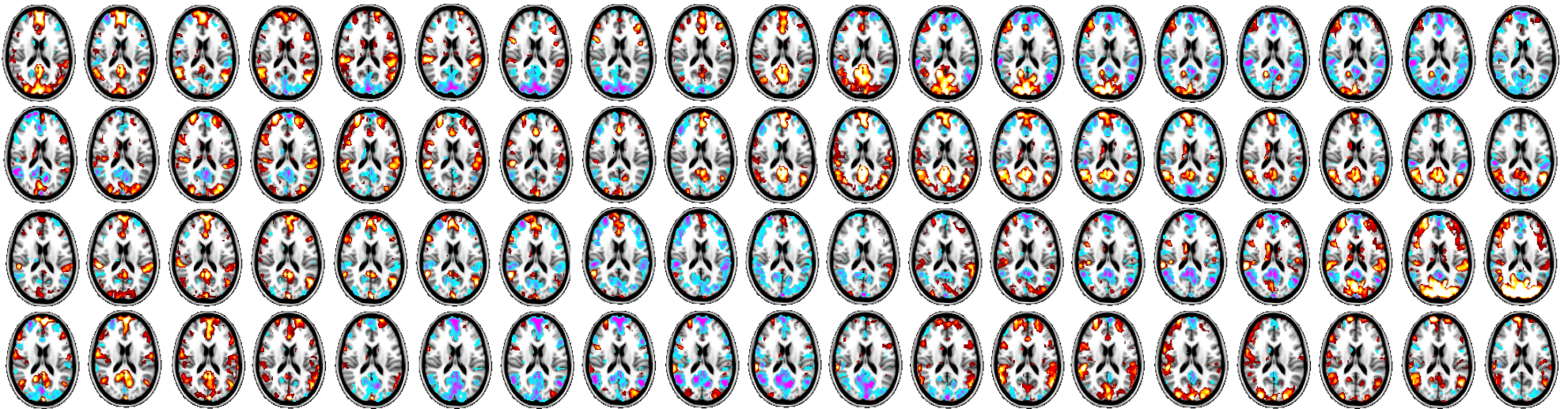


Clasificación de estados cerebrales usando neuroimágenes funcionales

Clase 2:

Reconocimiento de patrones en datos de neuroimágenes



Enzo Tagliazucchi (tagliazucchi.enzo@gmail.com)



universidad de buenos aires - exactas
departamento de Física

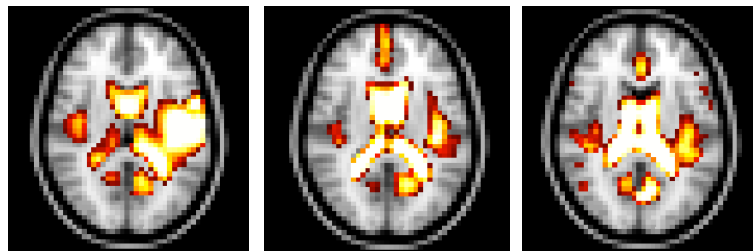
- Primera clase: introducción a las neuroimágenes funcionales + demostración práctica de preprocesado de datos funcionales.
- Segunda clase: introducción a la idea de conectividad cerebral (funcional y anatómica) + demostración de obtención de conectividad funcional (Python)
- Tercera clase: introducción básica a conceptos de machine learning + desarrollo de un clasificador para distinguir vigilia de sueño profundo (Python + scikit-learn)
- Cuarta clase: más métodos de machine learning (feature selection, mapeo de relevancia de features, clasificadores multi clase) cómo medir significancia estadística de clasificadores, etc)
- Quinta clase: temas pendientes + charla general sobre investigación actual en conciencia + preguntas + evaluación



Autos?

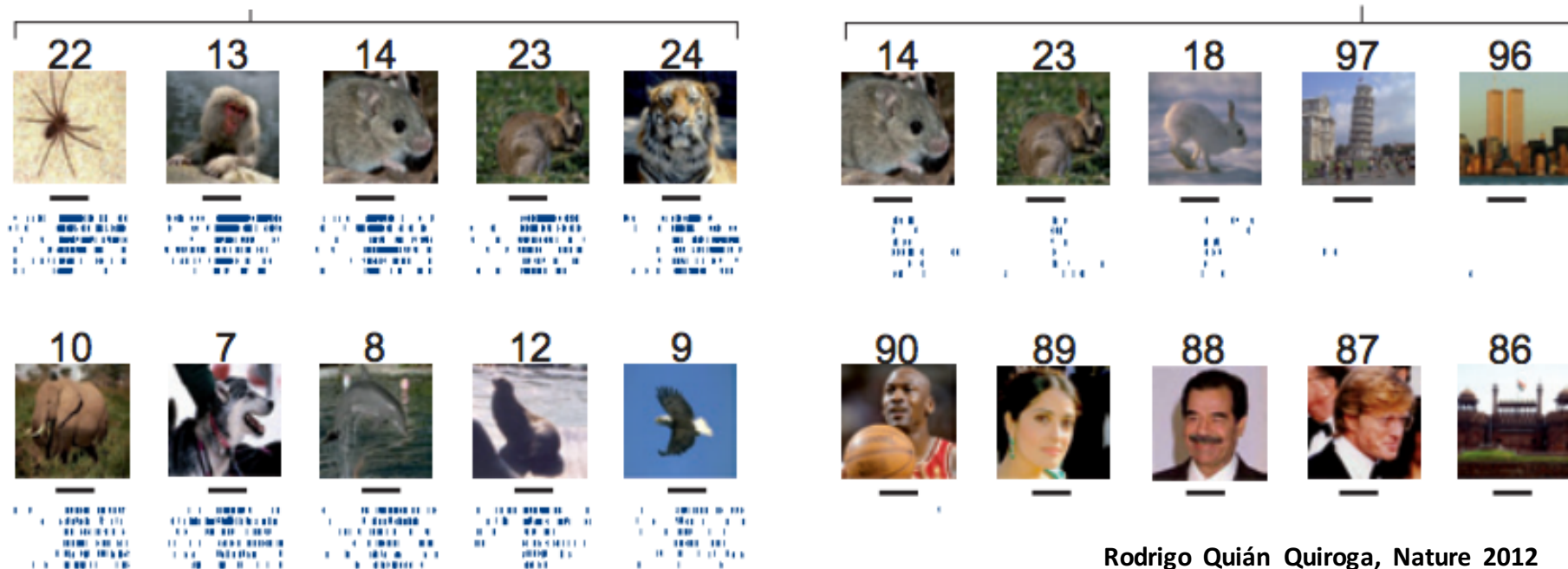


Emociones?



Epilepsia?

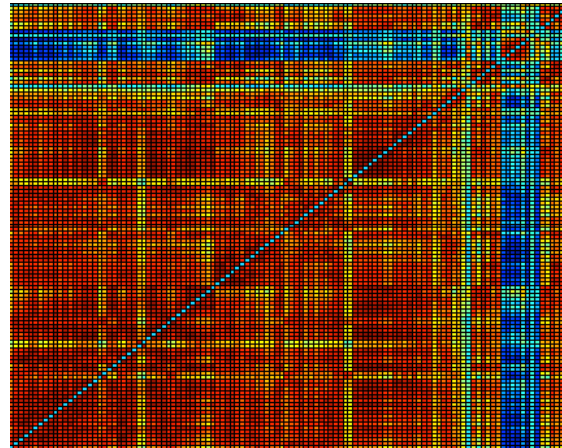
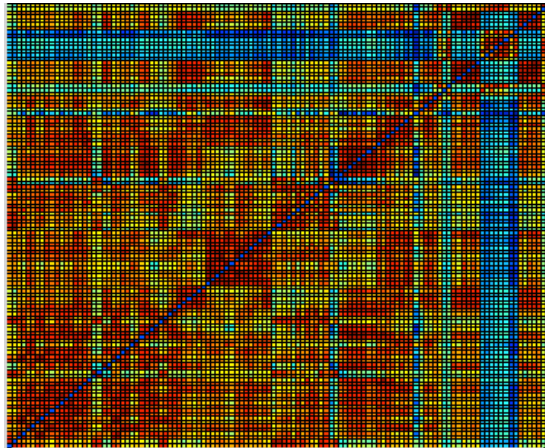
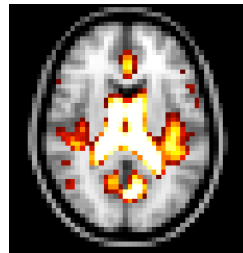
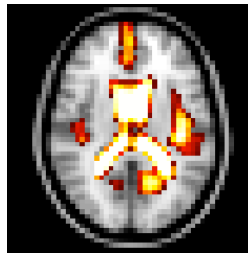
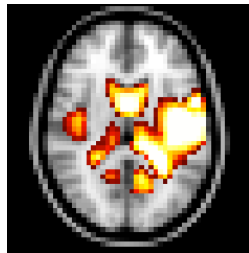
El cerebro humano evolucionó para detectar patrones en escenas naturales (por. ej. caras).



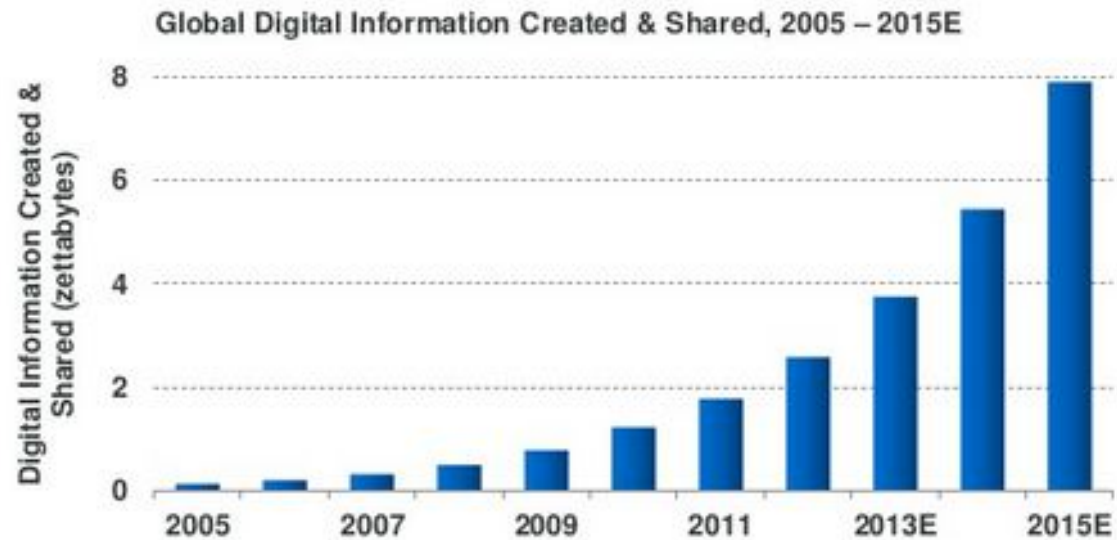
Rodrigo Quián Quiroga, Nature 2012

Una única neurona en el lóbulo temporal se activa solamente con imágenes de animales: el mejor clasificador de imágenes del mundo!

... pero no siempre es sencillo encontrar la información relevante en *datasets* representados arbitrariamente...



Imágenes de resonancia magnética funcional o matrices que codifican la conectividad entre cada par de regiones cerebrales **no se pueden distinguir visualmente para sujetos saludables vs. pacientes con epilepsia del lóbulo temporal**



Demasiada cantidad de datos para que los patrones sean reconocidos por humanos (*“datificación”*!)

Aprendizaje supervisado

Ejemplos



Definir y
"entrenar" un
modelo
matemático



Evaluación del modelo: más ejemplos

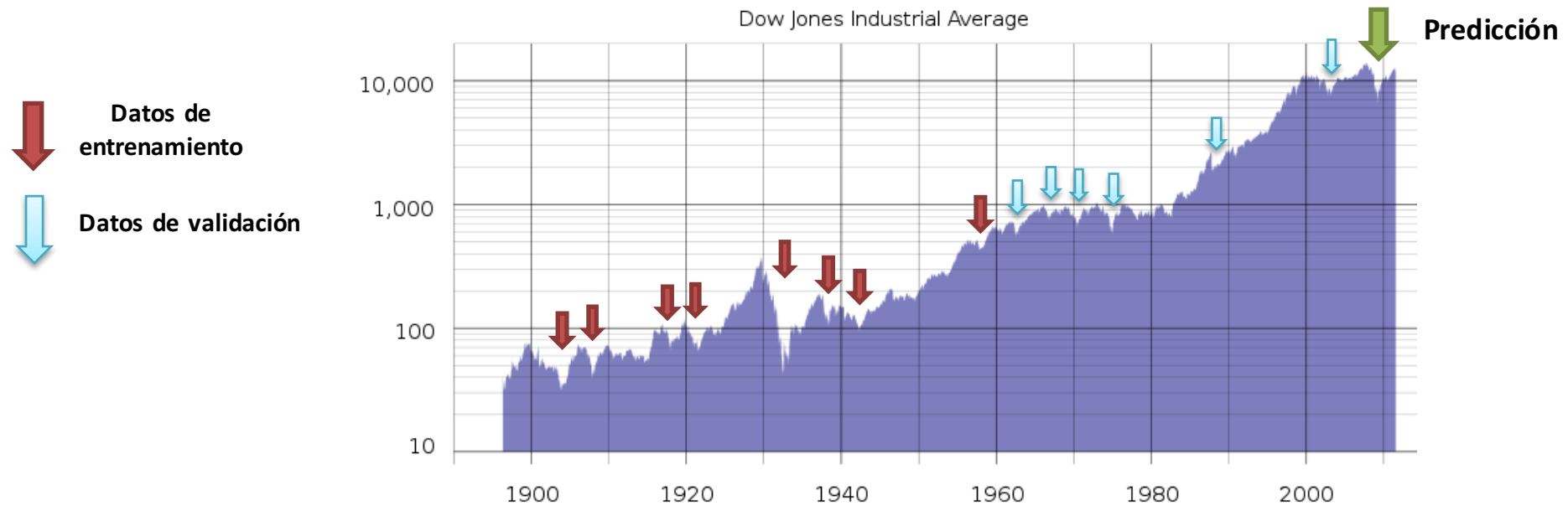


5/7
correctas

Índice Dow Jones, 2002-2011



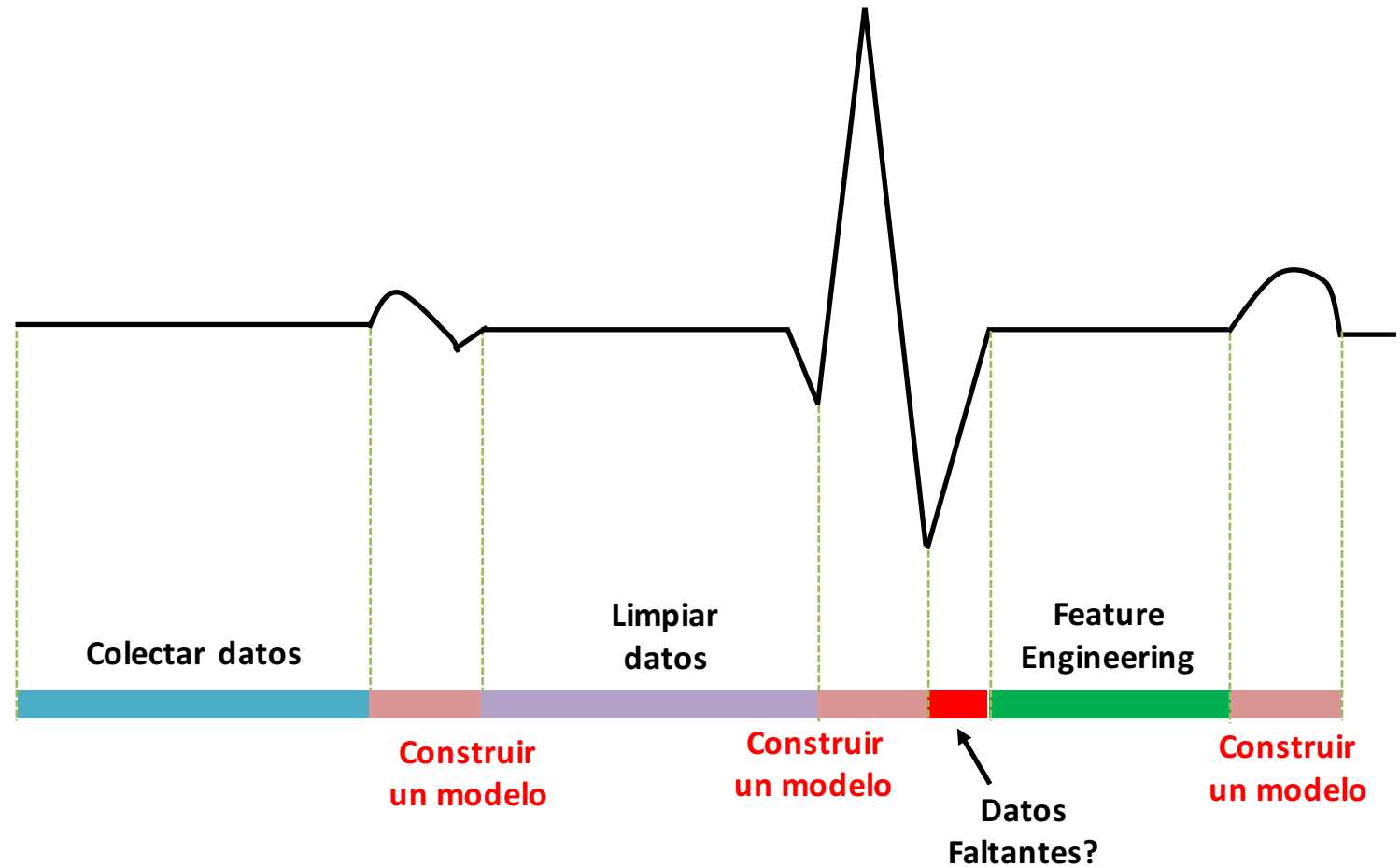
Predecir una recesion?



Se utiliza aprendizaje predictivo con información **previa** el evento que queremos predecir

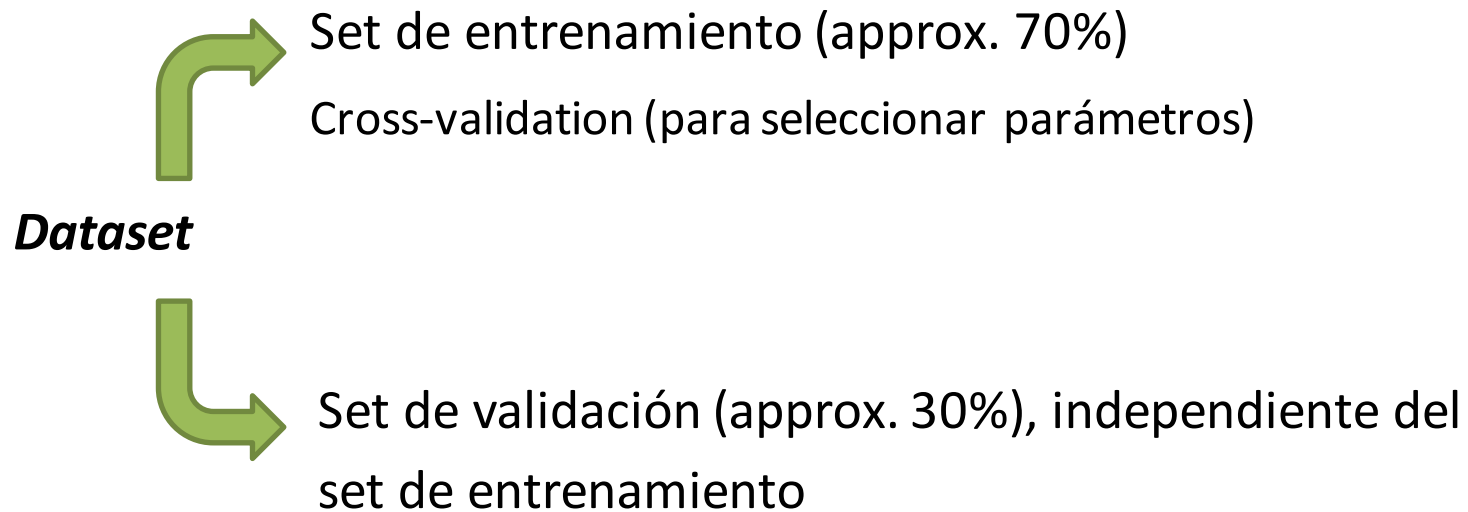
En *modelado predictivo* nuestra predicción interactúa con el resultado final

(por ej. predecir un *market crash* puede terminar ocasionando uno, aún si la predicción era equivocada en primer lugar!)



Data Scientist: The Sexiest Job of the 21st Century

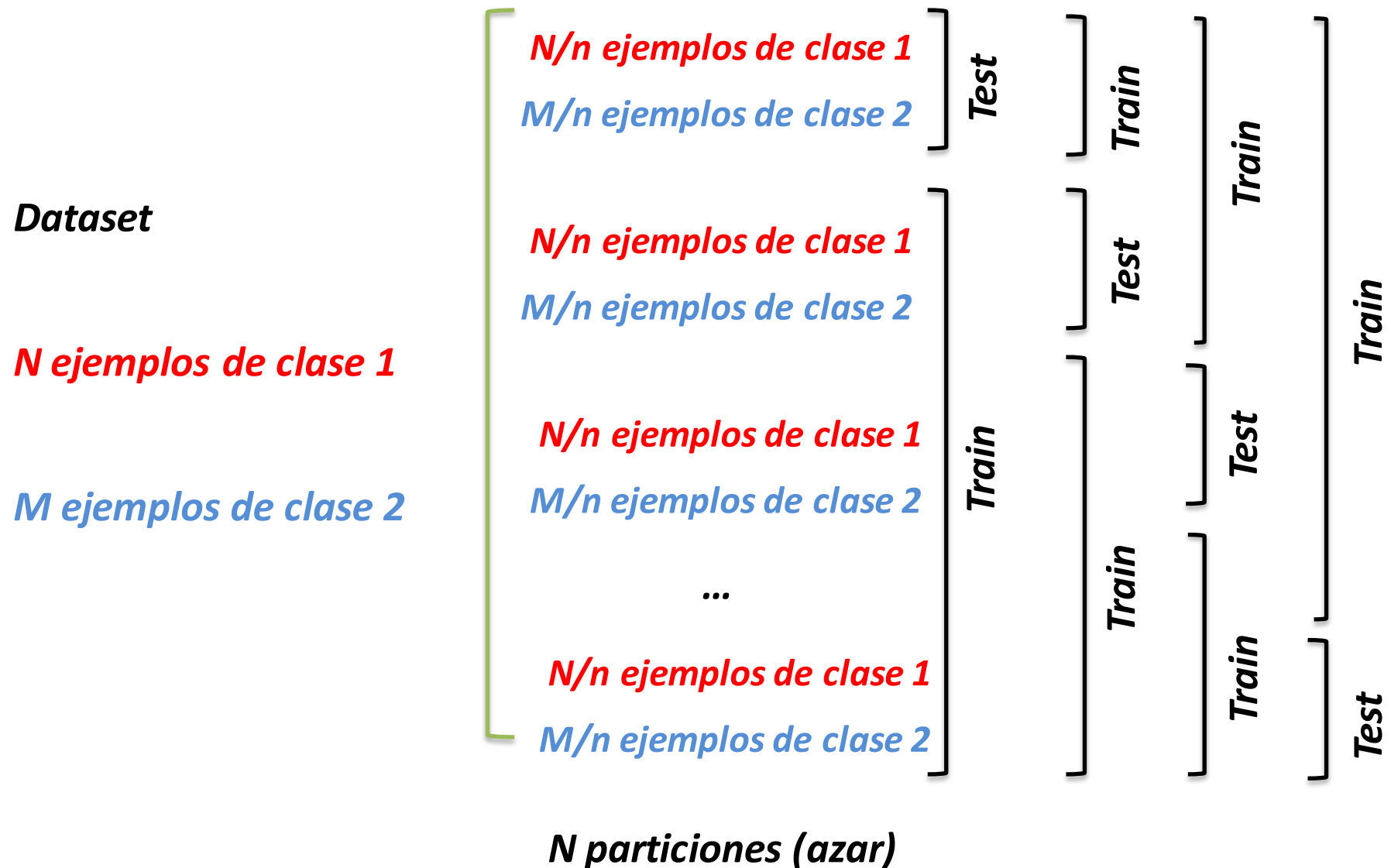
by Thomas H. Davenport and D.J. Patil



“Confusion matrix” (importante! Clase desbalanceadas)

	Predicción: Sí	Predicción: No
Real: Sí	# positivos	# falsos negativos
Real: No	# falsos positivos	# negativos

Cross-validation estratificada (con n folds)

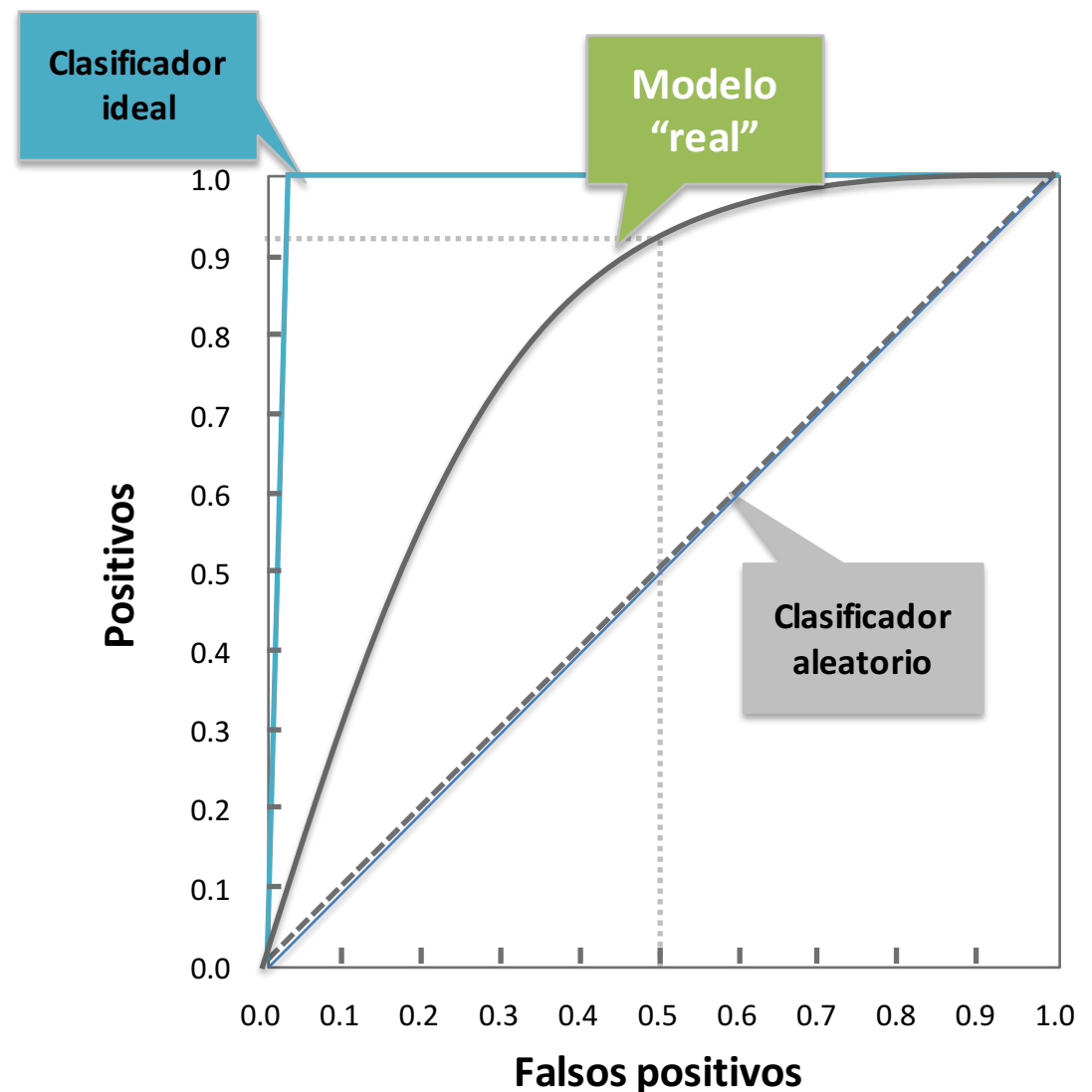


Receiver Operator Characteristic (ROC)

Si el clasificador no devuelve la clase (0 o 1) sino la probabilidad de pertenecer una clase, e.g. probabilidad de pertenecer a 1:

Se prueban distintos “umbrales” T (si la probabilidad de pertenecer a la clase 1 es mayor que T decimos que pertenece a la clase 1)

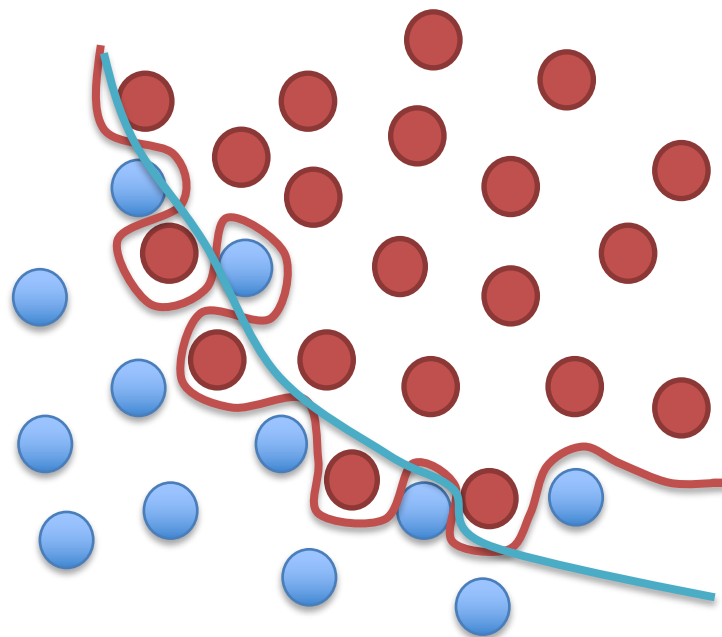
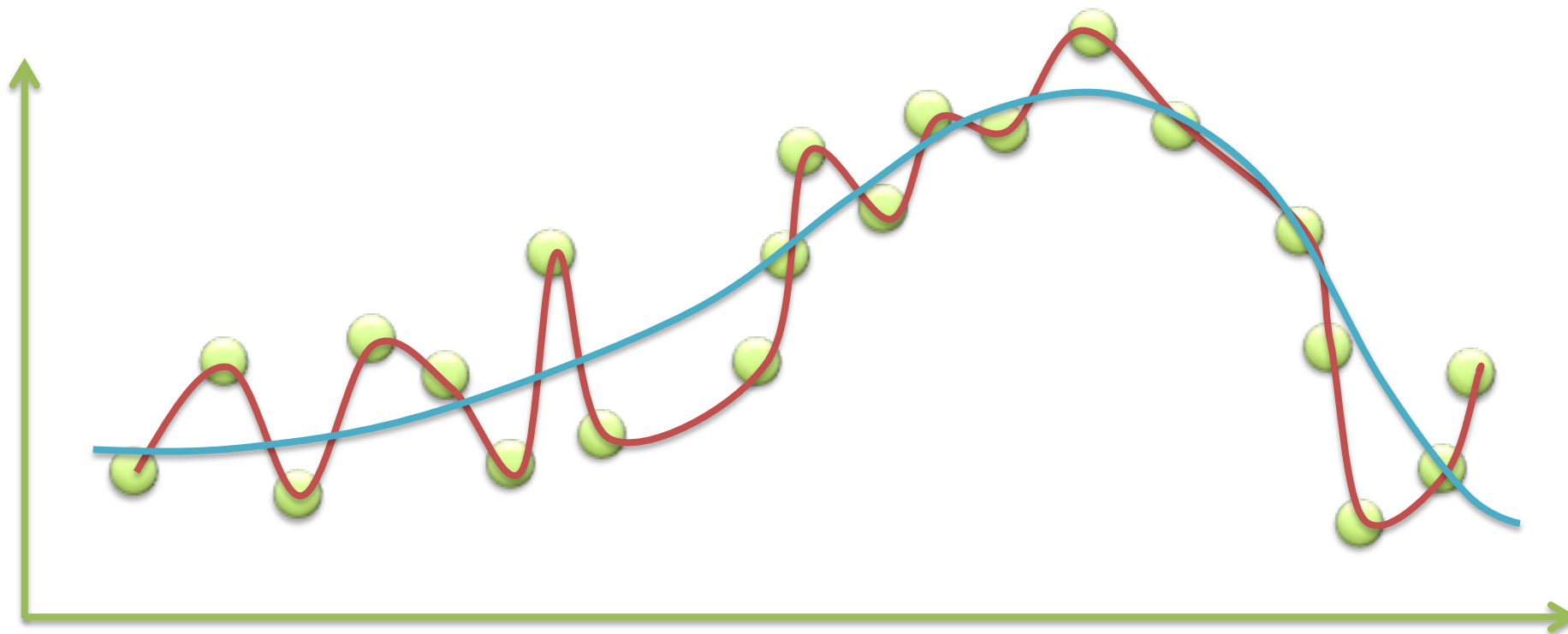
Para cada seleccion de umbral marcamos en un diagrama la proporcion de falsos negativos vs. falsos positivos.



El área debajo de la curva puede usarse como métrica para evaluar el modelo (0.5 en el nivel de chance, >0.5 para un modelo mejor que elegir al azar la predicción), 1 para un clasificador perfecto.

Overfitting: ajustar *demasiado bien* el modelo

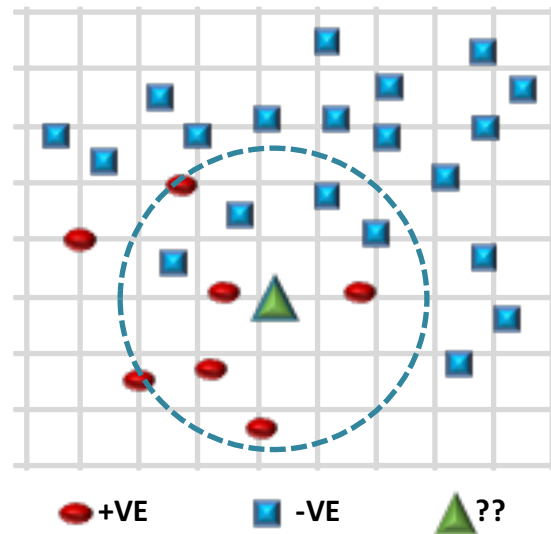
(excelente precisión en el dataset de entrenamiento, pero muy baja precision en el dataset de validación)



Ejemplos de modelos - I

K-Nearest Neighbors (K-NN)

- El clasificador más sencillo e intuitivo
- Considerar los k vecinos más cercanos (k ejemplos más parecidos al caso que queremos clasificar)
- Clase resultante: el tipo de vecinos más frecuente



Vecindad: 6  & 4 

Predicción: 

Ventajas

- Fácil de implementar
- Fácil de interpretar el resultado
- Pocos parámetros a elegir

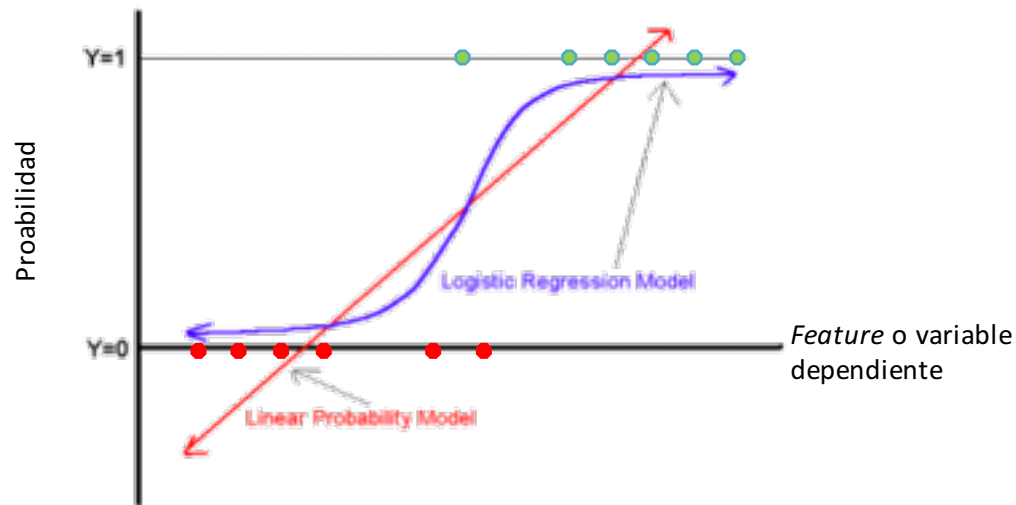
Desventajas

- *Run-time* performance pobre
- Muy sensible a ruido y a ejemplos o a *features* redundantes o innecesarios
- Para *samplear* suficientemente bien los primeros vecinos, si la cantidad de *features* es grande, se necesitan muchos datos.
- Los resultados no se interpretan de manera probabilística.

Ejemplos de modelos - II

Regresión logística

- Se puede utilizar para predecir probabilidades
- Se utiliza regresión lineal con logaritmo natural del cociente entre la probabilidad de que el caso sea de una clase (p) sobre la probabilidad de que el caso sea de la otra clase ($1-p$).



$$LP: Y = \alpha + BX + e$$

$$LR: \ln[p/(1-p)] = \alpha + BX + e$$
$$p/(1-p) = \exp(\alpha + BX + e)$$

Ventajas

- El resultado se interpreta de manera probabilística
- Pocos parámetros para ajustar, inclusive utilizando regularización
- Relativamente fácil de evitar *over-fitting*

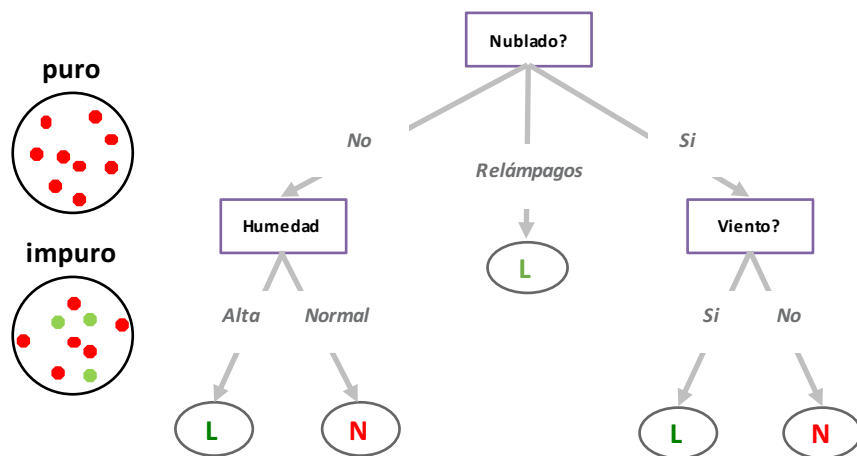
Desventajas

- Solo sirve si el borde que separa las clases es lineal (más sobre esto pronto)
- A diferencia de una regresión lineal, no se pueden calcular los parámetros con una fórmula, hay que usar estimadores

Ejemplos de modelos - III

Árboles de decisión

- El proceso de decisión se forma con una serie de decisiones binarias sobre los *features* o atributos
- En cada decisión binaria, se toma la decisión de manera tal que en cada rama de la decisión binaria la distribución de clases sea lo más pura posible
- El algoritmo continua de manera recursiva hasta que no es posible continuar dividiendo el *dataset* de entrenamiento



Ventajas

- Muy sencilla interpretación (el resultado es una serie de decisiones que deben tomarse para clasificar cada ejemplo en una de las dos clases)
- No requiere ajuste de múltiples parámetros

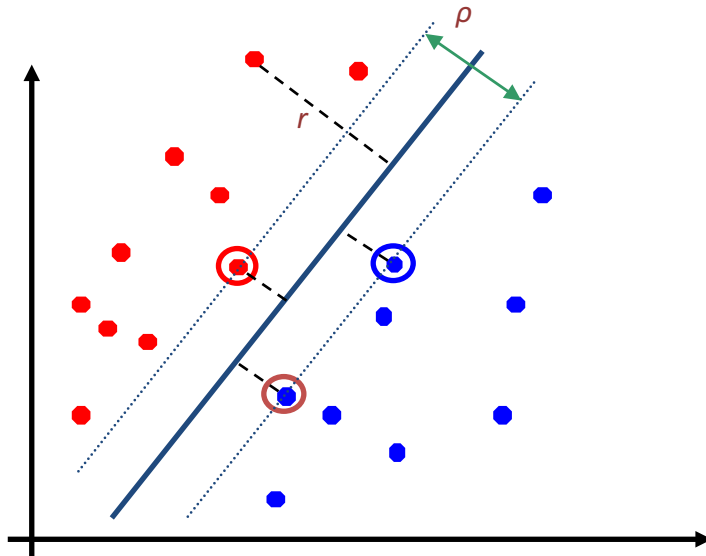
Desventajas

- Si se utiliza el algoritmo sin modificaciones es muy propenso a over-fitting (
- Variaciones en los datos, si bien no muy grandes, pueden dar árboles de decisión muy diferentes.
- Los resultados no se interpretan directamente de manera probabilística.

Ejemplos de modelos - IV

Support Vector Machine (SVM)

- El objetivo es encontrar un borde de decisión separando los ejemplos de ambas clases.
- El borde de decisión se elige de manera tal que se maximiza la distancia de los puntos más cercanos al borde (margen)



Ventajas

- Es capaz de aprender bordes de decisión complicados, más allá de simples rectas, usando transformaciones no lineales del espacio (*kernels*)
- El problema de optimización que debe resolverse es convexo, es decir, existe un *mínimo global* (solución óptima)

Desventajas

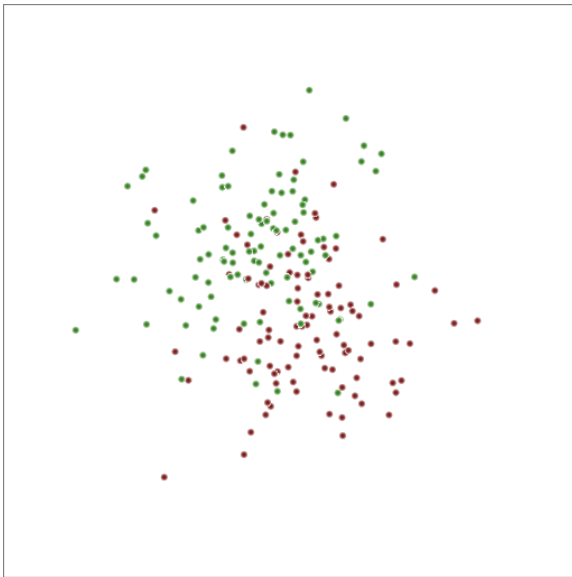
- Deben seleccionarse muchos parámetros distintos.
- El entrenamiento con grandes volúmenes de datos es muy lento.
- Los resultados no se interpretan directamente de manera probabilística.

El algoritmo *random forest*

Se entrena un numero grande (>1000) de arboles de decision, cada uno se entrena sobre m de las M features, elegidas al azar (tipicamente m es del orden de la raiz de M)

Para predecir la clase de un nuevo ejemplo, se aplican todos los arboles de decision entrenados y se hace una “votacion”, la clase con mas votos es la predicha.

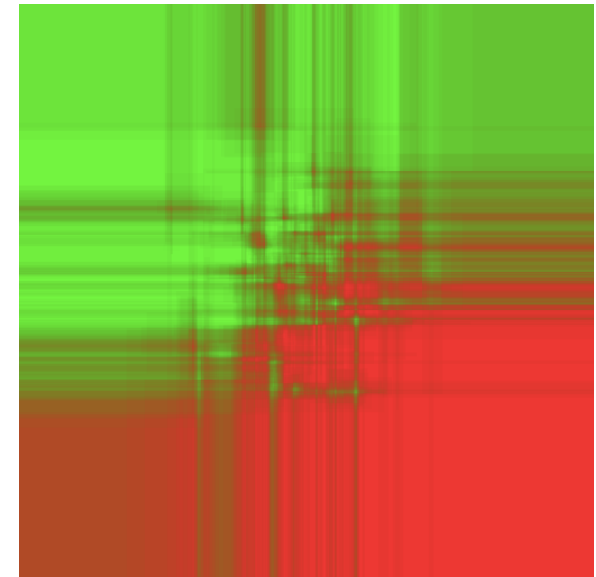
La votacion puede usarse para estimar la probabilidad de pertenecer a cada clase.



Data



Regresion logistica

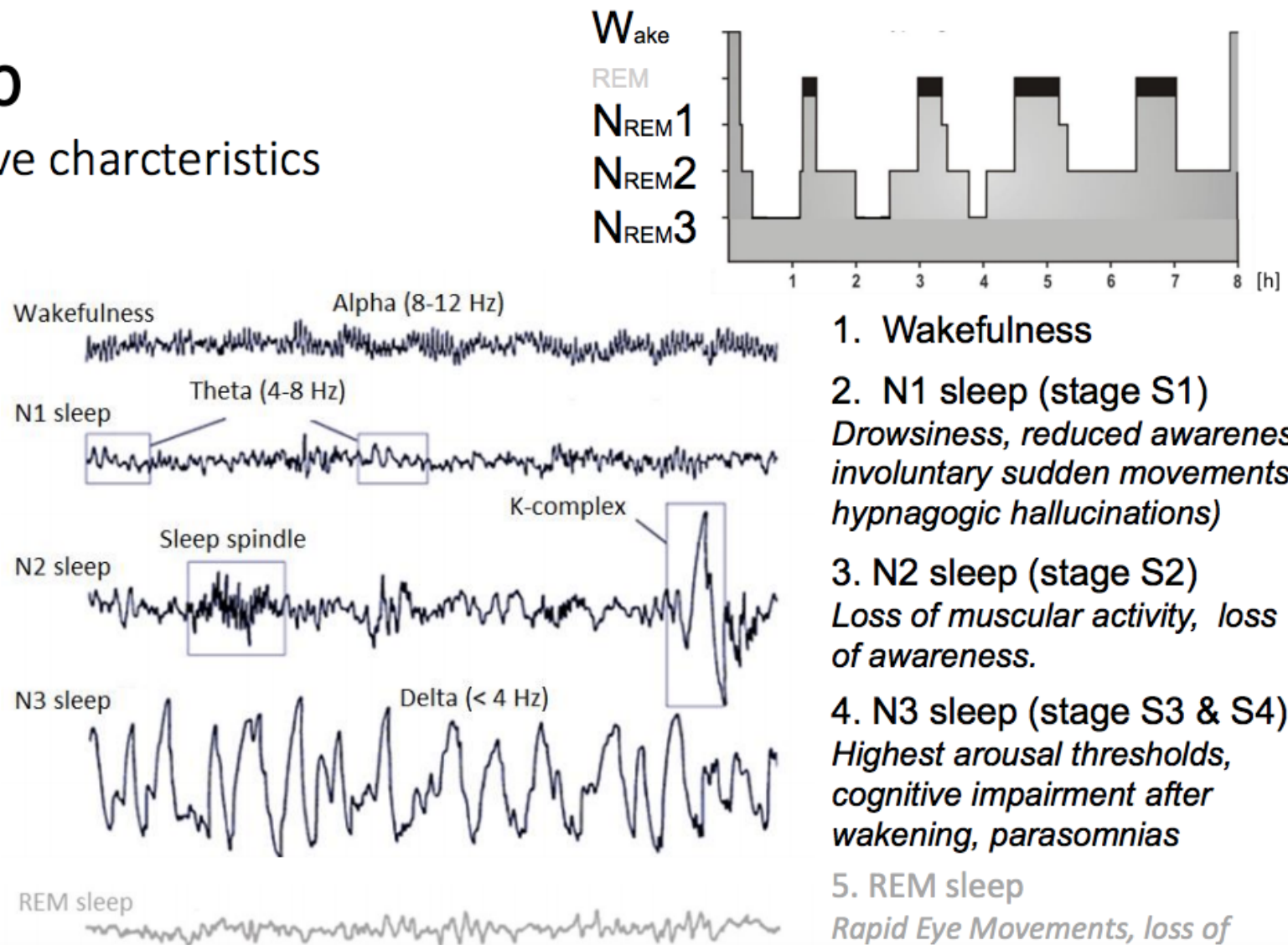


Random forest

Un ejemplo de aplicacion en neuroimagenes

Sleep

objective characteristics



1. Wakefulness

2. N1 sleep (stage S1)

Drowsiness, reduced awareness, involuntary sudden movements, hypnagogic hallucinations)

3. N2 sleep (stage S2)

Loss of muscular activity, loss of awareness.

4. N3 sleep (stage S3 & S4).

Highest arousal thresholds, cognitive impairment after waking, parasomnias

5. REM sleep

Rapid Eye Movements, loss of muscular tone. Dreaming.

Polysomnography during fMRI

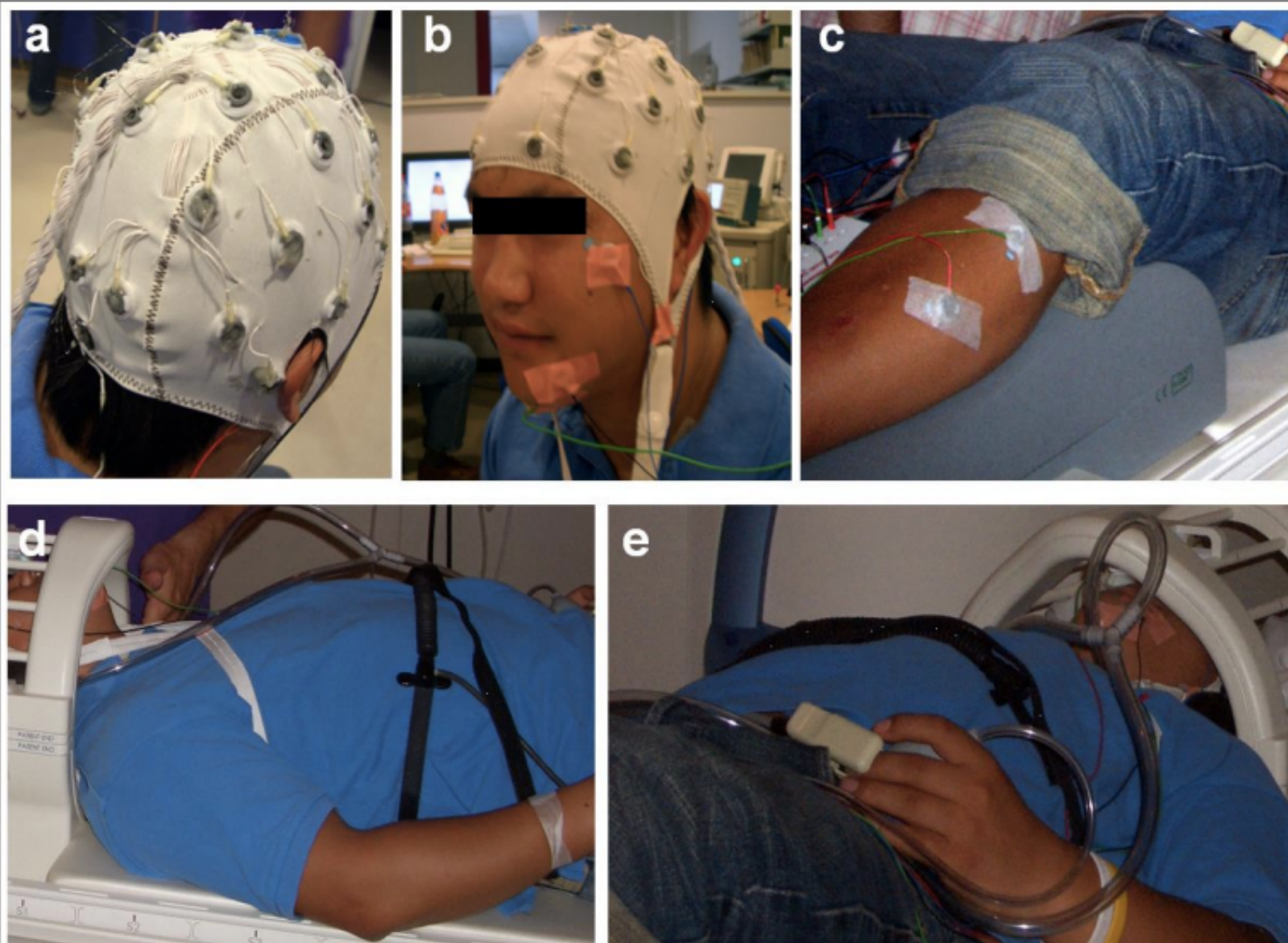
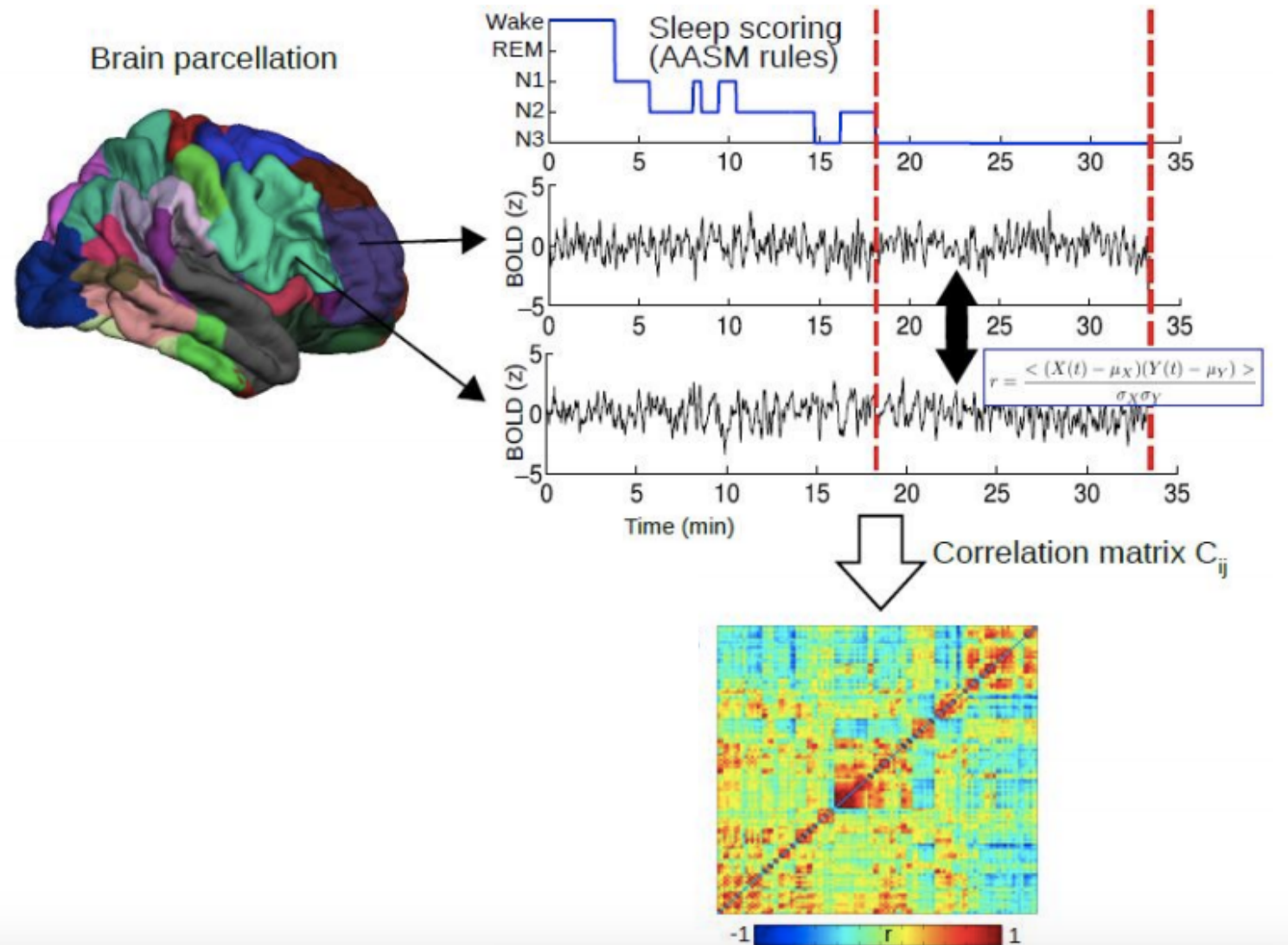


Figure 1: a: EEG cap; b: EOG and M. mentalis EMG electrodes; c: M. tibialis EMG electrodes; d: respiration belt; e: pulse oximetry

Crosscorrelation of regional time series

(pulse, respiration, motion regressed out, no global scaling)



fMRI sleep classification (trained with EEG)

INPUT

functional connectivity matrices
(sliding window, 60 images)

sleep staging based
on RS correlations



support
vector
machine

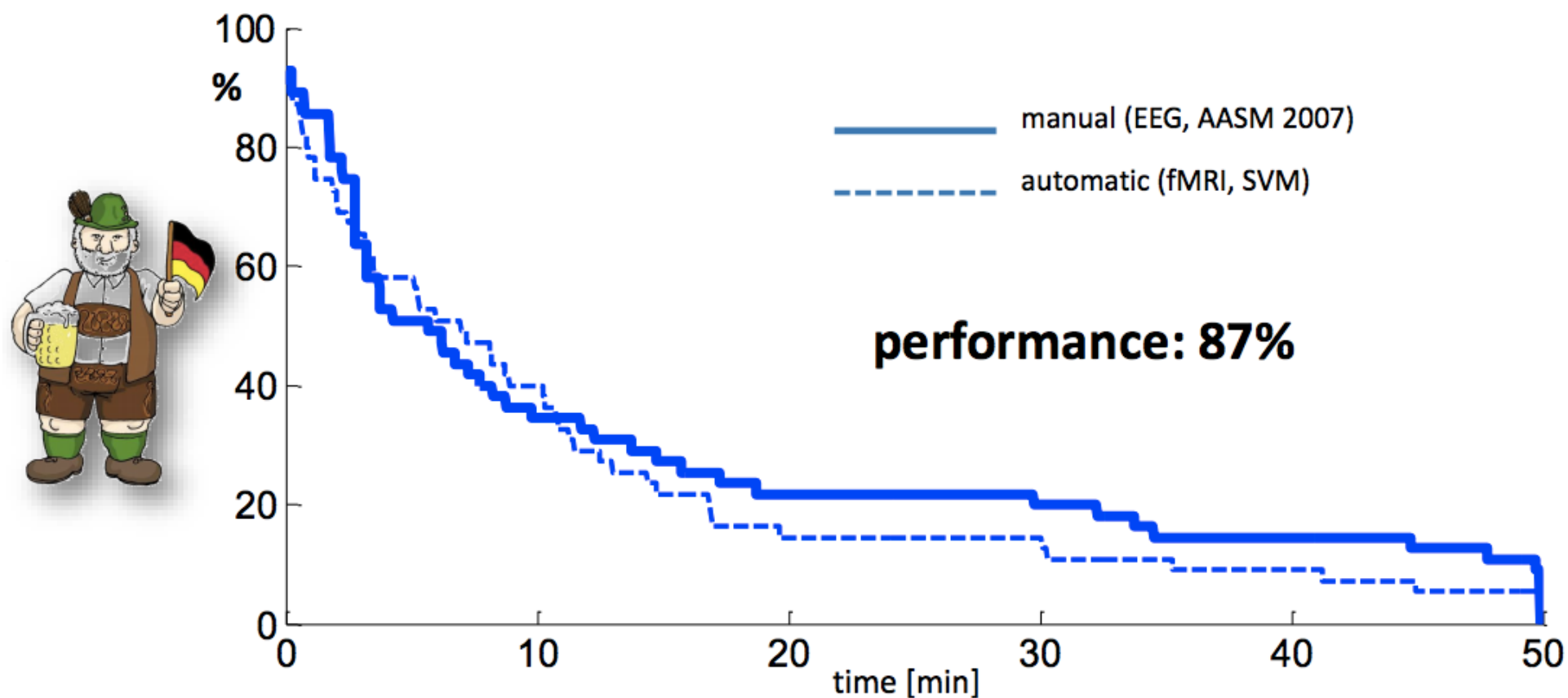
training



hypnogram

OUTPUT

classifier trained on fMRI data



$$n_F=71 \text{ (EEG+sleep scoring)}, n_C=1147 \text{ (fMRI+classifier)}$$
$$n_F=71 \text{ (EEG+sleep scoring)}, n_C=1147 \text{ (fMRI+classifier)}$$
