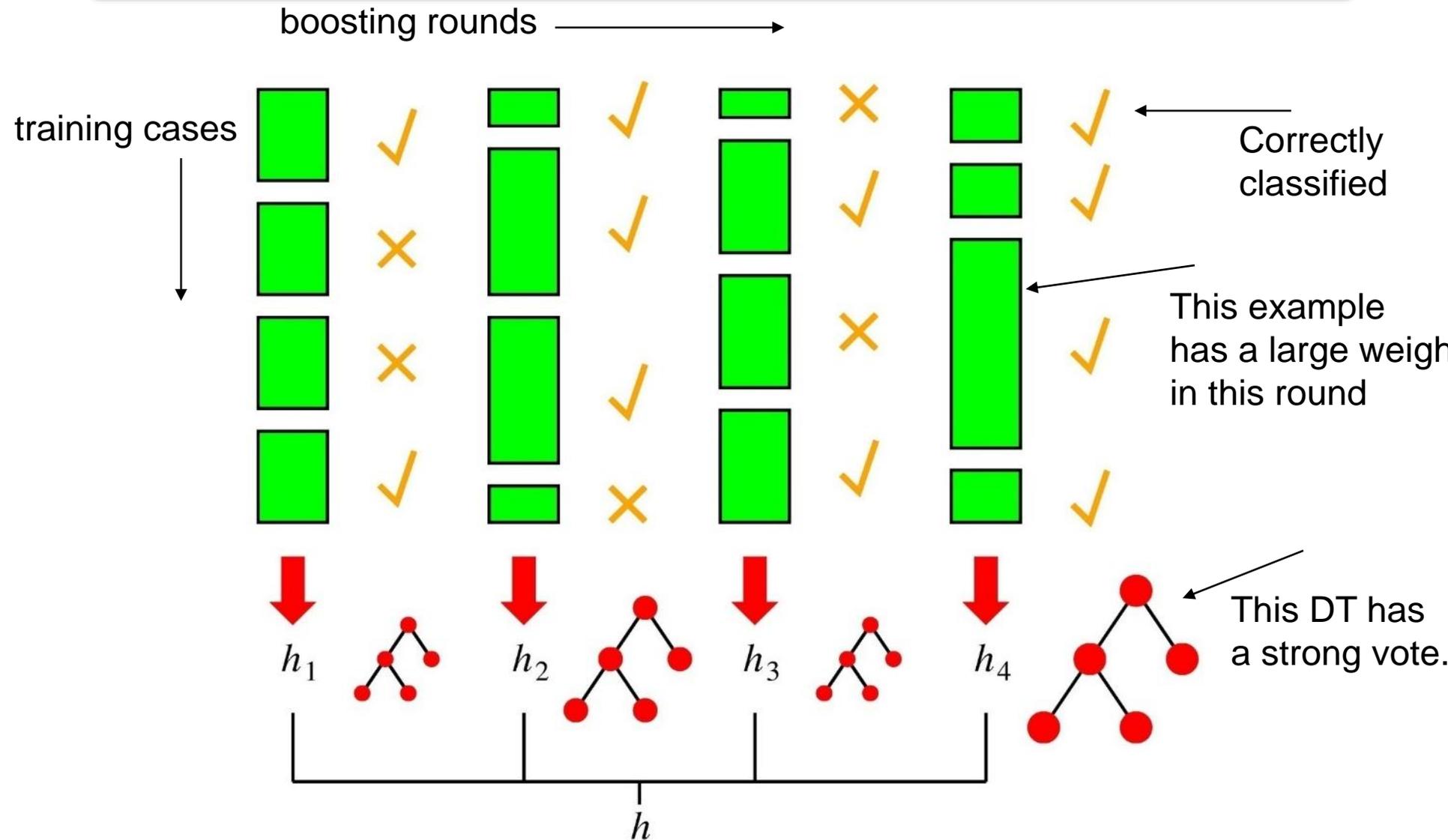


This line is one simple classifier saying that everything to the left + and everything to the right is -

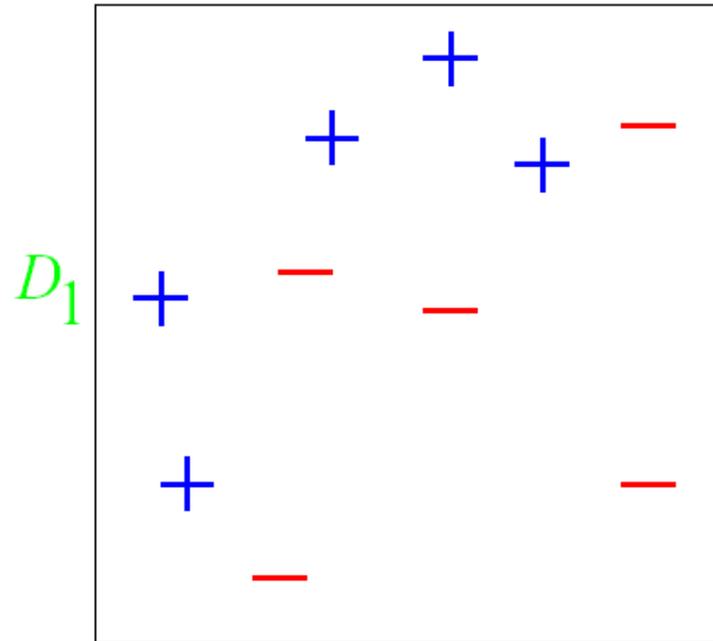
Boosting Intuition

- We adaptively weight each data case.
- Data cases which are wrongly classified get high weight (the algorithm will focus on them)
- Each boosting round learns a new (simple) classifier on the weighed dataset.
- These classifiers are weighed to combine them into a single powerful classifier.
- Classifiers that obtain low training error rate have high weight.

Boosting in a Picture



And in animation

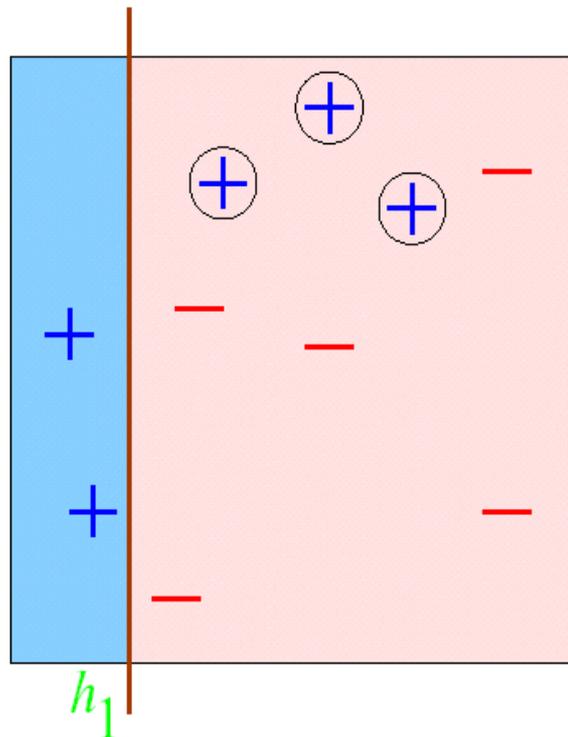


Original training set: equal weights to all training samples

AdaBoost example

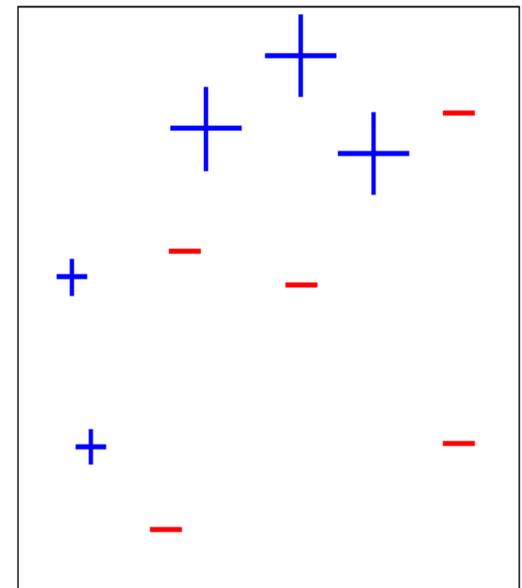
ϵ = error rate of classifier
 α = weight of classifier

ROUND 1



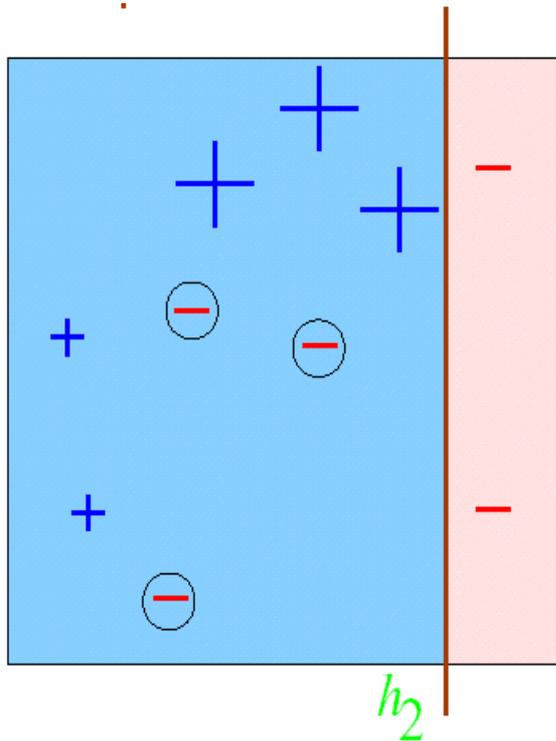
$$\epsilon_1 = 0.30$$
$$\alpha_1 = 0.42$$

D_2



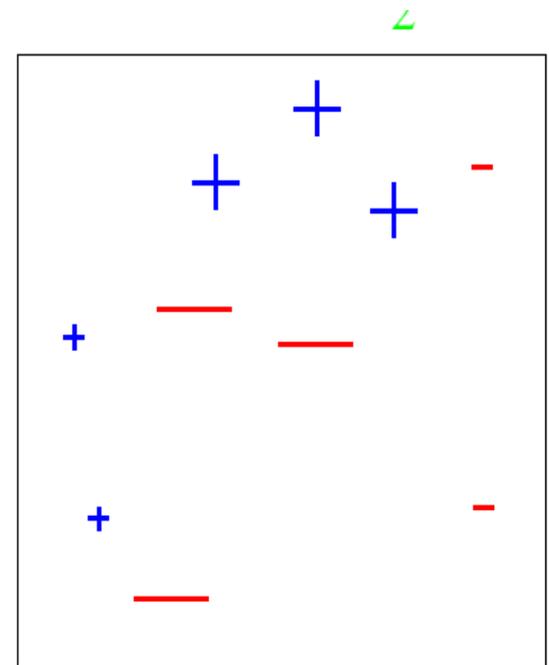
AdaBoost example

ROUND 2



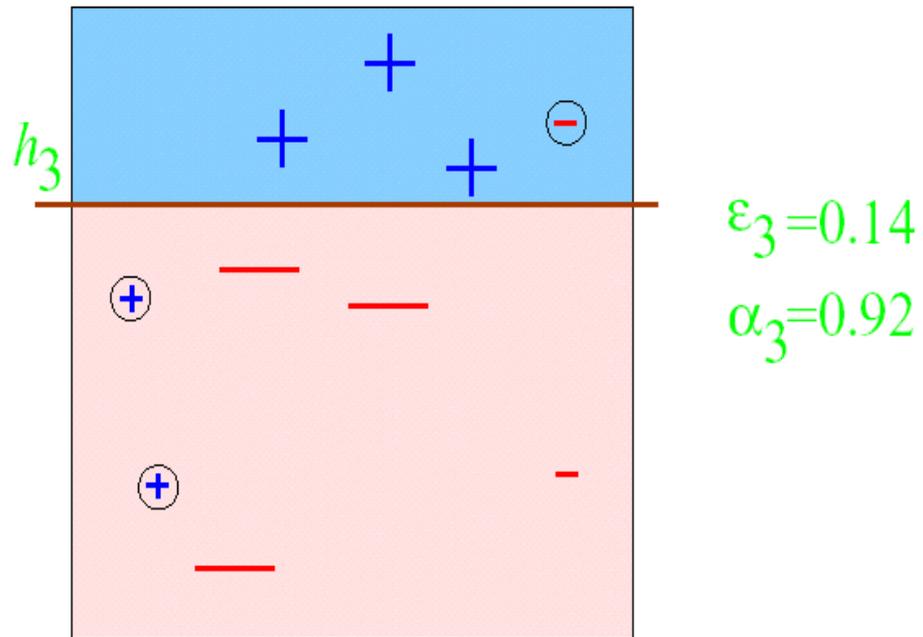
$$\begin{aligned}\epsilon_2 &= 0.21 \\ \alpha_2 &= 0.65\end{aligned}$$

D_3

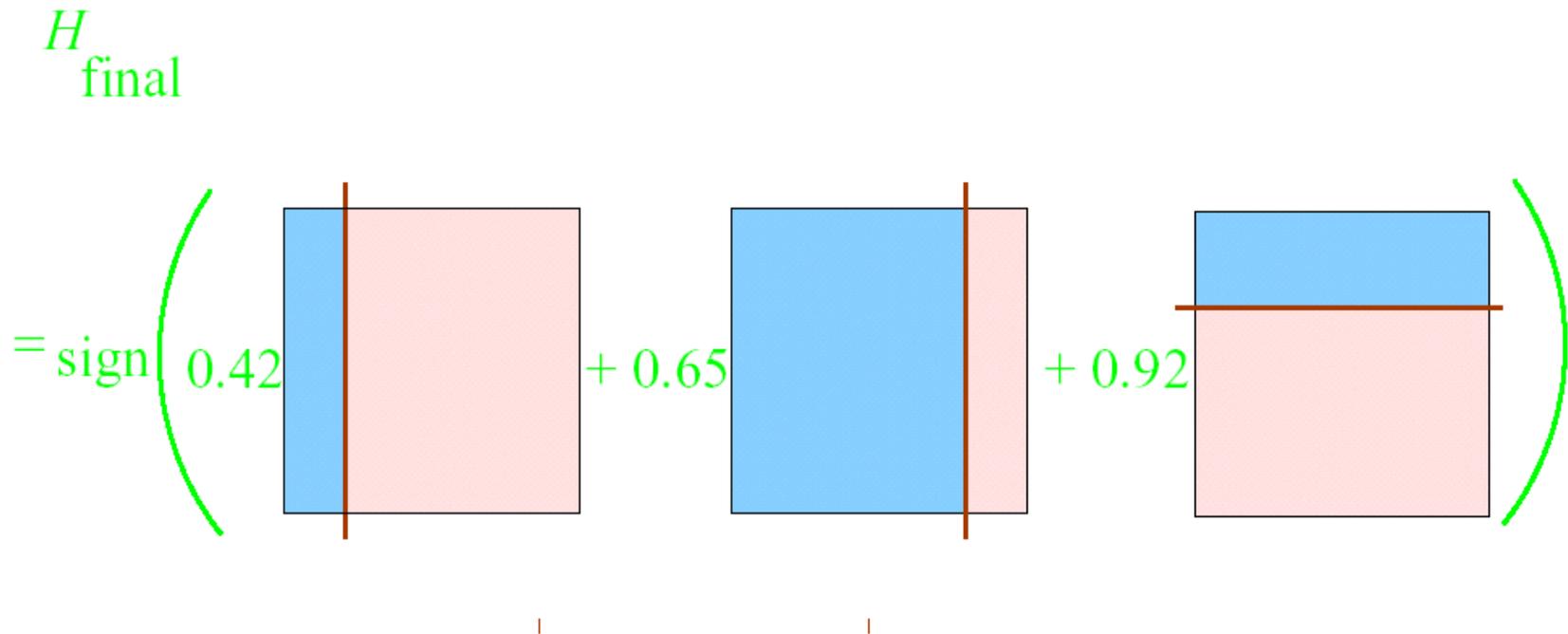


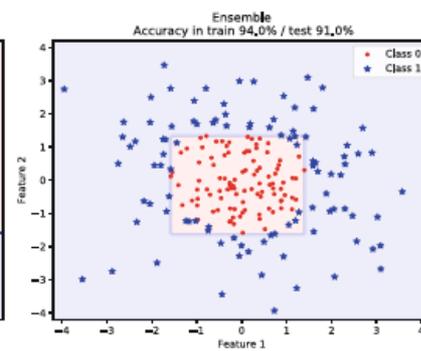
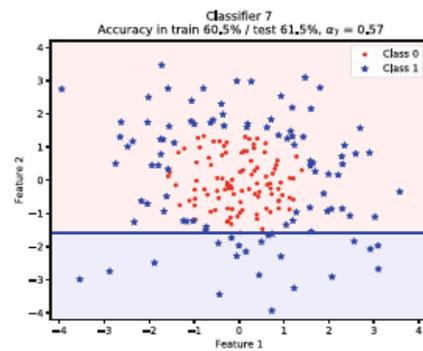
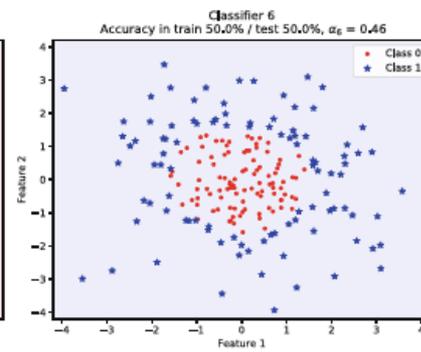
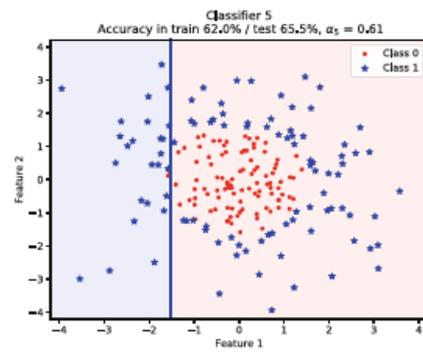
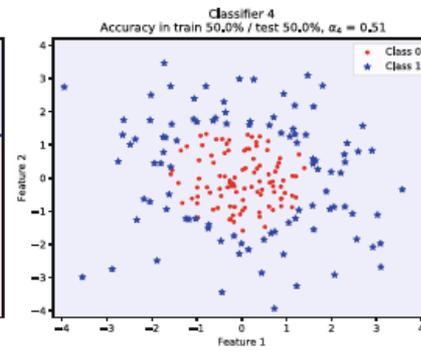
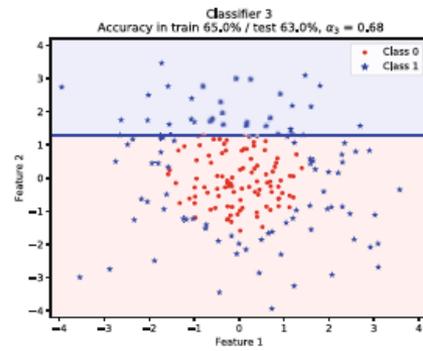
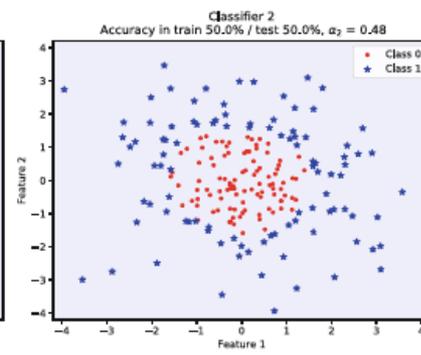
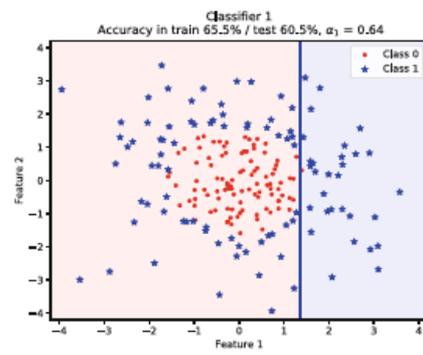
AdaBoost example

ROUND 3



AdaBoost example



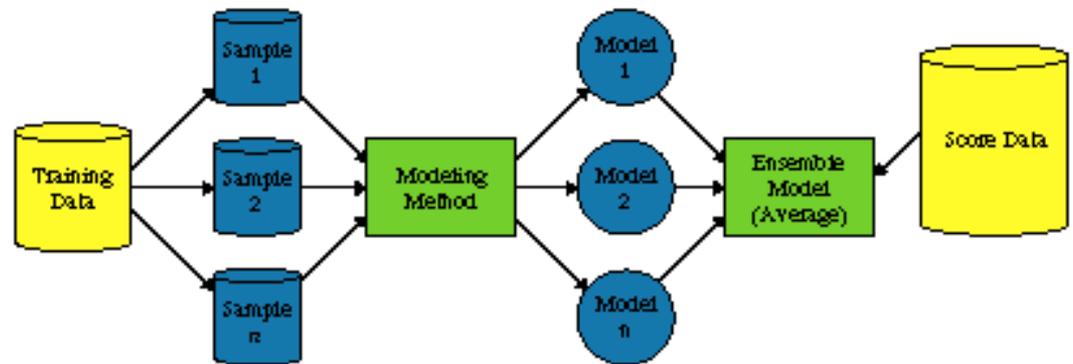


Bagging and Boosting

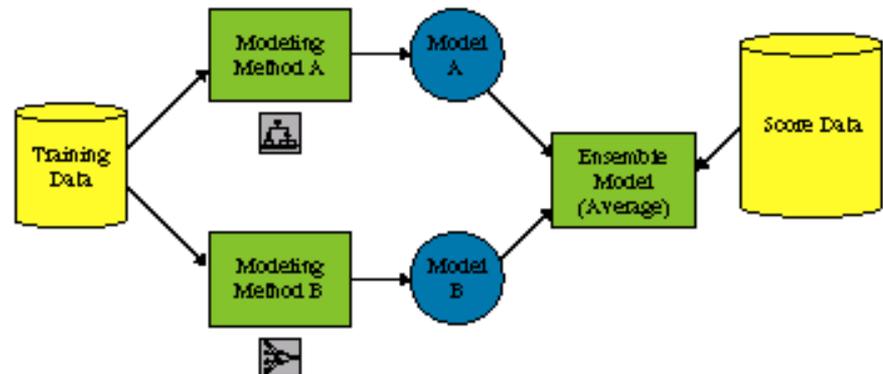
Bagging and Boosting

Resumen:

- Bagging =
Manipulation with data set

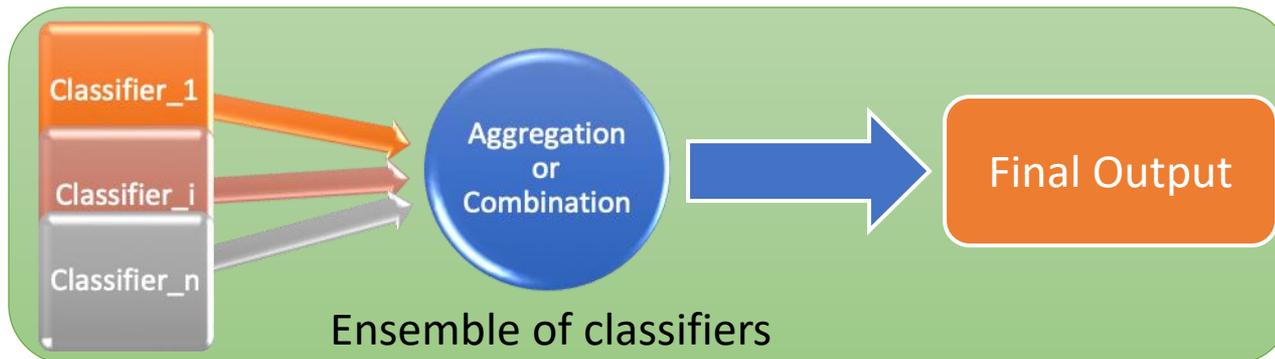


- Boosting =
Manipulation with model



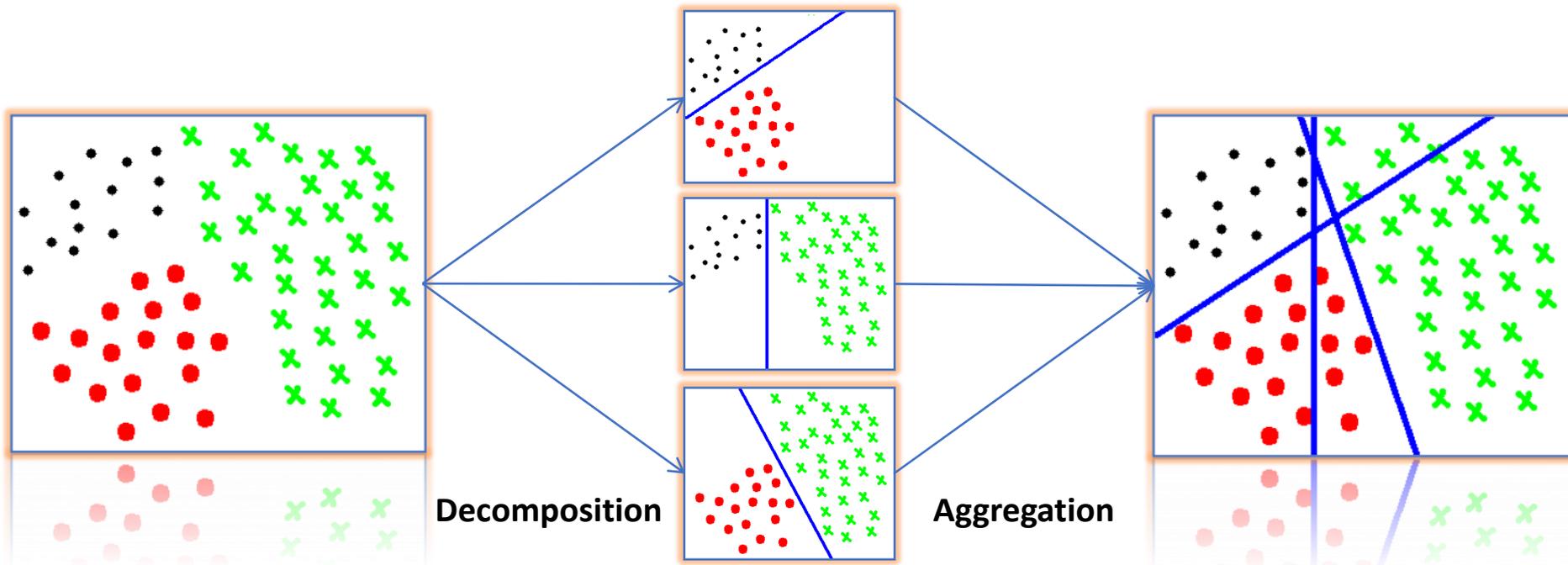
Binarization

- Decomposition of the multi-class problem
 - Divide and conquer strategy
 - Multi-class \rightarrow Multiple easier to solve binary problems
 - For each binary problem
 - 1 binary classifier = base classifier
 - Problem
 - How we should make the decomposition?
 - How we should aggregate the outputs?



Decomposition Strategies

- “One-vs-One” (OVO)
 - 1 binary problem for each pair of classes
 - Pairwise Learning, Round Robin, All-vs-All...
 - *Total = $m(m-1) / 2$ classifiers*



One-vs-One

- Disadvantages
 - Higher testing times (more classifiers)
 - Non-competent examples
- Many different aggregation proposals
 - From the score Matrix:

$$R = \begin{pmatrix} - & r_{12} & \cdots & r_{1m} \\ r_{21} & - & \cdots & r_{2m} \\ \vdots & & & \vdots \\ r_{m1} & r_{m2} & \cdots & - \end{pmatrix}$$

State-of-the-art on aggregation for OVO

- Voting strategy (VOTE)
 - Each classifier gives a vote for the predicted class
 - The class with the largest number of votes is predicted

$$Class = \arg \max_{i=1, \dots, m} \sum_{1 \leq j \neq i \leq m} s_{ij}$$

- where s_{ij} is 1 if $r_{ij} > r_{ji}$ and 0 otherwise.

- Weighted voting strategy (WV)
 - WV = VOTE but weight = confidence

$$Class = \arg \max_{i=1, \dots, m} \sum_{1 \leq j \neq i \leq m} r_{ij}$$

Example of prediction

- Classify x , whose real class is c_1

$$\square R(x) = \begin{pmatrix} & c1 & c2 & c3 & c4 & c5 \\ c1 & - & 0,55 & 0,6 & 0,75 & 0,7 \\ c2 & 0,45 & - & 0,4 & 1 & 0,8 \\ c3 & 0,4 & 0,6 & - & 0,5 & 0,4 \\ c4 & 0,25 & 0,0 & 0,5 & - & 0,1 \\ c5 & 0,30 & 0,2 & 0,6 & 0,9 & - \end{pmatrix}$$

Example of prediction

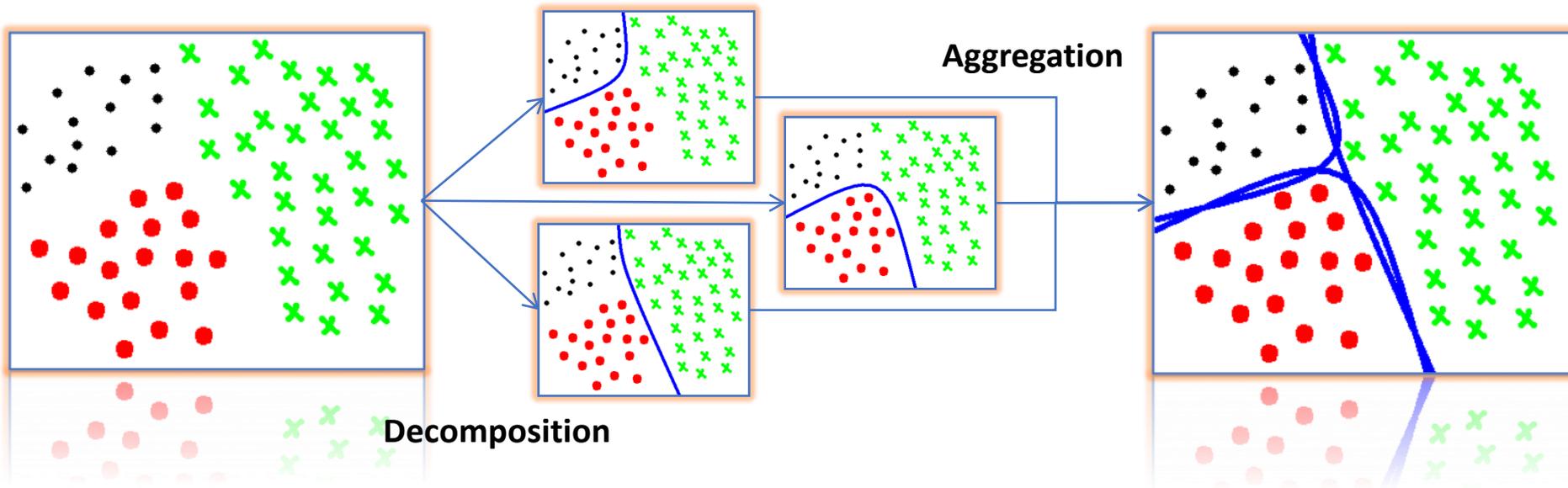
- Beware of **Non-Competent** Classifiers:

- Consider WV aggregation, c_2 is predicted
- **None** of the classifiers **considering c_1 failed**
- **Non-competent** classifiers **strongly voted for c_2**

$$\square R(x) = \begin{pmatrix} & c1 & c2 & c3 & c4 & c5 & WV \\ c1 & - & 0,55 & 0,6 & 0,75 & 0,7 & 2,6 \\ c2 & 0,45 & - & 0,4 & 1 & 0,8 & \mathbf{2,65} \\ c3 & 0,4 & 0,6 & - & 0,5 & 0,4 & 1,9 \\ c4 & 0,25 & 0,0 & 0,5 & - & 0,1 & 0,85 \\ c5 & 0,30 & 0,2 & 0,6 & 0,9 & - & 2,1 \end{pmatrix}$$

Decomposition Strategies

- “One-vs-All” (OVA)
 - 1 binary problem for each class
 - All instances in each problem
 - Positive class: instances from the class considered
 - Negative class: instances from all other classes
 - *Total = m classifiers*



One-vs-All

Advantages

- Less n^o of classifiers
- All classifiers are “competent”

Disadvantages

- Less studied in the literature
 - low n^o of aggregations
- More complex problems
- Imbalance training sets

State-of-the-art on aggregation for OVA

- Starting from the score-vector

$$R = (r_1, r_2, \dots, r_i, \dots, r_m)$$

- r_i = confidence of classifier in favor of class i
 - Respect to all other classes
- Usually more than 1 classifier predicts the positive class
 - Tie-breaking techniques
- Maximum confidence strategy (MAX)
 - Predicts the class with the largest confidence

$$Class = \arg \max_{i=1, \dots, m} r_i$$

Resumen

1. Redes Neuronales. Fundamentos y Deep Learning
2. Ensemble Learning
- 3. Calidad de datos. Preprocesamiento de Datos**
4. Big Data
5. Comentarios Finales

Calidad de datos

IN



=

OUT



Quality Data

- Sin datos de calidad, no hay modelos de calidad
- Las técnicas de **preprocesamiento de datos** generan datos de calidad (*quality data*) → modelos de calidad



Las decisiones de calidad deben basarse en datos de calidad

Preprocesamiento de Datos

Preprocesamiento: Tareas para disponer de datos de calidad previos al uso de algoritmos de extracción de conocimiento.

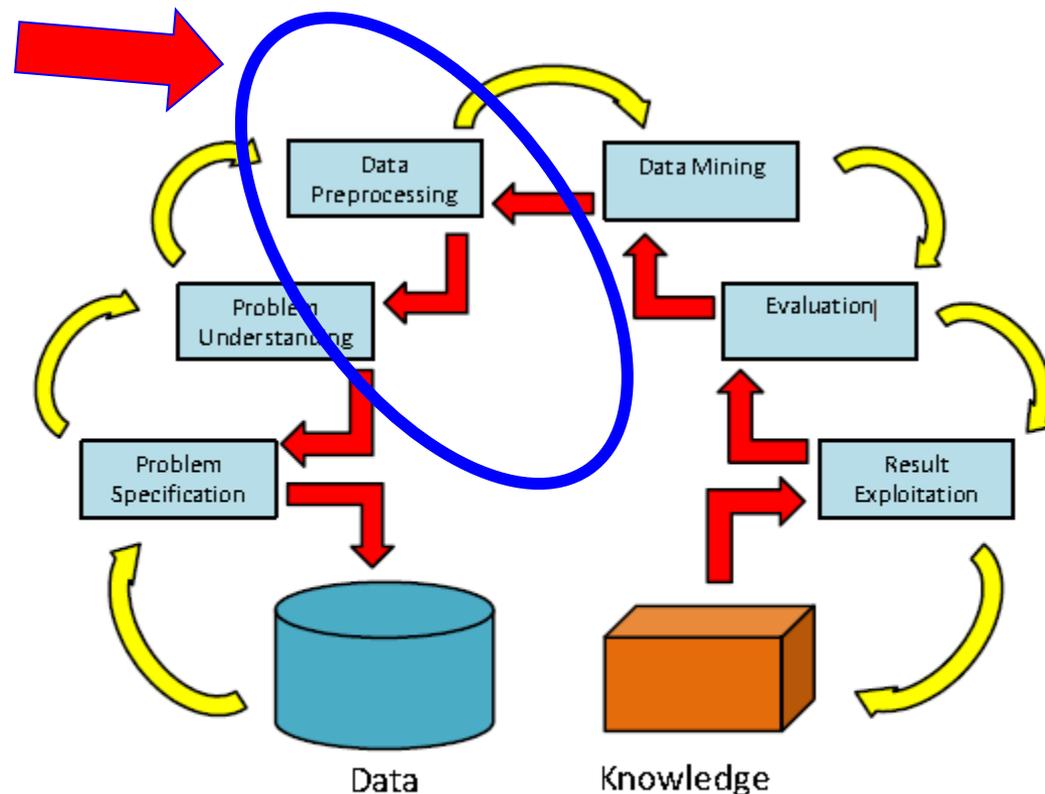


Fig. 1.1: KDD process.

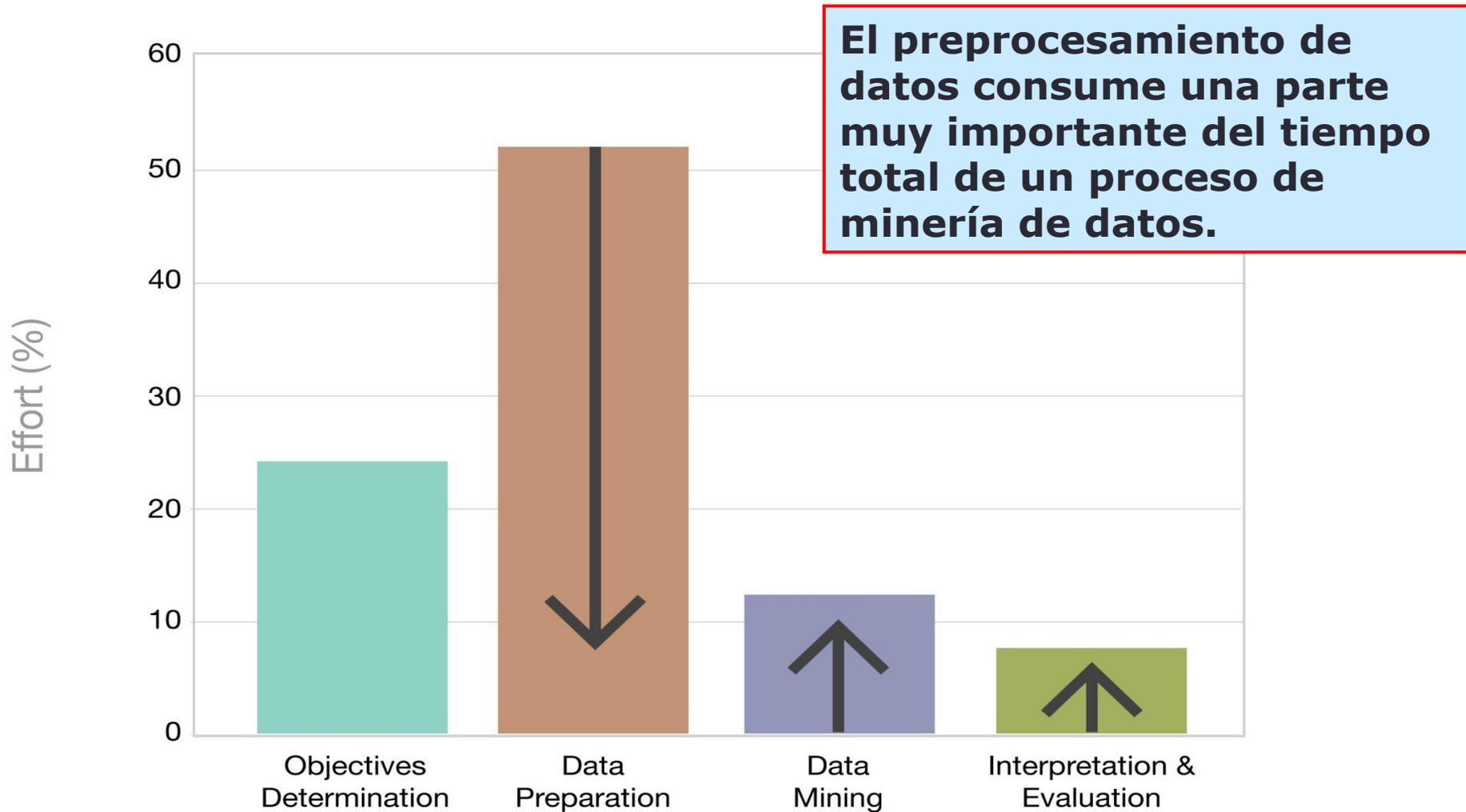
Preprocesamiento de Datos

1. Introducción. Preprocesamiento
2. Integración, Limpieza y Transformación
3. Datos Imperfectos
4. Reducción de Datos
5. Comentarios Finales

Bibliografía.

S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining
Springer, 2015

INTRODUCCIÓN



Preprocesamiento de Datos

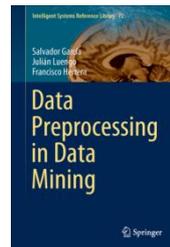
¿Qué incluye el Preprocesamiento de Datos?

S. García, J. Luengo, F. Herrera, 2015, Preface vii:

“Data preprocessing includes data preparation, compounded by integration, cleaning, normalization and transformation of data; and data reduction tasks; such as feature selection, instance selection, discretization, etc.

...

The result expected after a reliable chaining of data preprocessing tasks is a final dataset, which can be considered correct and useful for further data mining algorithms.”



S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining
Springer, 2015

Preprocesamiento de Datos

¿Qué incluye el Preprocesamiento de Datos?

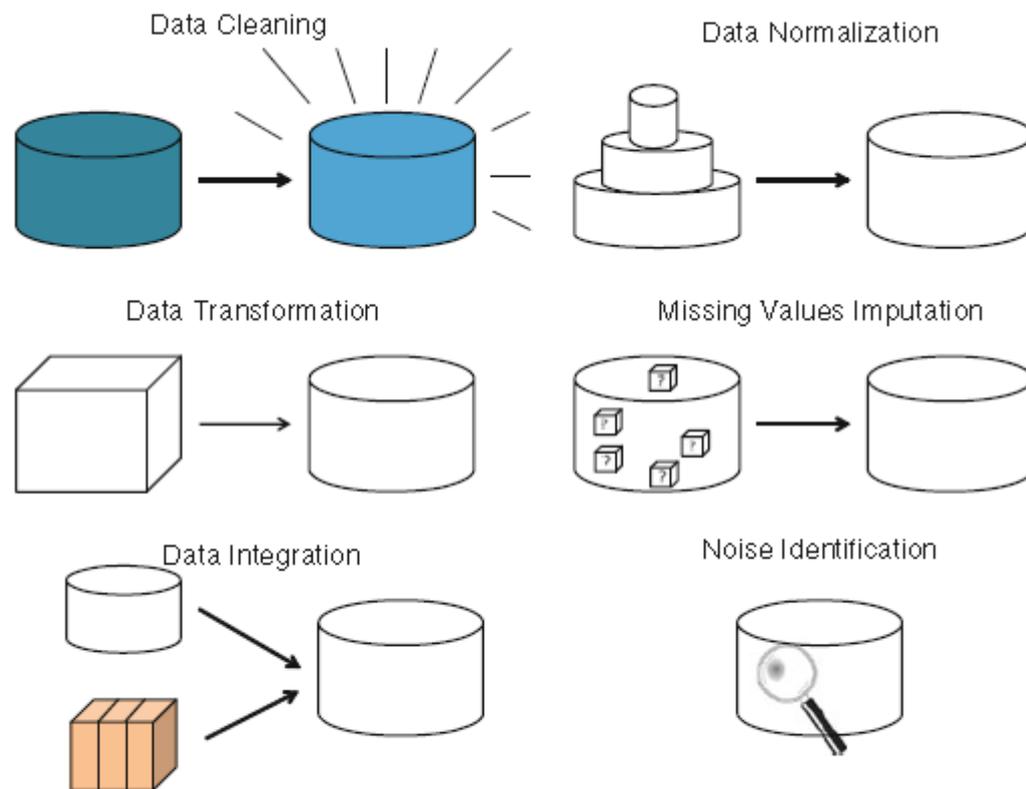


Fig. 1.3 Forms of data preparation

Preprocesamiento de Datos

¿Qué incluye el Preprocesamiento de Datos?

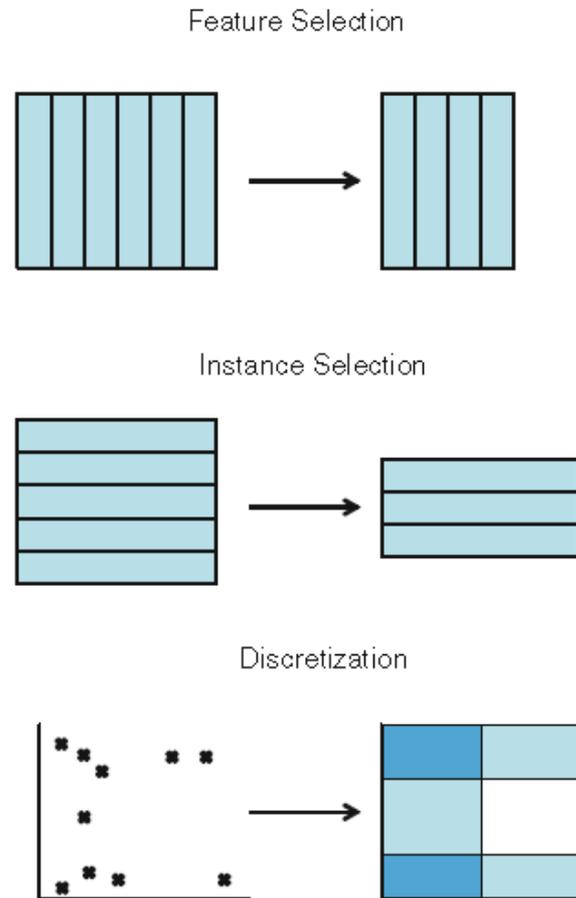
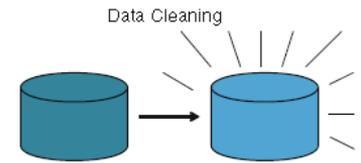


Fig. 1.4 Forms of data reduction

Preprocesamiento de Datos

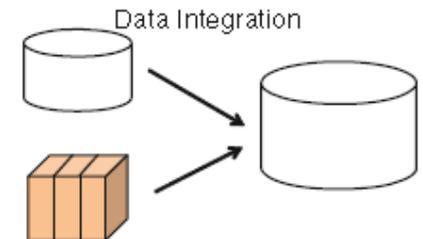
Data Preparation

- How do I clean up the data?—**Data Cleaning**



- How do I incorporate and adjust data?

—**Data Integration**

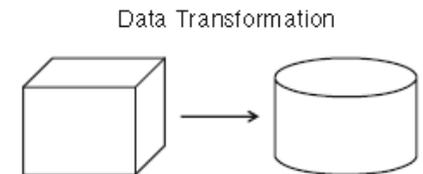


How do I provide accurate data?

—**Data Transformation**

- How do I unify and scale data?

—**Data Normalization**

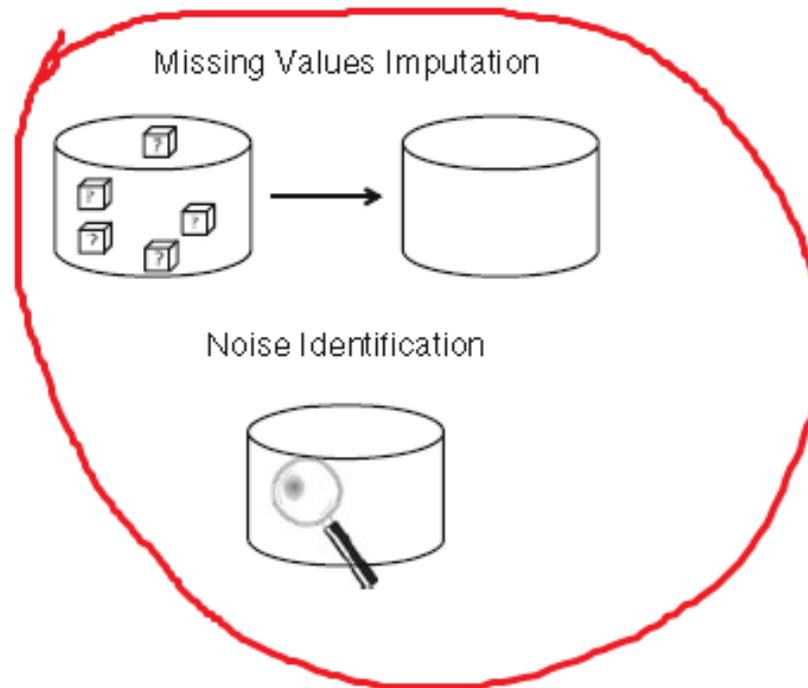


Integración, Limpieza y Transformación

Preprocesamiento de Datos

Data Preparation

- How do I handle imperfect data?— **Imputation, filtering, relabeling, ...**



Valores perdidos

		Attributes						
		1	2	3	4	5	...	m
Instances	1					?		
	2			?				
	3		?		?			
	4							
	5							
	6						?	
	7			?		?		
	8							
	9							
	10			?			?	
	11		?					
	⋮				?			
	n							?

Information Sources

Attributes

Class

Att 1	Att 2	Class
0.25	red	positive
0.25	red	negative
0.99	green	negative
1.02	green	positive
2.05	?	negative
=	green	positive

Att. Noise

Class Noise

Kinds of Noise

Limpieza de datos con ruido

Tipos de ejemplos

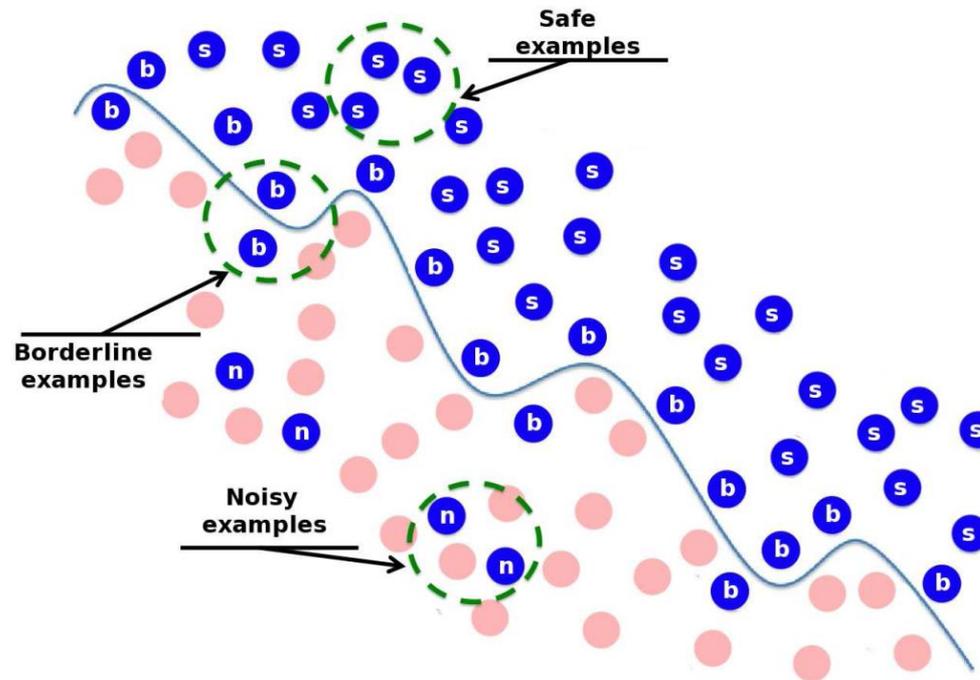


Fig. 5.2 The three types of examples considered in this book: safe examples (labeled as *s*), *borderline* examples (labeled as *b*) and *noisy* examples (labeled as *n*). The *continuous line* shows the decision boundary between the two classes

Limpieza de datos con ruido

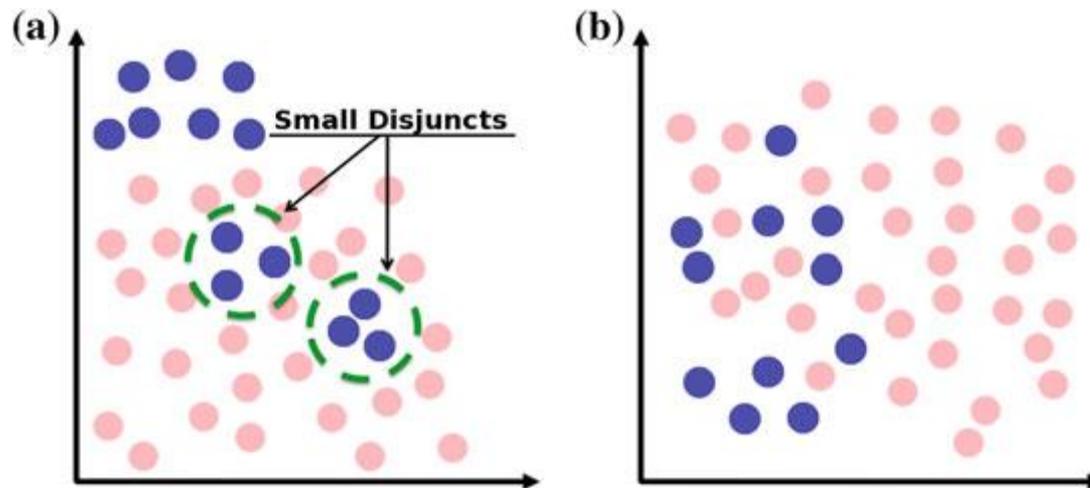


Fig. 5.1 Examples of the interaction between classes: a) small disjuncts and b) overlapping between classes

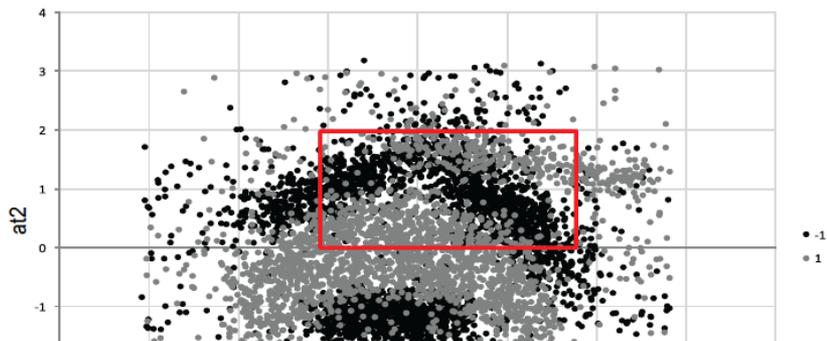
Limpieza de datos con ruido

Software en R:

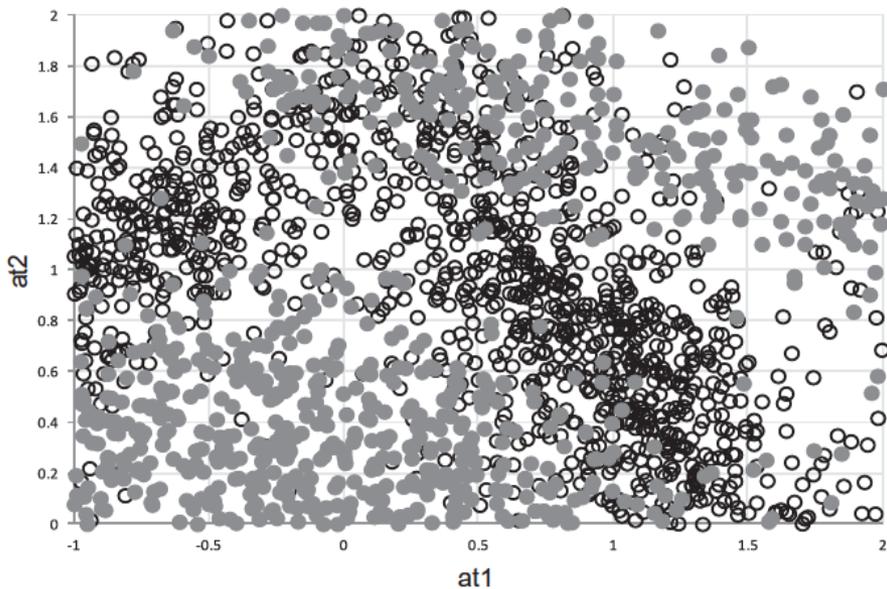


NoiseFiltersR

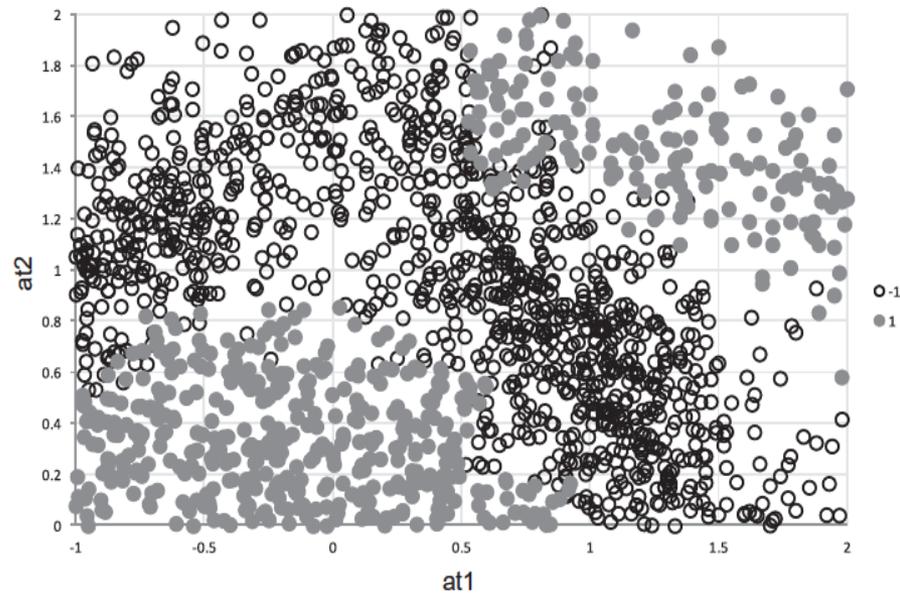
Banana 10% noise



Banana 10% noise



IPF

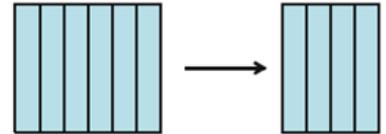


Reducción de Datos

- ✿ Diferentes vías para la Reducción de Datos:
 - ❖ Selección/Extracción de Características
 - ❖ Selección de Instancias
 - ❖ Discretización

Reducción de Datos

- How do I reduce the dimensionality of data?
— **Feature Selection**

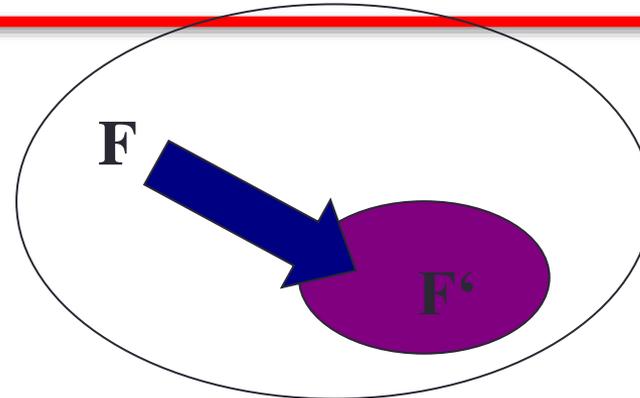


- How do I remove redundant and/or conflictive examples?
— **Instance Selection**
(prototype Selection vs training set selection)



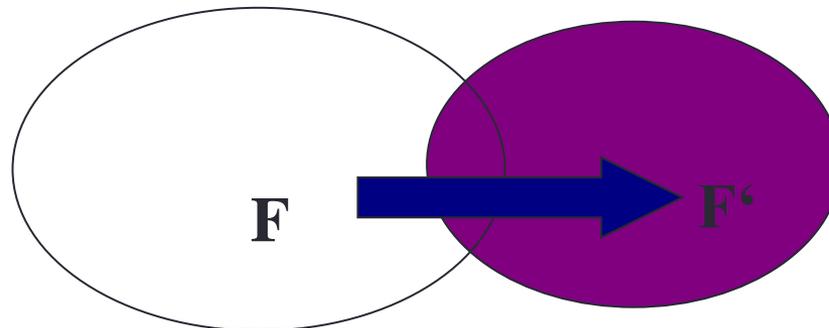
Feature Selection / - Extraction

- Feature Selection:



$$\{f_1, \dots, f_i, \dots, f_n\} \xrightarrow{f.\textit{selection}} \{f_{i_1}, \dots, f_{i_j}, \dots, f_{i_m}\} \quad \begin{array}{l} i_j \in \{1, \dots, n\}; j = 1, \dots, m \\ i_a = i_b \Rightarrow a = b; a, b \in \{1, \dots, m\} \end{array}$$

- Feature Extraction/Creation



$$\{f_1, \dots, f_i, \dots, f_n\} \xrightarrow{f.\textit{extraction}} \{g_1(f_1, \dots, f_n), \dots, g_j(f_1, \dots, f_n), \dots, g_m(f_1, \dots, f_n)\}$$

Selección de Características

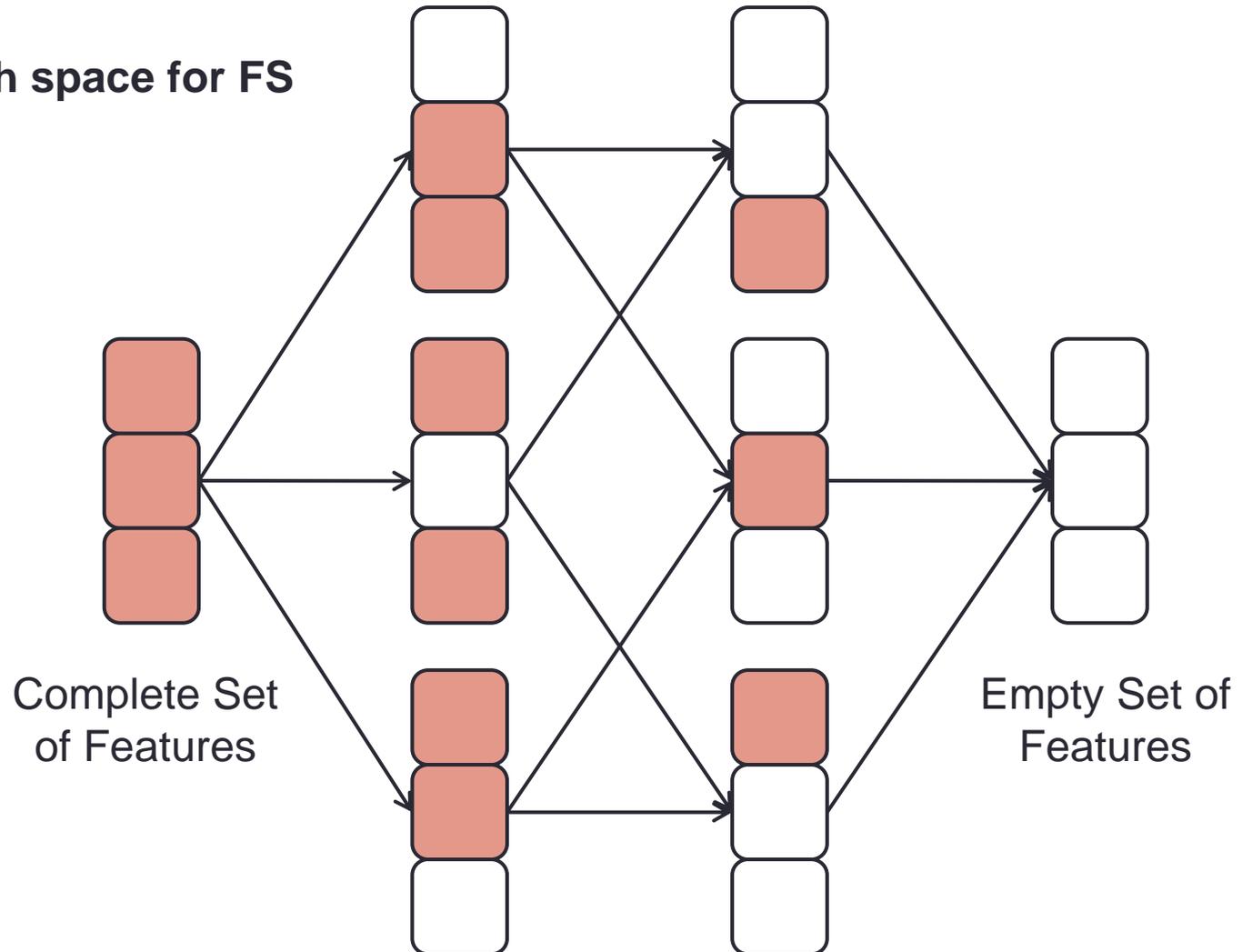
El problema de la selección de características (SC) o variables (*Feature Subset Selection, FSS*) consiste en encontrar un subconjunto de las variables del problema que optimice la probabilidad de clasificar correctamente

¿Por qué es necesaria la selección de variables?

- Más atributos no significa más éxito en la clasificación
- Trabajar con menos variables reduce la complejidad del problema y disminuye el tiempo de ejecución
- Con menos variables la capacidad de generalización aumenta
- Los valores para ciertos atributos pueden ser costosos de obtener

Selección de Características

Fig. 7.1 Search space for FS



Selección de Características

- En un algoritmo de selección de características se distinguen dos componentes principales
 - Una estrategia de búsqueda para seleccionar subconjuntos candidatos
 - Una función objetivo que evalúe esos subconjuntos
- **Estrategia de búsqueda**
 - Dadas N variables, explorar todos los subconjuntos posibles supone 2^N (p.e. $2^{20}=1048576$)
 - Si queremos exactamente subconjuntos de M variables ($M \leq N$) entonces supone $\binom{N}{M}$. P.e. explorar subconjuntos de 10 variables de 20 posibles, daría 184756
 - Una búsqueda exhaustiva no es aceptable
- **Función objetivo:** evaluar la bondad del subconjunto seleccionado

Selección de Características

Funciones objetivo: Se distinguen dos enfoques distintos

- **Filtro (*filter*).** La función objetivo evalúa los subconjuntos basándose en la información que contienen. Se utiliza como función objetivo medidas de separabilidad de clases, de dependencias estadísticas, basadas en teoría de la información,...)
- **Envolvente (*wrapper*).** La función objetivo consiste en aplicar la técnica de aprendizaje que se utilizará finalmente sobre la proyección de los datos al conjunto de variables candidato. El valor devuelto suele ser el porcentaje de acierto del clasificador construido

Selección de Características

Ventajas

■ Envoltentes:

- Exactitud: generalmente son más exactos que los filtro, debido a la interacción entre el clasificador y el conjunto de datos de entrenamiento
- Capacidad para generalizar: poseen capacidad para evitar el sobreajuste debido a las técnicas de validación utilizadas

■ Filtro:

- Rápidos. Suelen limitarse a cálculos de frecuencias, mucho más rápido que entrenar un clasificador
- Generalidad. Al evaluar propiedades intrínsecas de los datos y no su interacción con un clasificador, sus resultados pueden ser utilizados por cualquier clasificador

Selección de Características

Inconvenientes

■ Envoltentes:

- Muy costosos: para cada evaluación hay que aprender un modelo y validarlo. No es factible para clasificadores costosos
- Pérdida de generalidad: La solución está sesgada hacia el clasificador utilizado

■ Filtros:

- Tendencia a incluir muchas variables. Normalmente se debe a las características monótonas de la función objetivo utilizada
 - El usuario deberá seleccionar un umbral

Selección de Características

Algunos algoritmos relevantes:

- **Focus algorithm.** Consistency measure for forward search,
- Mutual Information based Features Selection (MIFS).
- Las Vegas Filter (LVF)
- Las Vegas Wrapper (LVW)
- Relief Algorithm
- mRMR

Software en R:



Fselector (varias formas de selección tanto filter como wrapper)

Caret

Boruta

Reducción de la Dimensionalidad

La “Maldición” de la Dimensionalidad

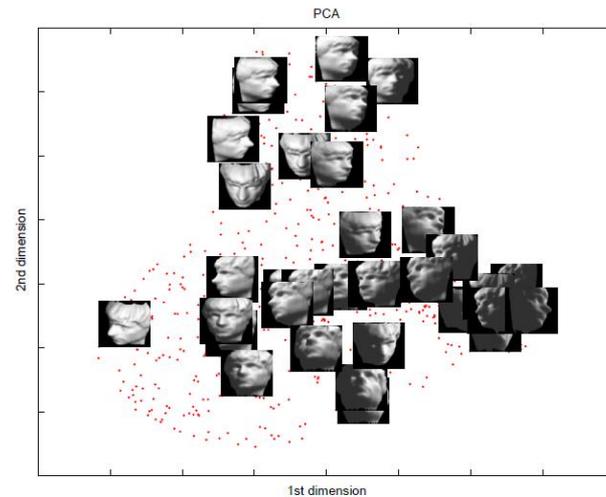
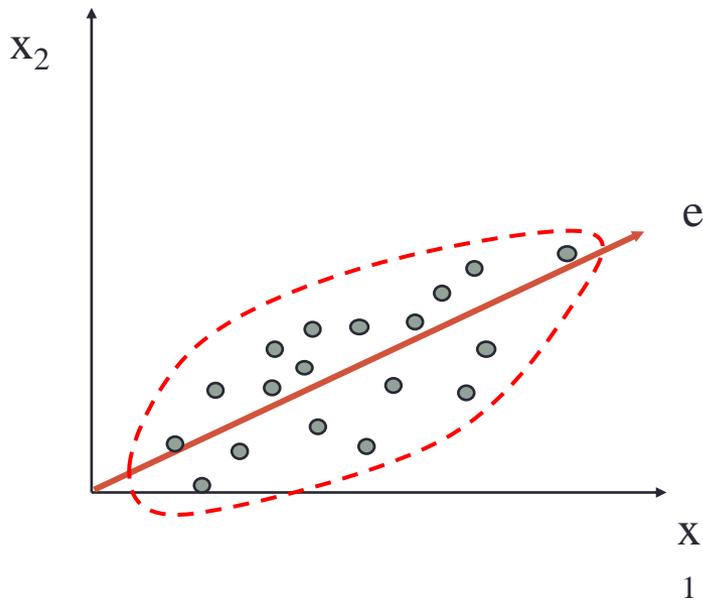
La dimensionalidad de los datos llega a ser un obstáculo serio para la eficiencia de la mayoría de los algoritmos de aprendizaje y minería de datos.

Se estima que conforme el número de dimensionales (variables) crece, el tamaño de la muestra de datos requiere también crecer exponencialmente para tener un estimador efectivo de las densidades multivariadas.

Reducción de la Dimensionalidad: Se responsabiliza de la reducción del número de variables en los conjuntos de datos evitando importantes pérdidas de información.

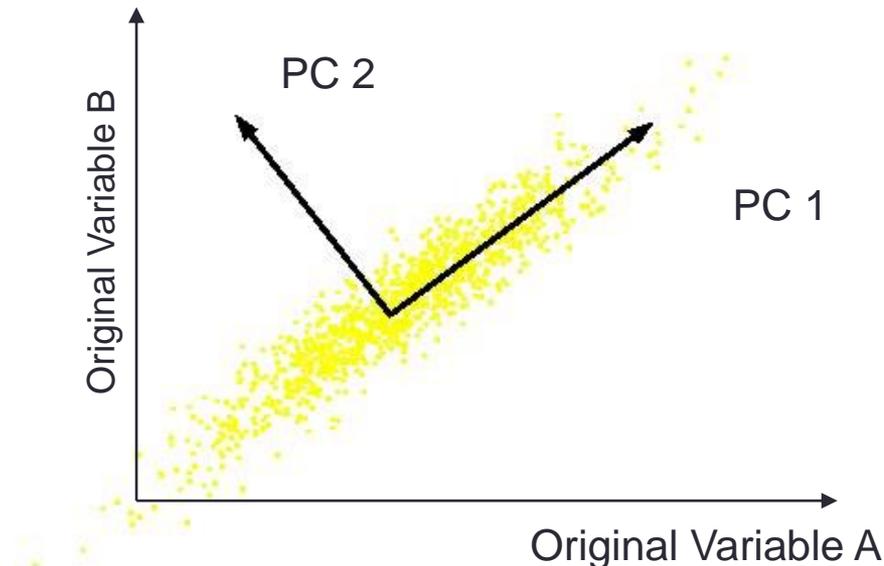
Principal Component Analysis (PCA)

- Historia: Karl Pearson, 1901
- Encontrar las proyecciones que capturen las mayores cantidades de variación en los datos
- Encontrar los autovectores de la matriz de covarianza, y utilizarlos para definir el nuevo espacio,

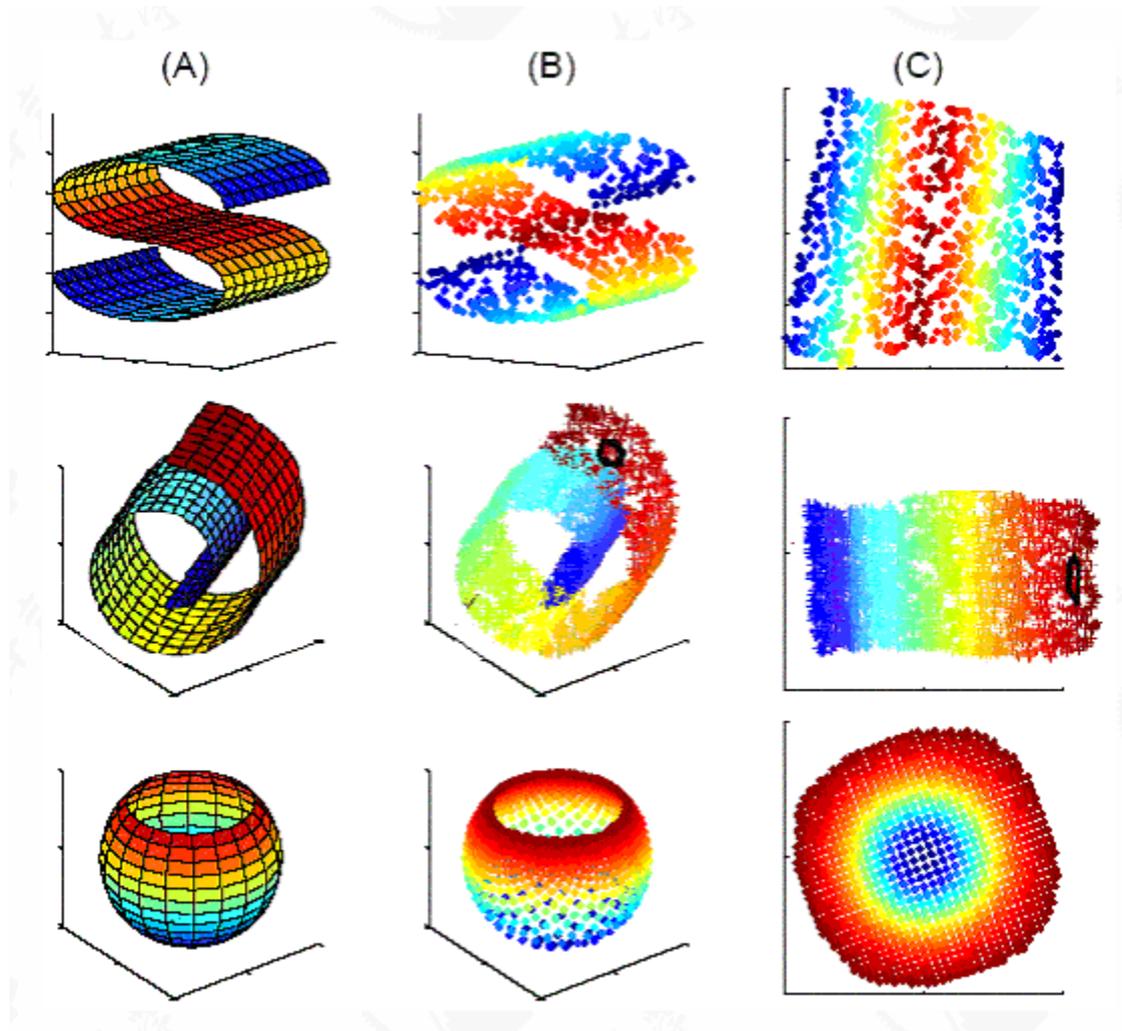


PCA

- Definición: Dado un dataset, encontrar los ejes principales que serán aquellos ejes ortogonales en los que la varianza observada en la proyección es máxima.



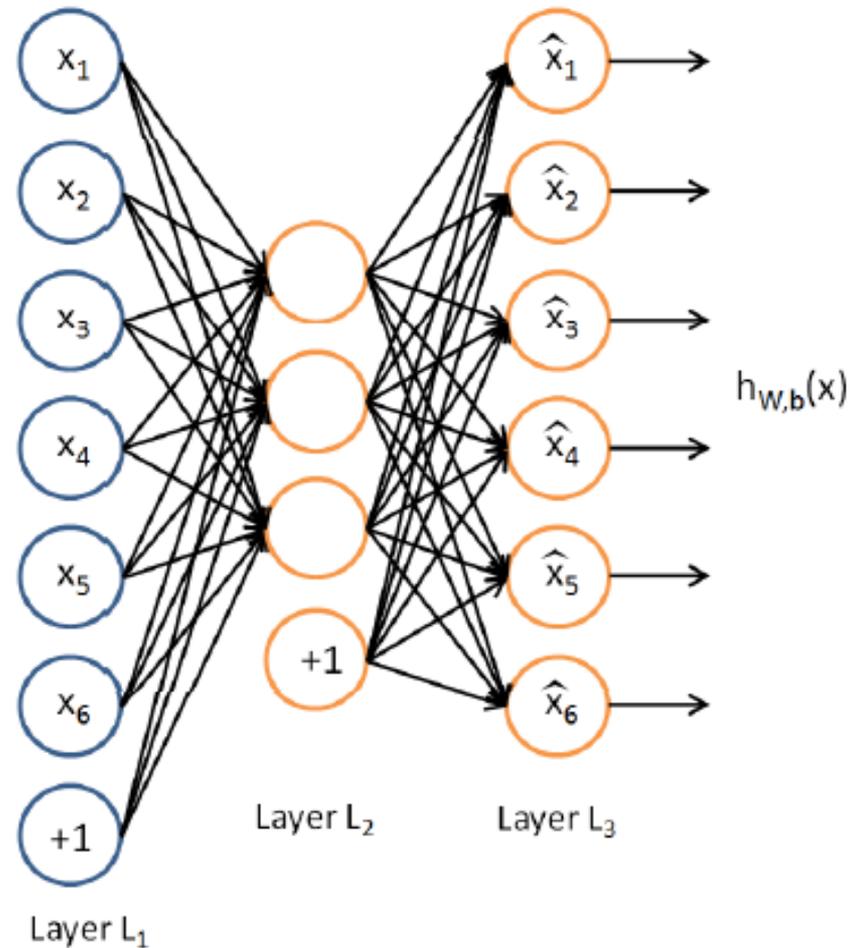
LLE Example



AutoEncoders

An **autoencoder** neural network is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs.

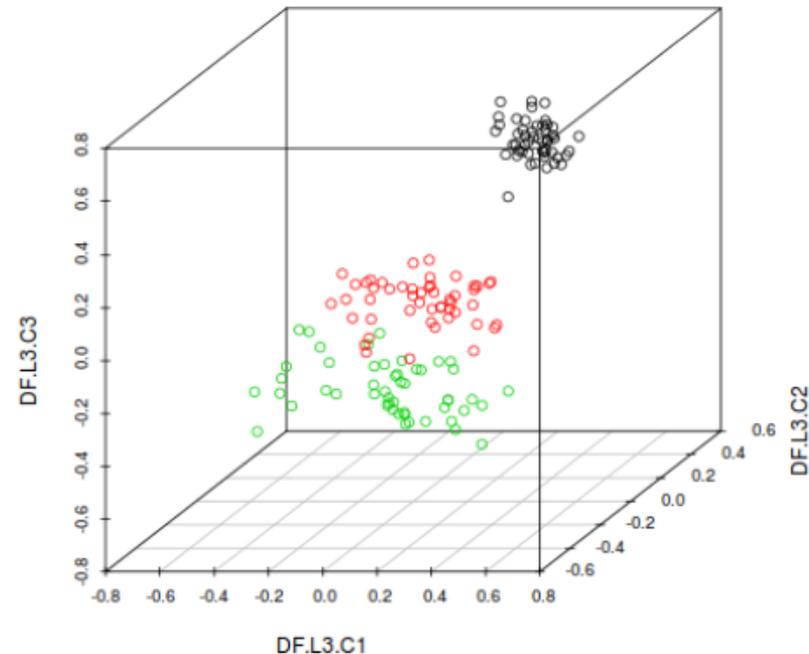
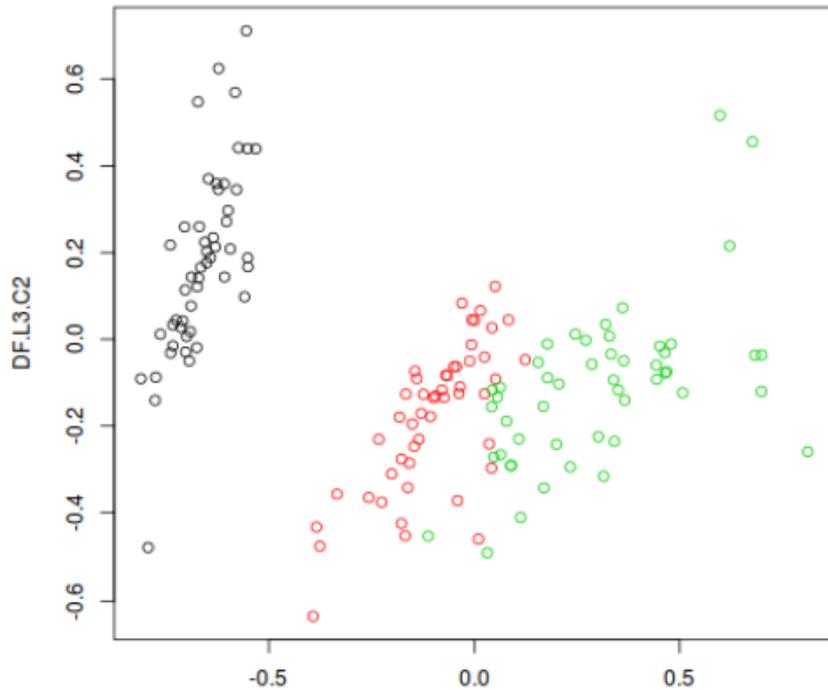
The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for the purpose of dimensionality reduction.



AutoEncoders

- **Autoencoder** (autoencoder de h2o, salida de la capa intermedia capas internas de [8, 5, 3, 5, 8] neuronas y 100 "epochs" (el tridimensional), y [8, 5, 2, 5, 8] neuronas con 1000 "epochs" (el bidimensional)).

setosa, versicolor y virginica

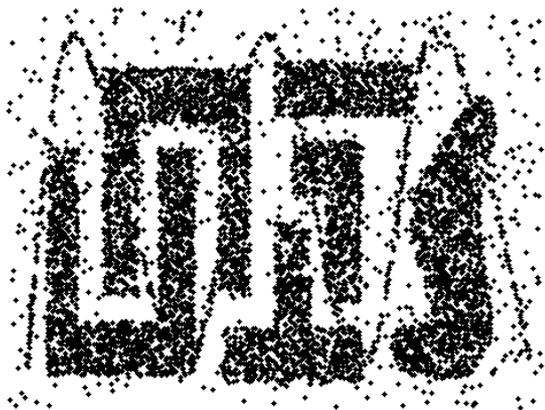


Selección de Instancias

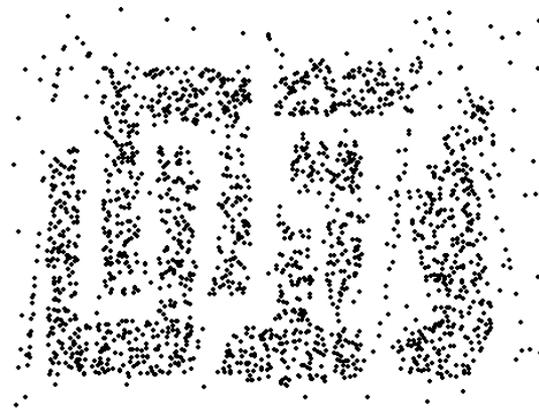
- ✿ La SI pretende elegir los ejemplos que sean relevantes para una aplicación y lograr el máximo rendimiento. El resultado de la SC sería:
 - ❖ Menos datos → los algoritmos pueden aprender más rápidamente
 - ❖ Mayor exactitud → el clasificador generaliza mejor
 - ❖ Resultados más simples → más fácil de entender
- ✿ SI y Transformación (compactación/agrupamiento)

Selección de Instancias

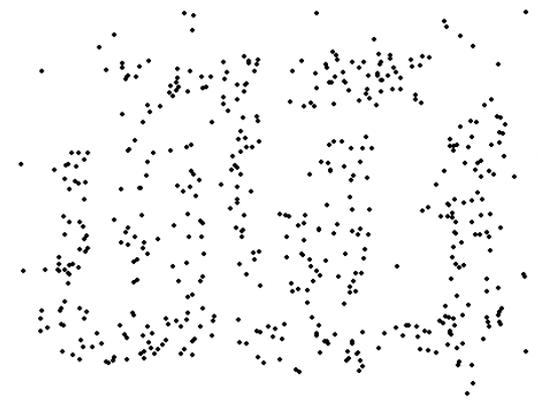
Ejemplos de diferentes tamaños



8000 puntos



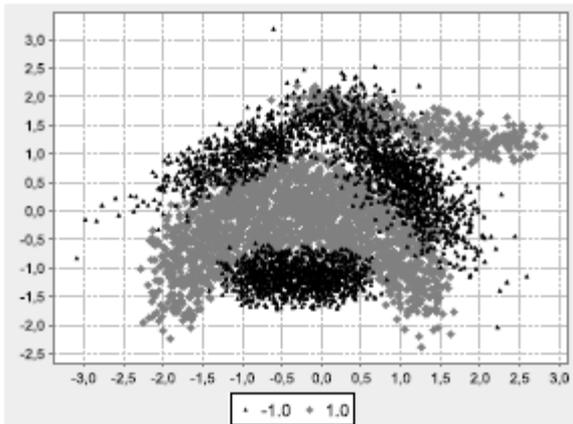
2000 puntos



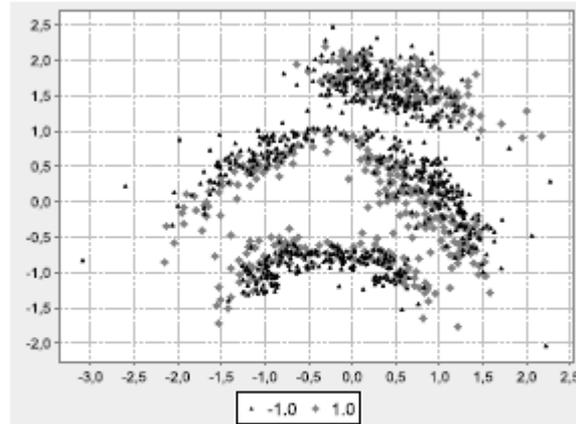
500 puntos

Selección de Instancias

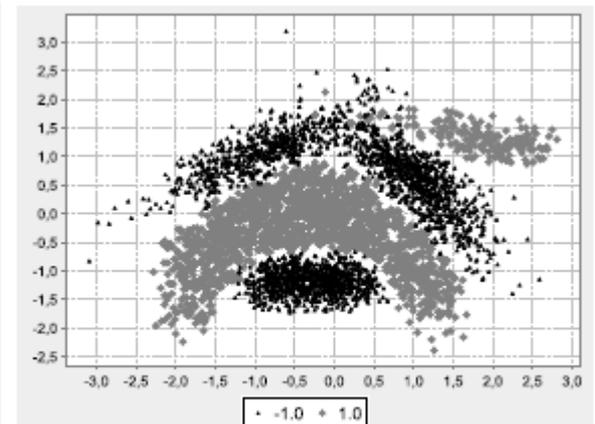
Ejemplos gráficos:



(a) Banana Original
(0.8751, 0.7476)



(b) CNN (0.7729, 0.8664, 0.7304)

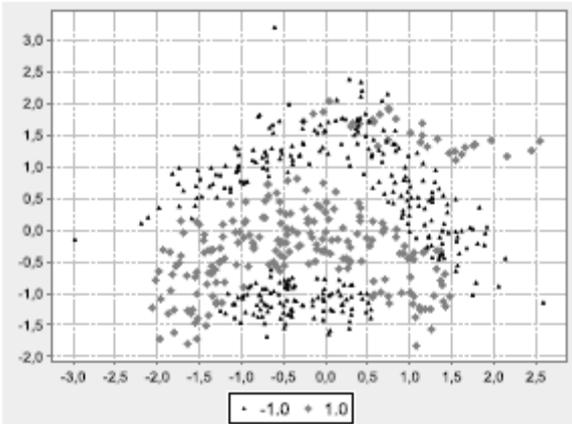


(h) AllKNN (0.1758, 0.8934, 0.7831)

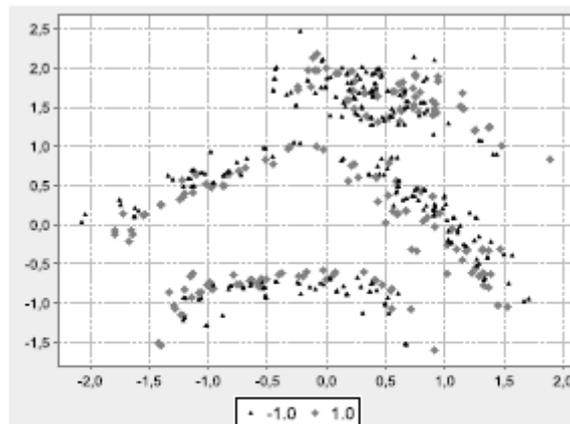
Conjunto banana con 5.300 instancias y dos clases. Conjunto obtenido por CNN y AllKNN (aplicación iterativa de ENN con $k=3, 5$ y 7).

Selección de Instancias

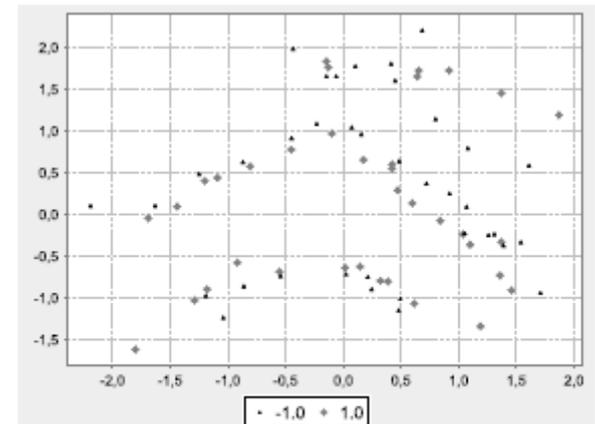
Ejemplos gráficos:



(k) RMHC (0.9000, 0.8972, 0.7915)



(e) DRO3
(0.9151, 0.8696, 0.7356)



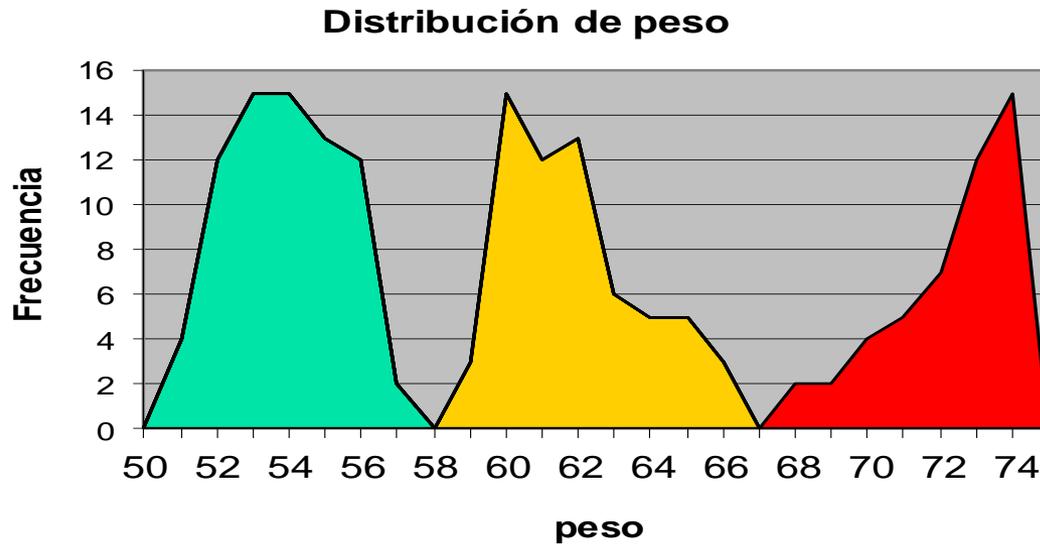
(l) SSMA (0.9879, 0.8964, 0.7900)

RMHC es una técnica de muestreo adaptativo basa en búsquedas locales con un tamaño final fijo.

DRO3 es la técnica híbrida más conocida y utilizada para NN.

SSMA es una aproximación evolutiva basada en algoritmo meméticos.

Discretización



50 - 58 kg
59-67 kg
> 68 kg

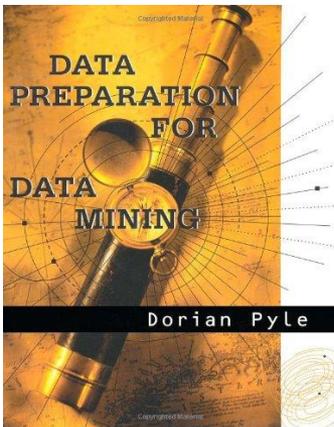
Comentarios Finales

Una ventaja: El preprocesamiento de datos permite aplicar los modelos de Aprendizaje/Minería de Datos de forma más rápida y sencilla, obteniendo modelos/patrones de más calidad: precisión e/o interpretabilidad.

Un inconveniente: El preprocesamiento de datos no es un área totalmente estructurada con una metodología concreta de actuación para todos los problemas. Cada problema puede requerir una actuación diferente, utilizando diferentes herramientas de preprocesamiento.

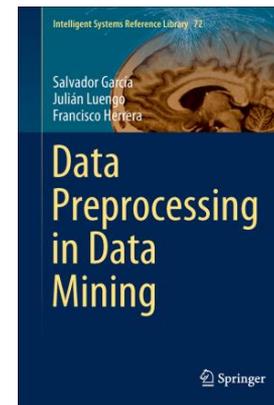
Bibliografía

Bibliografía –Preprocesamiento



Dorian Pyle
Morgan Kaufmann, Mar 15, 1999

S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining
Springer, 15, 2015

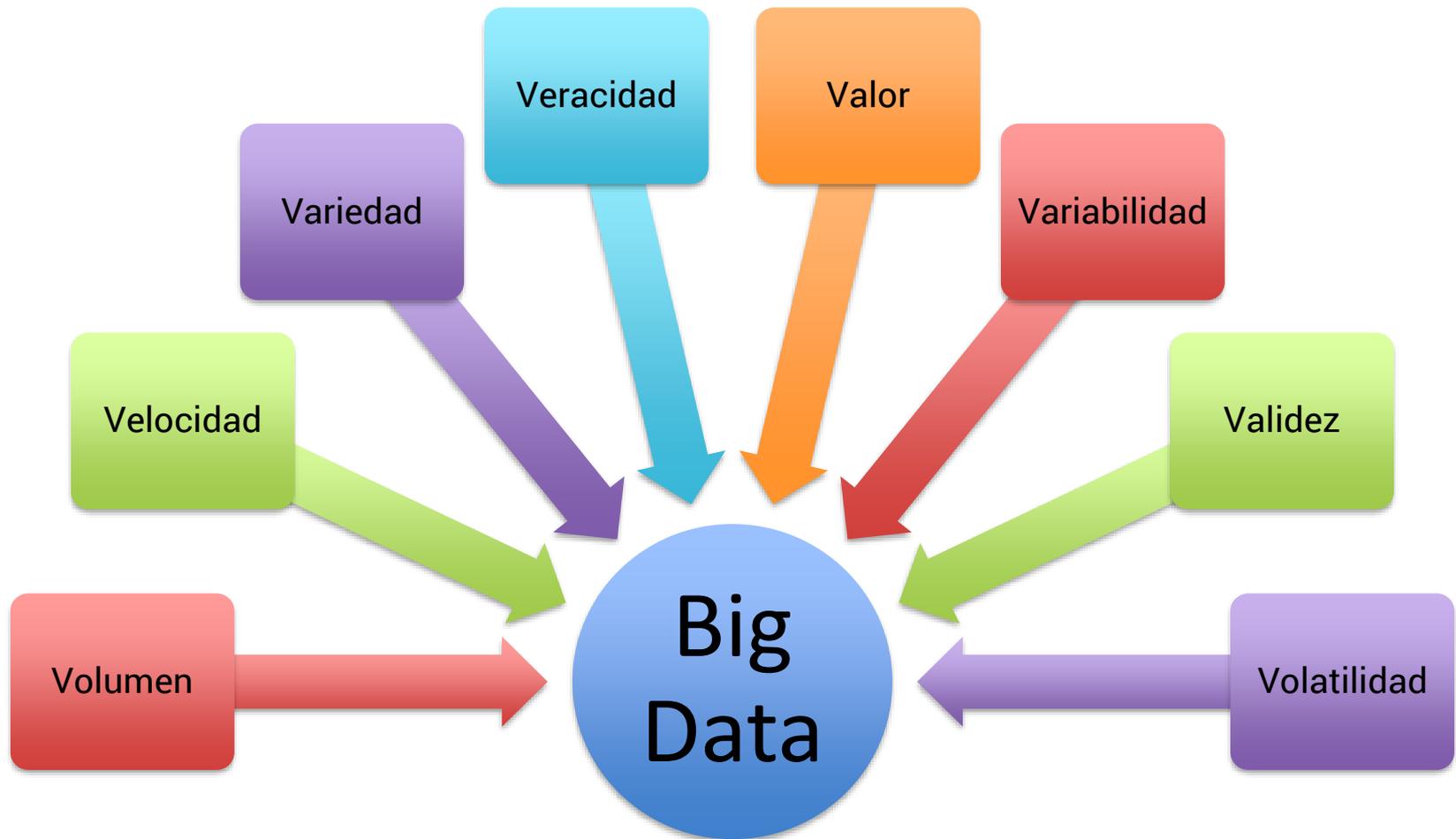


Resumen

1. Redes Neuronales. Fundamentos y Deep Learning
2. Ensemble Learning
3. Calidad de datos. Preprocesamiento de Datos
- 4. Big Data**
5. Comentarios Finales

¿Qué es Big Data?

Las 8 V's de Big Data



¿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 ...



¿Por qué Big Data?

- **Problema:** Escalabilidad de grandes cantidades de datos
- **Ejemplo:**
 - Exploración 100 TB en 1 nodo @ 50 MB/sec = 23 días
 - Exploración en un clúster de 1000 nodos = 33 minutos
- **Solución → Divide-Y-Vencerás**



Una sola máquina no puede gestionar grandes volúmenes de datos de manera eficiente

¿Por qué Big Data?

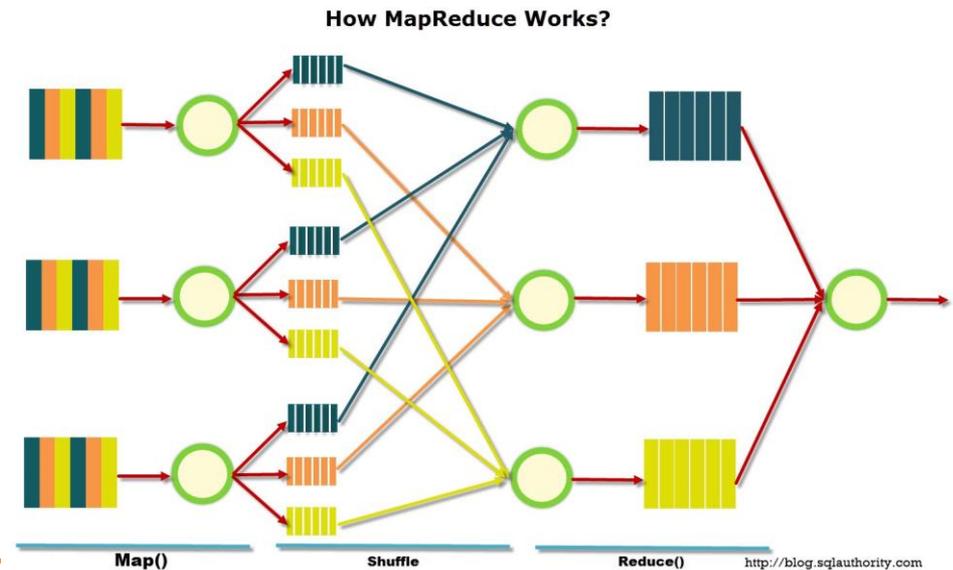
- **Problema:** Escalabilidad de grandes cantidades de datos
- **Ejemplo:**
 - Exploración 100 TB en 1 nodo @ 50 MB/sec = 23 días
 - Exploración en un clúster de 1000 nodos = 33 minutos
- **Solución → Divide-Y-Vencerás**

¿Cómo podemos procesar 1000
TB or 10000 TB?

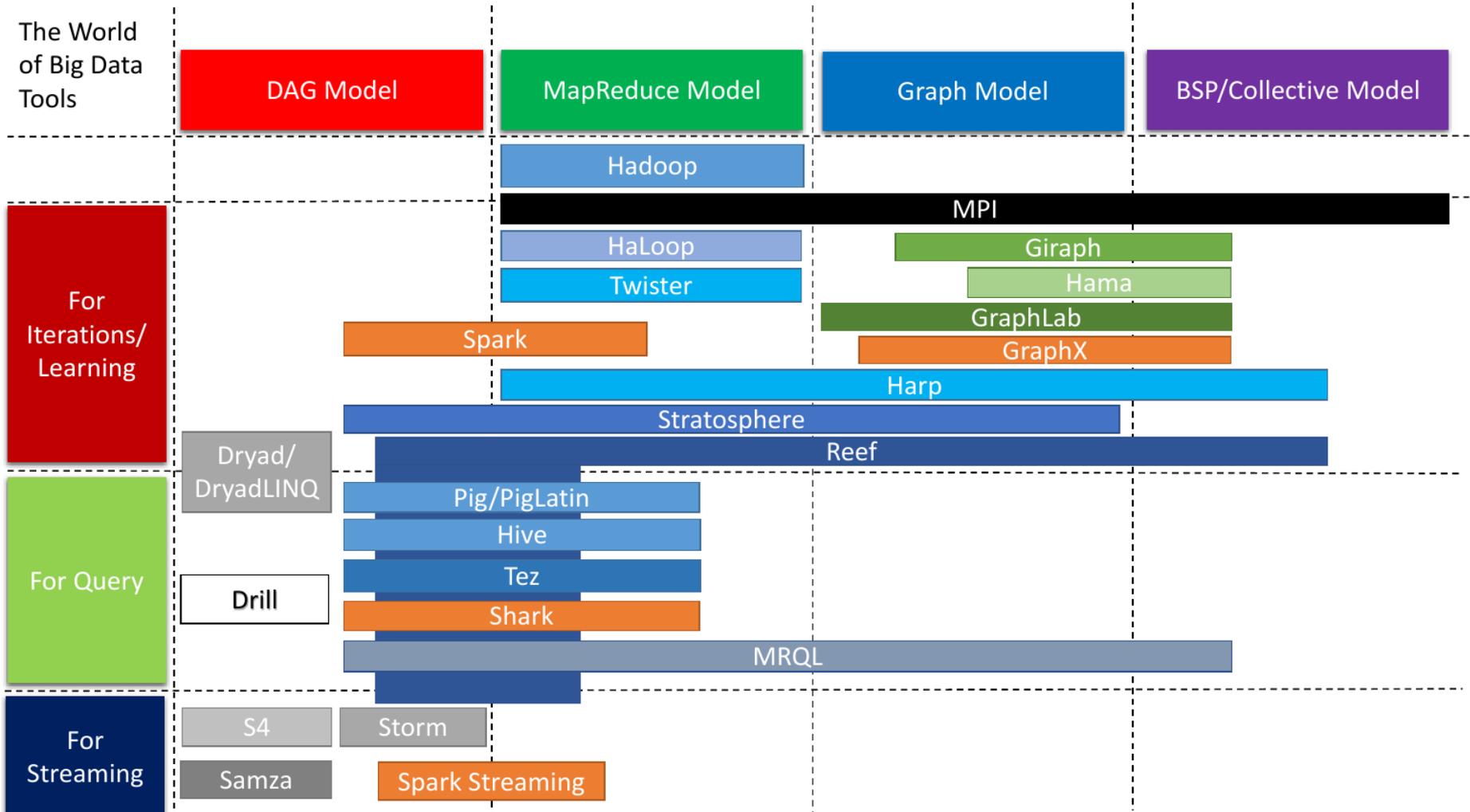


MapReduce

- MapReduce es el entorno más popular para Big Data
- Basado en la estructura Valor-llave.
- Dos operaciones:
 1. **Función Map** : Procesa bloques de información
 2. **Función Reduce**: Fusiona resultados previous de acuerdo a su llave.
- + Una etapa intermedia de agrupamiento por llave (**Shuffling**)

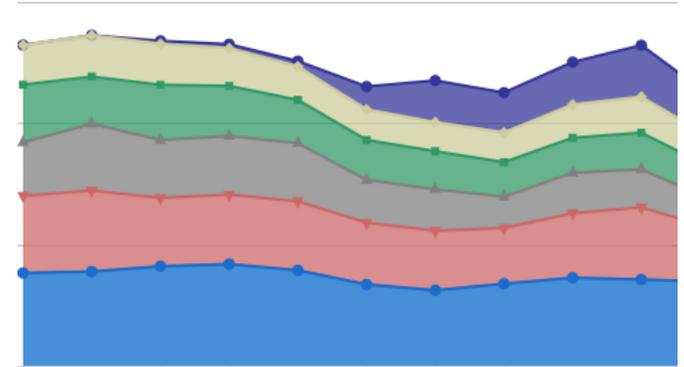
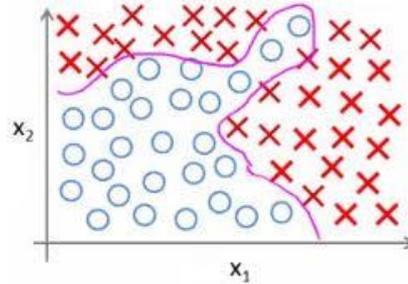
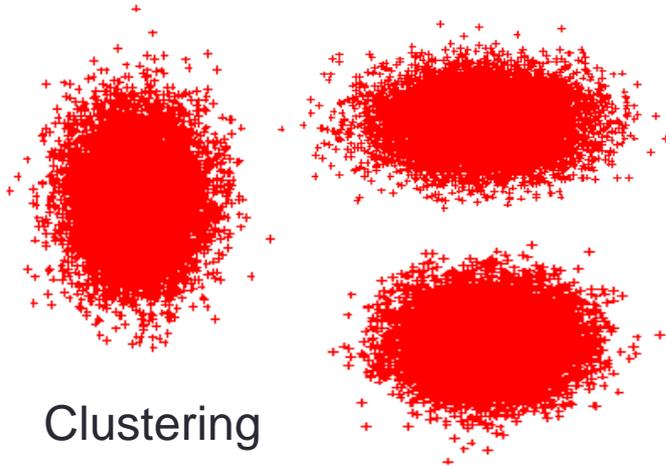


Limitaciones de MapReduce: Nuevas herramientas

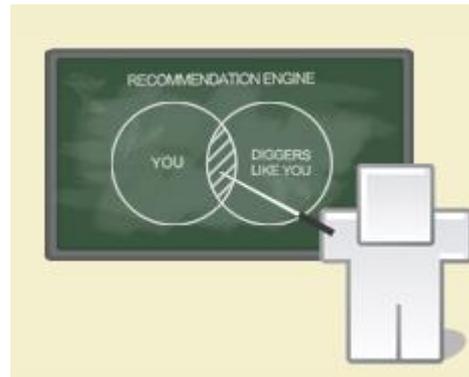
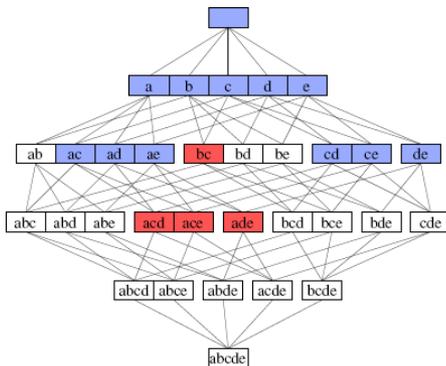


Big Data Analytics

Potenciales escenarios:



Association



Big Data Analytics

Image Credit: [Shutterstock](#)

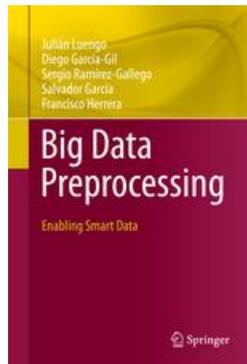
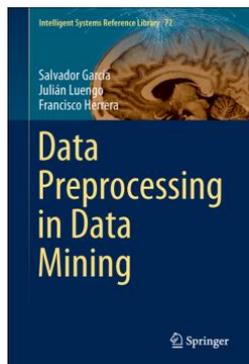
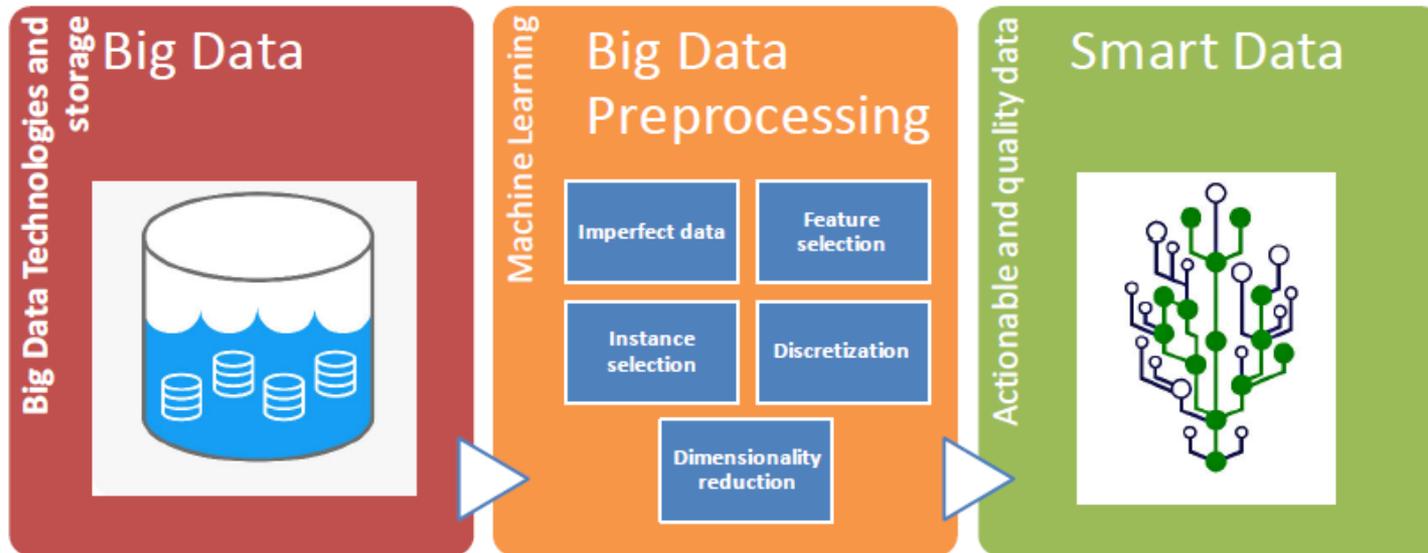


**Without Analytics,
Big Data is Just
Noise**

**Guest post by Eric
Schwartzman, founder and
CEO of [Comply Socially](#)**

Big Data Analytics: Smart Data

¡Se requieren datos de calidad para diseñar modelos de calidad!



Big data preprocessing is the key to transform raw big data into quality and Smart data

Resumen

1. Redes Neuronales. Fundamentos y Deep Learning
2. Ensemble Learning
3. Calidad de datos. Preprocesamiento de Datos
4. Big Data
- 5. Comentarios Finales**

Comentarios Finales

Los problemas de clasificación son los que nos encontramos con más frecuencia en ciencia de datos

Nuevas tendencias:

Deep learning

Ensamblaje de Modelos

Calidad de datos

Big Data

....