CURSOS DE VERANO 2014

APROXIMACIÓN PRÁCTICA A LA CIENCIA DE DATOS Y BIG DATA: HERRAMIENTAS KNIME, R, HADOOP Y MAHOUT

Análisis predictivo

María José del Jesus

Motivación ► ¿Quién predice?

- Página web → ¿Hará click?
- Amazon → ¿Qué libro comprará?
- Compañías de seguros → ¿Qué riesgo tiene este seguro?
- Fundación del olivar → ¿Qué precio tendrá el aceite dentro de tres meses?
- Hospital → ¿Qué demanda de plaquetas habrá durante la semana próxima?

Motivación ► ¿Por qué predecir?

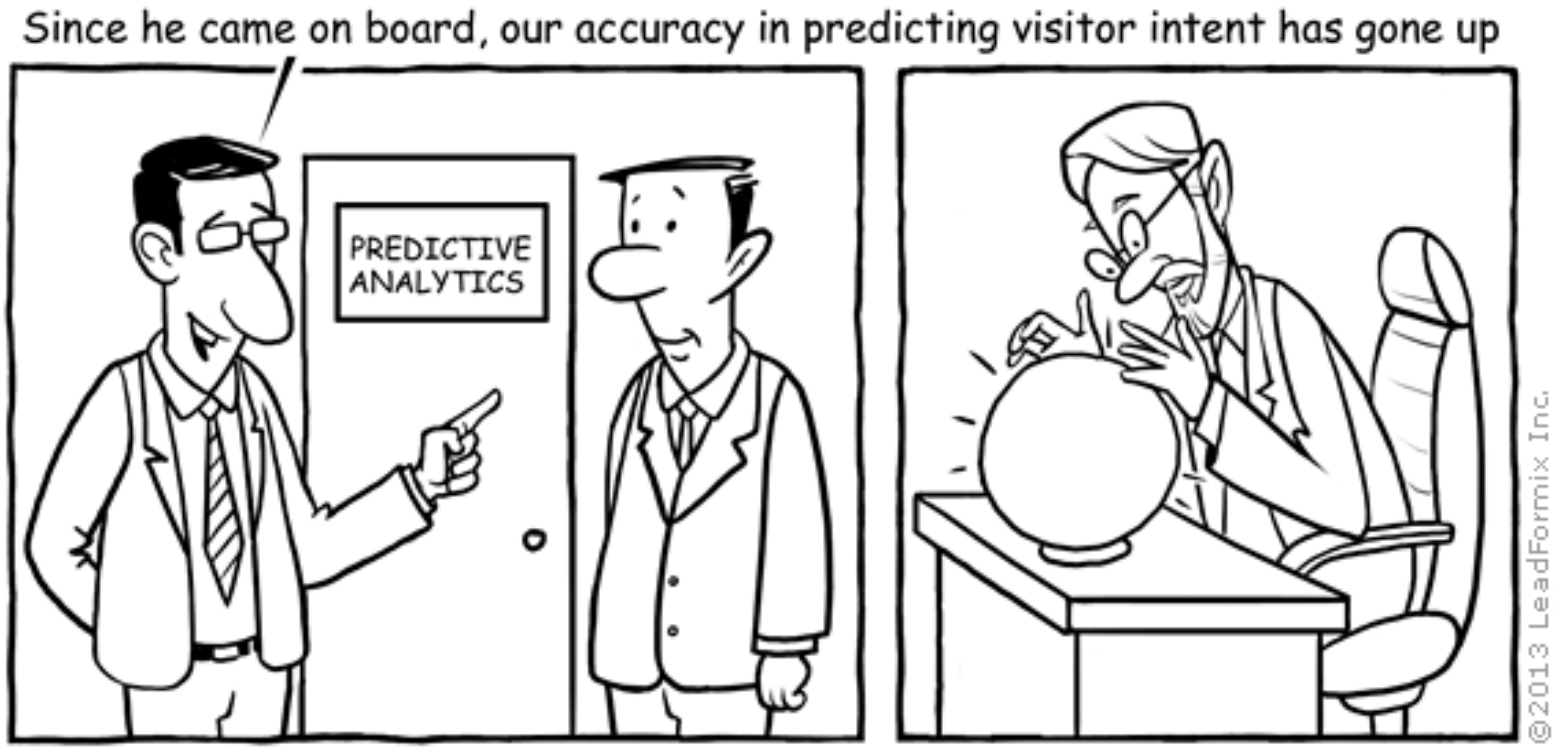
- Para conocer mejor cualquier organización o modelo de negocio, tomar decisiones y mejorarlo
  - Inteligencia de Negocio
- Como reto personal en ocasiones con recompensa económica
  - Premio Netfix http://www.netfixprize.com
  - Heritage Provider Network Health Prize http://www.heritagehealthprize.com/c/hhp
  - Kaggle http://www.kaggle.com
- Como contribución a la investigación
  - Medicina
    - Predicción del nivel de beneficio de tratamientos de quimioterapia y nivel de recurrencia

    http://www.oncotypedx.com/en-US/Home

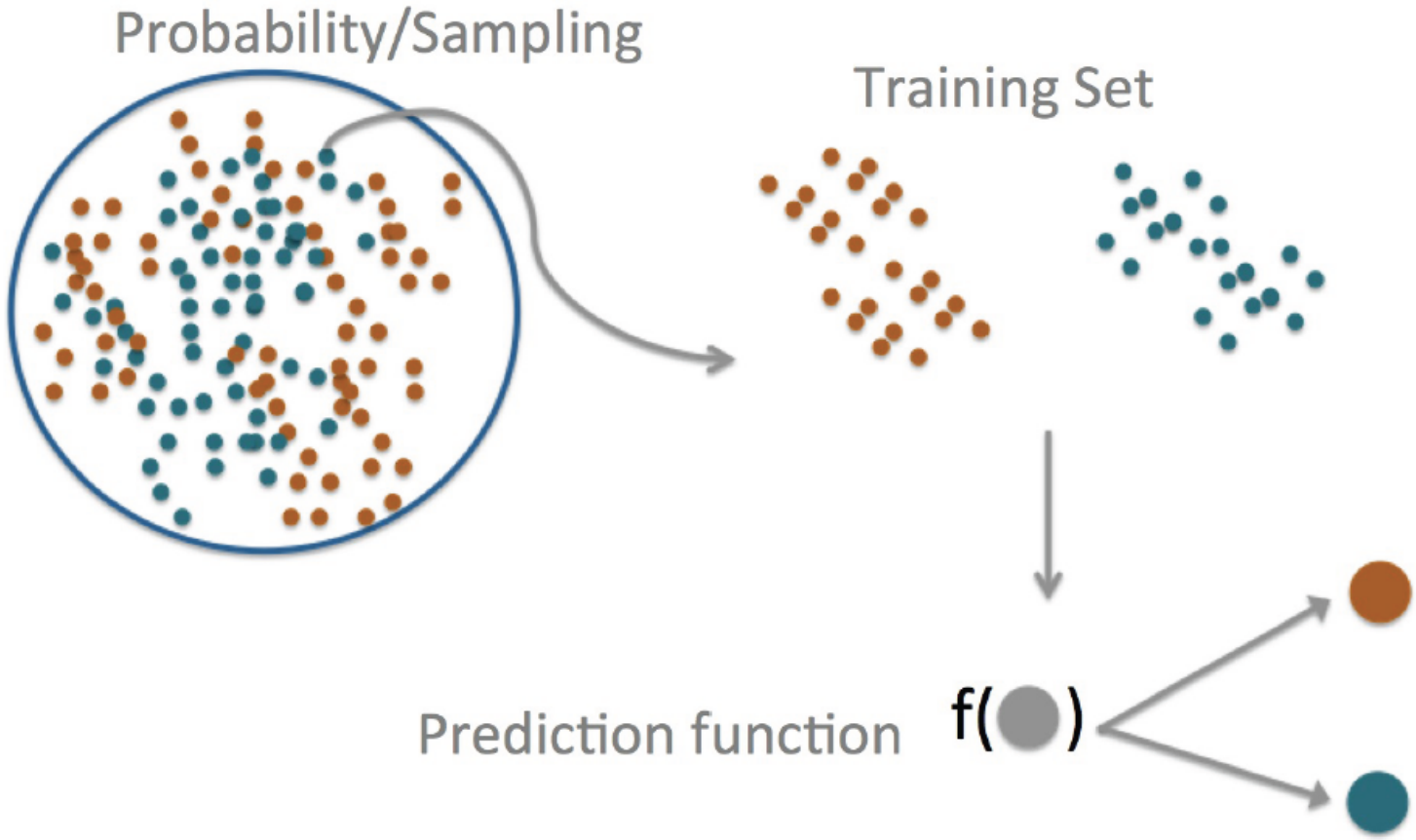
Analítica predictiva

- Motivación
- **¿Qué es predicción?**
- Diseño del modelo de predicción
- Medidas de error
- Preprocesamiento de datos
- Métodos de analítica predictiva

¿Qué es predicción?



¿Qué es predicción?



¿Qué es predicción? ► Clasificación Vs regresión

- Aprendizaje supervisado
- En función del tipo de variable a predecir
  - Discreta: **Clasificación**
    - Clasificación *binaria*: dos clases
    - Clasificación *multiclase*: más de dos clases
    - Caso especial: Cuando un ejemplo pertenece a más de una clase: clasificación *multietiqueta*
  - Continua: *Regresión*
  - Continua pero en un problema en el que los datos tienen dependencia temporal: *Predicción de series temporales*

Predicción de patologías de la faringe

Se encuadra en el ámbito de la Bioinformática: análisis de datos de secuencias de adn y proteínas de individuos sanos y con diferentes patologías de la faringue

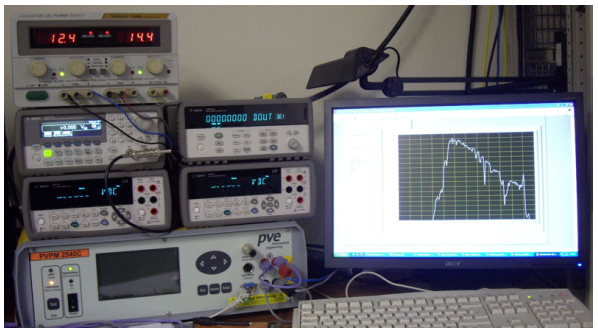
Objetivo: disponer de herramientas que ayuden en la predicción de la patología de un nuevo paciente (variable discreta)

Técnicas de MD: Predicción de variables discretas (clasificación multiclase)

Predicción de la producción de celulas solares de concentración

La tecnología fotovoltaica de concentración implica una reducción de costes y una mayor producción de energía pero no se dispone de modelos de predicción de la producción

Objetivo: predecir la potencia generada por un determinado módulo fotovoltaico de concentración (variable continua)



Técnicas de MD: Predicción de variables continuas (regresión)

Predicción de stock de productos sanguíneos

En colaboración con el Centro Regional de Transfusión Sanguínea y Banco Sectorial de Tejidos de Granada y Almería

Productos generados a partir de una extracción de sangre:

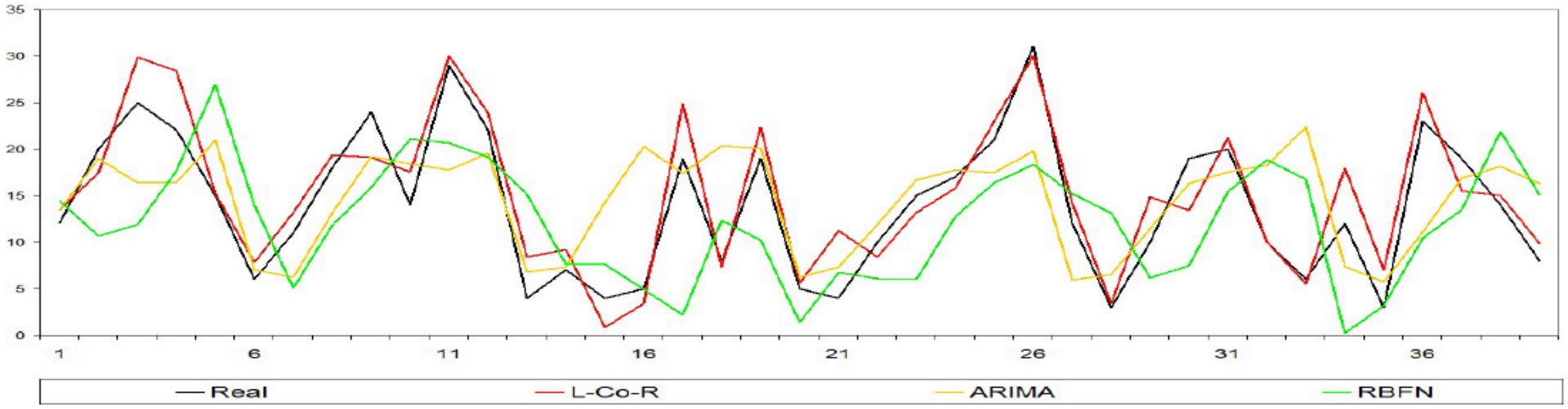
- Sangre total
- Hematíes (conservación: 42 días)
- Plaquetas (conservación: 5 días)
- Plasma

Objetivo: predecir el stock de los componentes de forma que se pueda planificar el abastec a Hospitales sin tener demasiado exceso.

Técnicas de MD: Predicción de series temporales

Predicción de stock de productos sanguíneos

Series temporal correspondiente a plaquetas



¿Qué es predicción? ► Componentes

1. Objetivo: ¿qué queremos predecir?
2. Datos de entrada
3. Variables
4. Algoritmo de Minería de Datos
5. Parámetros
6. Evaluación

¿Qué es predicción? ► Componentes

"The combination of some data and an aching desire for an answer does not ensure that a reasonable answer can be extracted from a given body of data"

John Tukey (1915-2000)
Estadístico eminente
Universidad de Princeton

Análisis exploratorio de datos
Visualización

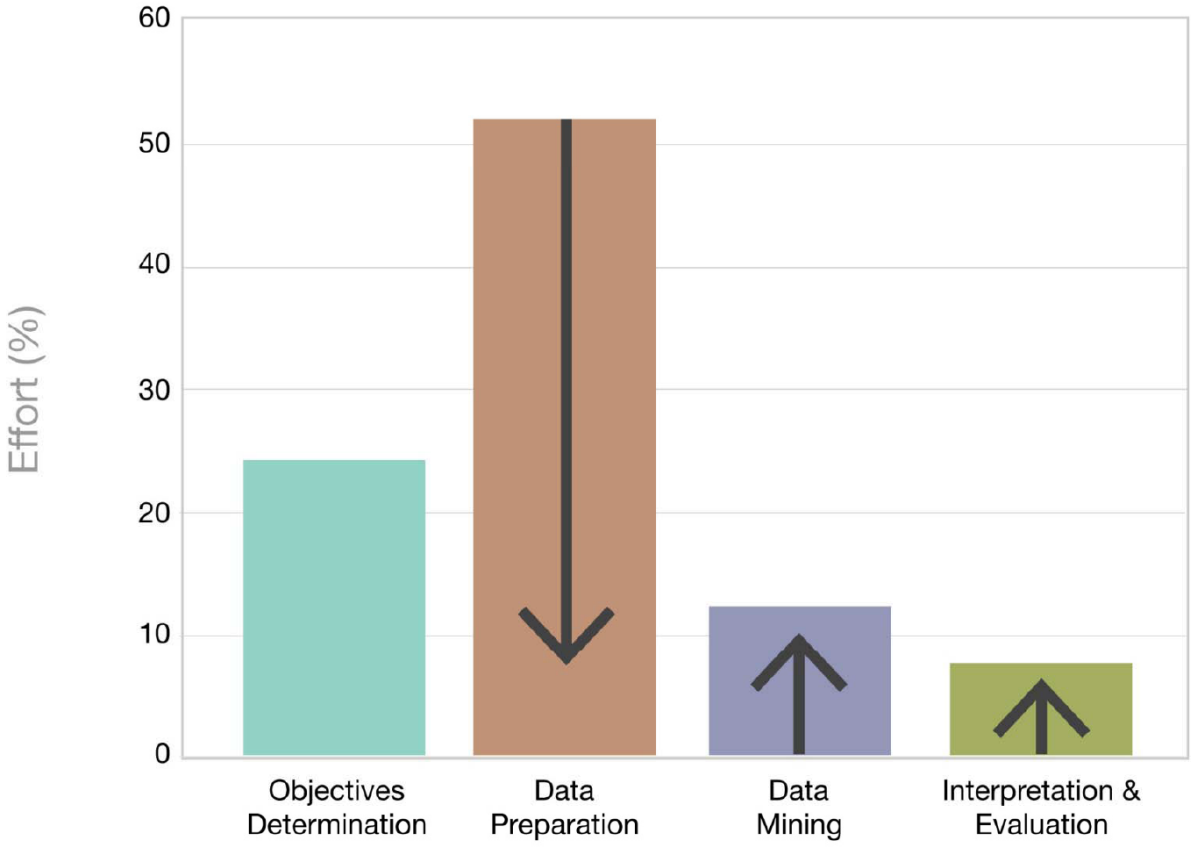
¿Qué es predicción? ► Componentes ► Datos de entrada

- La selección/preparación de datos es la etapa más importante
- Mejorar los datos de entrada puede ser fácil o muy difícil
- Con frecuencia, con más datos se obtienen mejores modelos

- Buenas variables permiten una buena compresión de datos, retienen información relevante
- Habitualmente se crean en base a conocimiento/experiencia en el problema
- Es un error no prestar atención a particularidades de los datos para el problema a resolver
- Es un error deshacerse de información innecesariamente

¿Qué es predicción? ► Componentes ► Variables

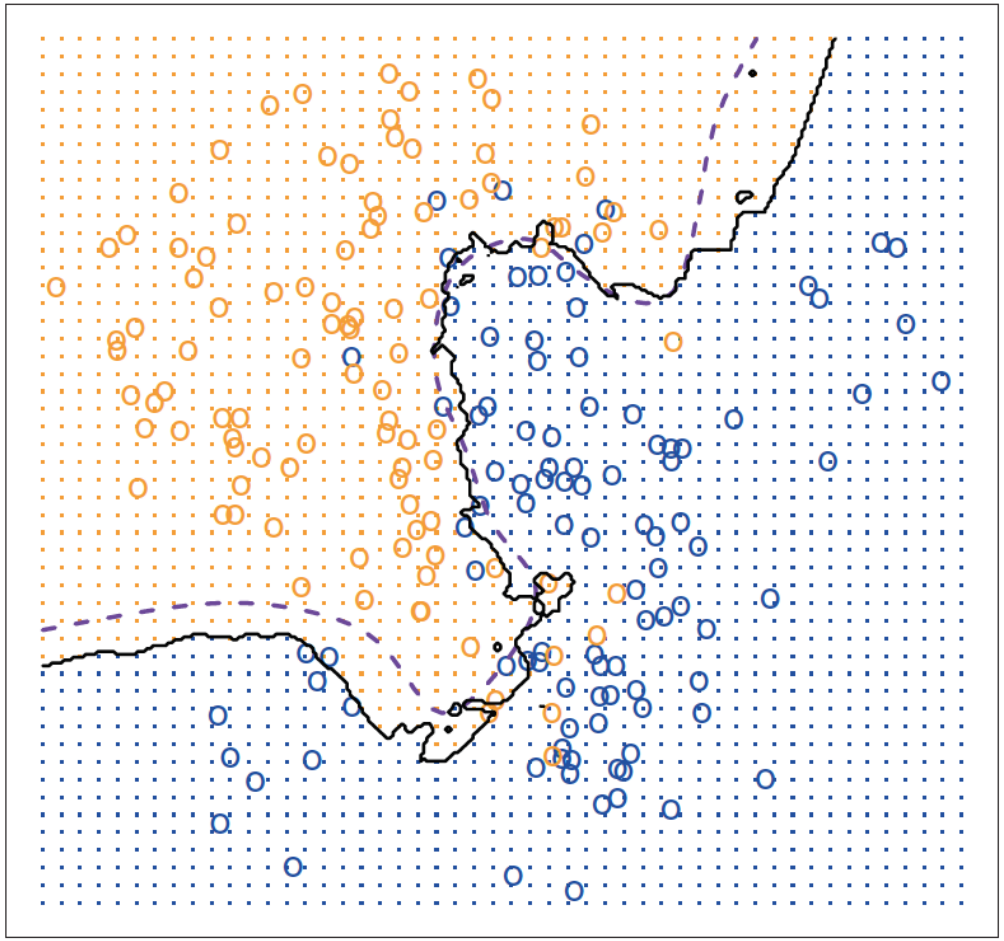
La selección de datos y variables y el tratamiento de los mismos determinan el **pre-procesamiento de los datos**



Algorithm	$RMSE_{tra}$	$RMSE_{test}$
CO^2RBFN	4.9716	4.9970
LMSQR	5.1199	5.1424
MLP-CG	5.1460	5.1614
LMSLR	5.3052	5.3089
v-SVM	5.5726	5.5748
Incr-RBFN	6.0948	6.1069
RBFN-LMS	6.2966	6.3136
WM-FRBS	6.6321	6.6799

Resultados obtenidos con diferentes técnicas para el problema de regresión de energía fotovoltaica de concentración

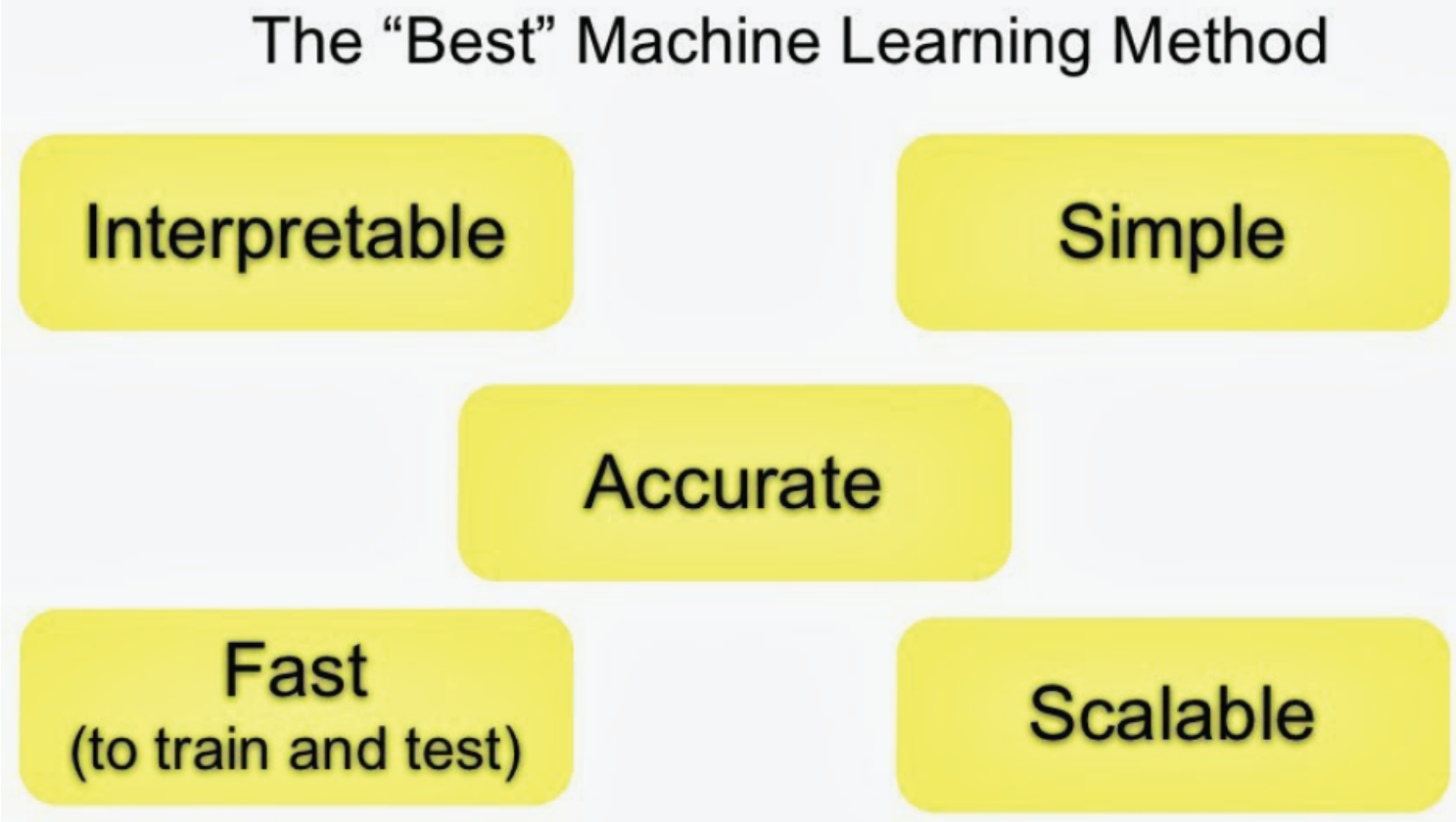
¿Qué es predicción? ► Componentes ► Algoritmos



----- 10NN

- - - Naïve Bayes

An Introduction to Statistical Learning with Applications in R. Springer, 2013

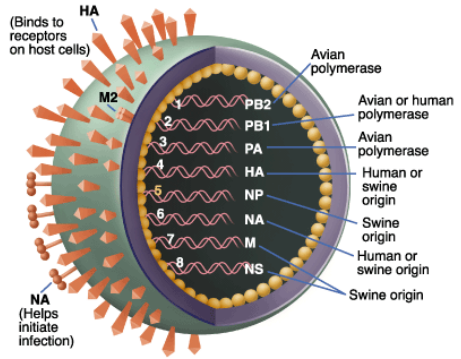


http://radar.oreilly.com/2013/09/gaining-access-to-the-best-machine-learning-methods.html

En predicción siempre se busca un equilibrio en el que interviene la precisión:

- Interpretabilidad Vs **precisión**
- Velocidad Vs **precisión**
- Simplicidad Vs **precisión**
- Escalabilidad Vs **precisión**
- …

En muchos problemas, la interpretabilidad es importante y condiciona la elección del algoritmo



Conjunto de reglas obtenidas para un problema de clasificación de subtipos de gripe A en los que **intervienen 256 variables**

SI (f44=Bajo Y f97=Bajo) → H1N1

SI (f9=Bajo Y f54=Bajo y f153=Bajo Y f217=Bajo) → H2N2

SI (f8=Bajo) → H3N2

SI (f141=Bajo y F207=Bajo Y f219=Bajo) → H3N2

SI (f115=Bajo)→ H5N1

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

- Un ejemplo: en 2009 el algoritmo ganador del premio de 1.000.000 $ Netfix, no se implementó nunca
- Motivo: Complejidad computacional

https://www.techdirt.com/blog/innovation/articles/20120409/03412518422

Why Netflix Never Implemented The Algorithm That Won The Netflix $1 Million Challenge

from the *times-change* dept

¿Qué

mo?

You probably recall all the excitement that went around when a group **finally won** the big Netflix $1 million prize in 2009, improving Netflix's recommendation algorithm by 10%. But what you might *not* know, is that **Netflix never implemented that solution itself**. Netflix recently put up a blog post **discussing some of the details of its recommendation system**, which (as an aside) explains why the winning entry never was used. First, they note that they *did* make use of an earlier bit of code that came out of the contest:

> *A year into the competition, the Korbell team won the first Progress Prize with an 8.43% improvement. They reported more than 2000 hours of work in order to come up with the final combination of 107 algorithms that gave them this prize. And, they gave us the source code. We looked at the two underlying algorithms with the best performance in the ensemble: Matrix Factorization (which the community generally called SVD, Singular Value Decomposition) and Restricted Boltzmann Machines (RBM). SVD by itself provided a 0.8914 RMSE (root mean squared error), while RBM alone provided a competitive but slightly worse 0.8990 RMSE. A linear blend of these two reduced the error to 0.88. To put these algorithms to use, we had to work to overcome some limitations, for instance that they were built to handle 100 million ratings, instead of the more than 5 billion that we have, and that they were not built to adapt as members added more ratings. But once we overcame those challenges, we put the two algorithms into production, where they are still used as part of our recommendation engine.*

Neat. But the winning prize? Eh... just not worth it:

> *We evaluated some of the new methods offline but the additional accuracy gains that we measured did not seem to justify the engineering effort needed to bring them into a production environment.*

- ***In sample error:*** Se calcula con los mismos datos con los que se construye el predictor
- ***Out sample error*** (**error de generalización**): calculado con un conjunto de datos nuevo
  - El error obtenido con los datos utilizados para obtener el modelo es una medida demasiado optimista
  - **In *sample error* < *out sample error*** por el sobreajuste
    - Los datos tienen dos partes: señal y ruido
    - El objetivo de un predictor es encontrar la señal
    - Si el diseño del predictor se hace solo en base al *in sample error* se puede obtener un predictor perfecto (en esos ejemplos)
      - Estaríamos capturando señal y ruido
      - El predictor no funcionaría bien en nuevos ejemplos

Ideas clave:

- Se debe estimar la precisión sobre un conjunto independiente (test set)
- Para ajustar/elegir el modelo no se puede utilizar el test set

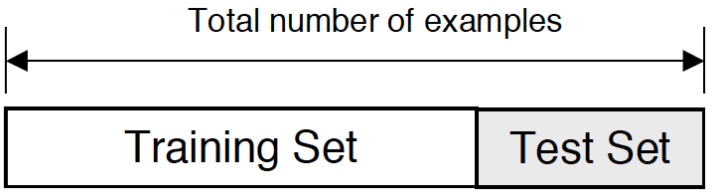
Técnicas de validación de predictores

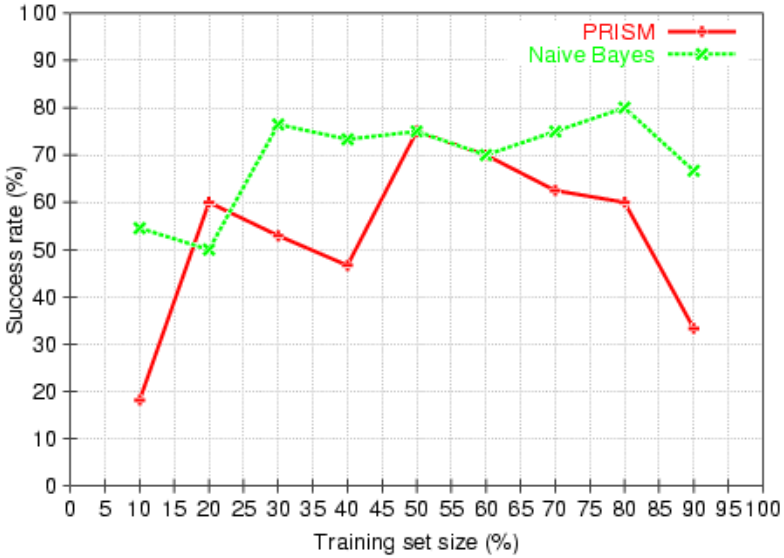
- Objetivo: realizar una estimación honesta de la calidad del predictor construido
- Dividen el dataset al menos en dos conjuntos (entrenamiento y test)
- Existen diferentes técnicas de validación:
  - *Hold-out*
  - Validación cruzada
  - *Leaving one out*
  - *Boostraping*

Hold-out

- Validación de solo una vuelta
- Divide la BD en dos conjuntos independientes: entrenamiento (CE) y test (CT)
- El tamaño del CE es mayor que el del CT (2/3, 1/3, 4/5, 1/5,…)
- Los elementos del CE suelen obtenerse mediante muestreo sin reemplazamiento de la BD

  El CT está formado por los elementos no incluidos en el CE
- Suele utilizarse en BBDD grandes





Test set (%) + Training set (%) = 100%

Validación cruzada

- Consiste en:
  1. Dividir la BD (conjunto de entrenamiento original) en CE/CT
  2. Construir un predictor basado en CE
  3. Evaluar el predictor con el CT correspondiente
  4. Repetir el proceso (volver a 1) y devolver como tasa de acierto (error) el promedio obtenido en los CT
- *Validación cruzada estratificada*: Los subconjuntos CE/CT se estratifican en función de la variable clase

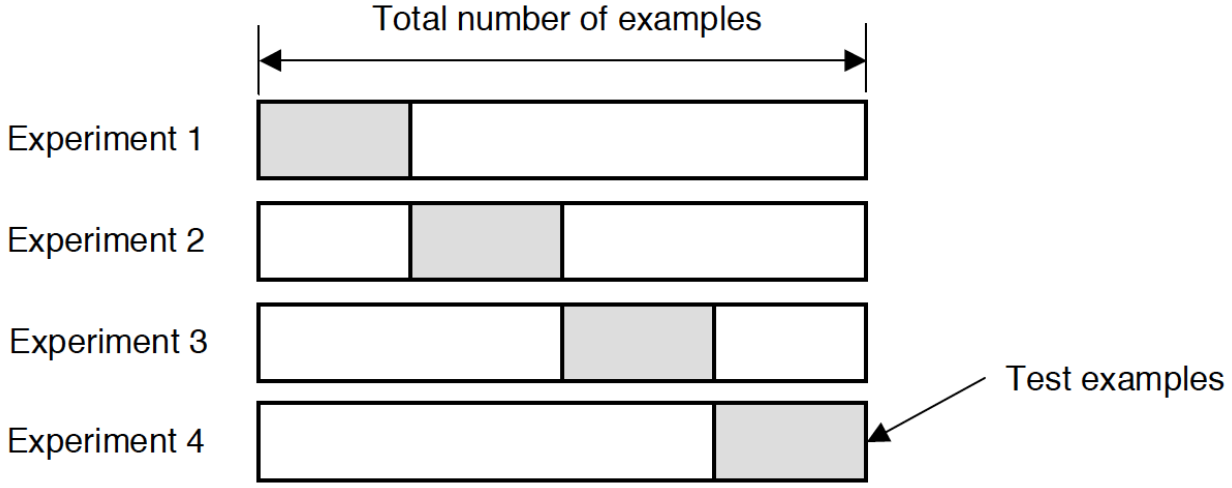
Validación cruzada

- ¿Para qué se utiliza?
  - Para elegir las variables a incluir en el modelo
  - Para elegir el tipo de función de predicción a utilizar
  - Para determinar los parámetros del modelo
  - Para comparar diferentes predictores

- **Tipos de validación cruzada**
  - Random subsampling
  - K-fold
  - Leaving one out

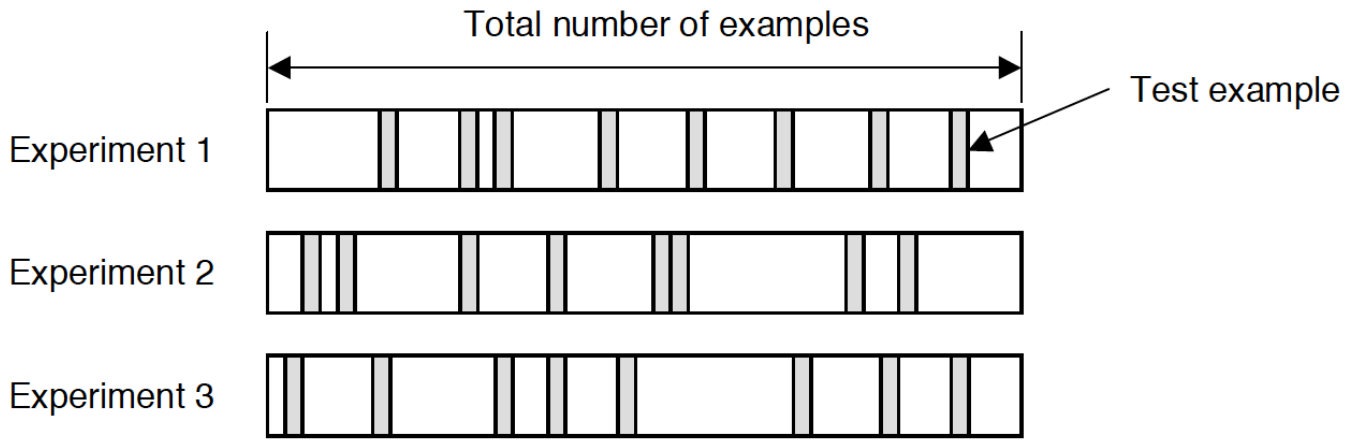
K-fold cross validation

- Consiste en dividir la BD en k subconjuntos de igual tamaño
- De los k subconjuntos, uno se mantiene como CT y el resto como CE
- Este proceso se repite k veces utilizando en cada ocasión un subconjunto diferente como CT



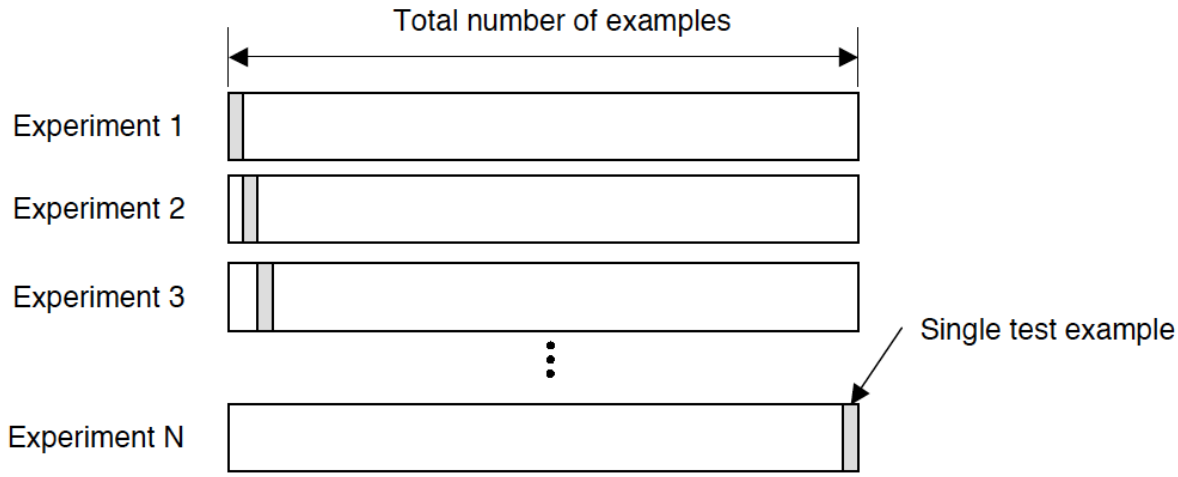
Random subsampling cross validation

- Consiste en dividir la BD en k subconjuntos CE/CT con muestreo aleatorio sin reemplazamiento
- En cada división se extrae el modelo con CE y se evalúa con CT
- La estimación se promedia



Leaving one out

- Es un caso especial de k-fold en el que *k* es igual al número de registros
  - El proceso es determinista
  - Se utiliza el máximo posible de datos para la inducción del clasificador
  - Se utiliza en BBDD muy pequeñas, debido a su alto costo computacional



Bootstrap

- Muestreo con reemplazo
- A partir de una BD con *n* registros se obtiene un CE con *n* casos
- Como CT se utilizan los registros de la BD no seleccionados para el CE

¿Cuántos casos habrá en CT? ¿qué porcentaje respecto a *n*?

- La probabilidad de que se elija un registro es *1/n*
- Se hacen *n* extracciones →la probabilidad de que un ejemplo no sea elegido es

$$\left(1-\frac{1}{n}\right)^{n} \approx e^{-1} = 0.368$$

- El CE tendrá aproximadamente el 63.2% de los registros de la BD y el CT el 36.8 %

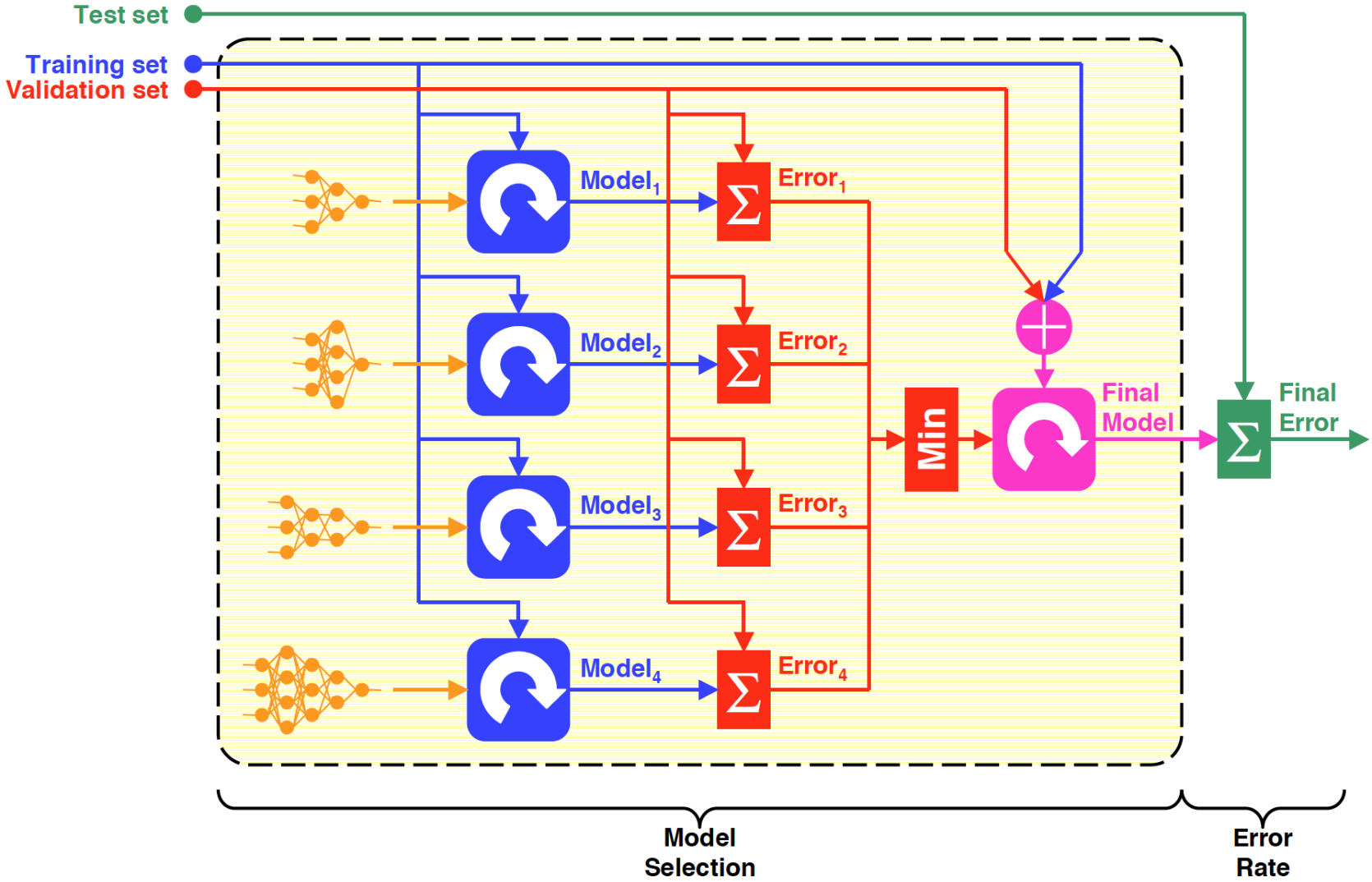
- Esta técnica se conoce como 0.632 bootstrap
- El error sobre el CT suele ser bastante pesimista por lo que se corrige

$$error = 0.632 \cdot error_{CT} + 0.368 \cdot error_{CE}$$

¿Cuántas particiones CE/CT son necesarias?

- Alto número de particiones CE/CT:
  - Sesgo de la estimación del error pequeño → Estimador bastante preciso
  - Alta varianza del estimador
  - Costo computacional alto
- Bajo número de particiones CE/CT:
  - Sesgo alto de la estimación
  - Varianza baja
  - Costo computacional reducido
- La elección del número de particiones depende del tamaño del conjunto de datos
  - Para datasets grandes, 3 es un número es adecuado
  - Para datasets pequeños, mayor e incluso leaving one out
  - Un valor común para k-fold cross validation es K=5 ó 10

¿Qué es predicción? ▶ Componentes ▶ Evaluación



Analítica predictiva

- Motivación
- ¿Qué es predicción?
- **Diseño del modelo de predicción**
- Medidas de error
- Preprocesamiento de datos
- Métodos de analítica predictiva

Diseño del modelo de predicción

Etapas en el proceso de diseño de un modelo de predicción:

- Elección de la medida de error / medidas de calidad
- División del conjunto de datos en: training y test
- Para determinar el mejor modelo:
  - Sobre el conjunto de training, utilizar validación cruzada
  - Preprocesar los datos, si es necesario
  - Obtener diferentes modelos de predicción, determinar los párametros, compararlos y elegir el mejor
- Para el mejor modelo, obtener la predicción en los datos de test

Debe existir un conjunto de datos (test) separado desde el inicio que nunca se utilice para entrenar/ajustar/elegir el modelo

Analítica predictiva

- Motivación
- ¿Qué es predicción?
- Diseño del modelo de predicción
- **Medidas de error**
- Preprocesamiento de datos
- Métodos de analítica predictiva

Medidas de error

- Medidas de error para problemas con variable a predecir de tipo discreto
  - Clasificación
- Medidas de error para datos con variable a predecir de tipo continuo
  - Regresión
- Medidas de error para datos con variable continua y dependencia temporal
  - Series temporales

Medidas de error ► Clasificación

- Conceptos básicos en problemas de clasificación
  - **True positive (TP):** ejemplos clasificados como positivos correctamente
  - **False positive (FP)**: ejemplos clasificados incorrectamente como positivos
  - **True negative (TN)**: ejemplos clasificados como negativos de forma correcta
  - **False negative (FN):** ejemplos clasificados como negativos de forma incorrecta

Matriz de confusión

		Clasificación como	
		Si	No
Clase real	SI	Verdadero positivo (TP)	Falso negativo (FN)
	NO	Falso Positivo (FP)	Verdadero Negativo (TN)

Medidas de error ▶ Clasificación

- Accuracy $$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

- Sensitivity/recall $$Sensitivity(recall) = \frac{TP}{TP + FN}$$
  - Potencia que queden pocos positivos mal clasificados

- Specificity $$Specificity = TN / (FP + TN)$$
  - Potencia que existan pocos negativos mal clasificados como positivos

- $$Positive\,\mathrm{Pr}\,editiveValue = TP / (TP + FP)$$

- $$Negative\,\mathrm{Pr}\,edictiveValue = TN / (FN + TN)$$

Medidas de error ► Regresión

- Mean squared error (MSE)

$$MSE = \frac{\sum_{i=1}^{n} (v_i - v_i')^2}{n}$$

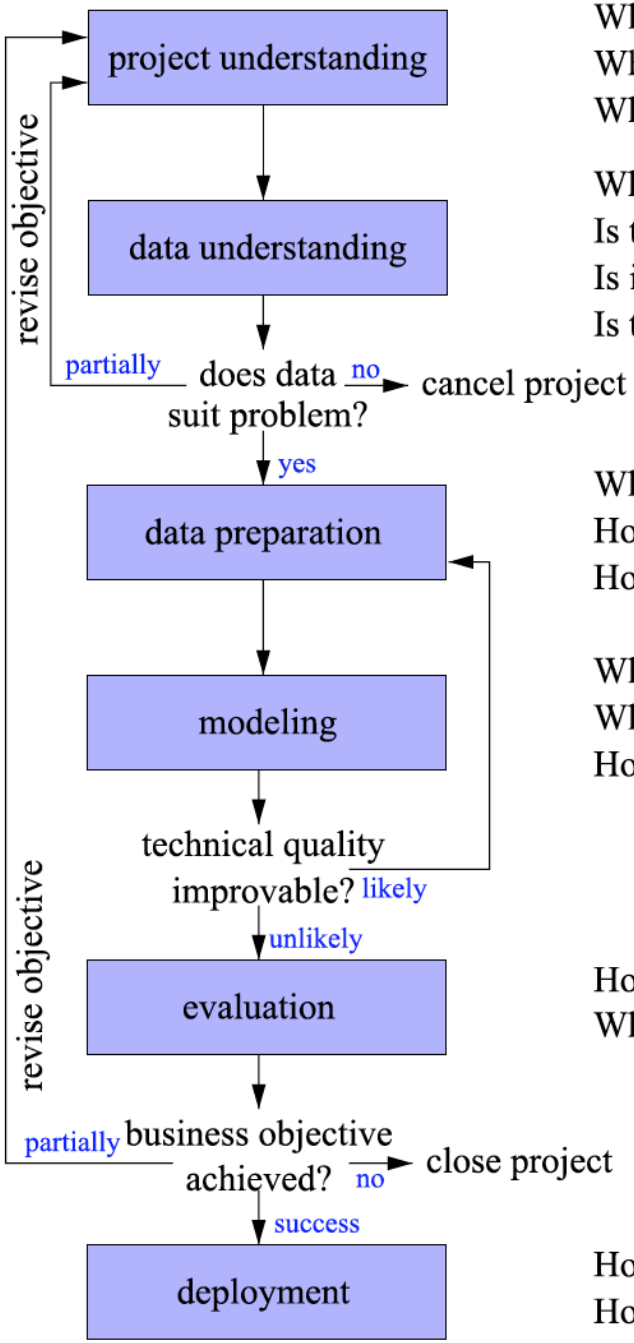
  - Sensible a outliers

- Root mean squared error (RMSE)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (v_i - v_i')^2}{n}}$$

En resumen…

https://www.youtube.com/watch?v=PoD84TVdD-4



What exactly is the problem, the expected benefit?
What should a solution look like?
What is known about the domain?

What data do we have available?
Is the data relevant to the problem?
Is it valid? Does it reflect our expectations?
Is the data quality, quantity, recency sufficient?

Which data should we concentrate on?
How is the data best transformed for modeling?
How may we increase the data quality?

What kind of model architecture suits the problem best?
What is the best technique/method to get the model?
How well does the model perform technically?

How good is the model in terms of project requirements?
What have we learned from the project?

How is the model best deployed?
How do we know that the model is still valid?

Visión general del proceso CRISP-DM

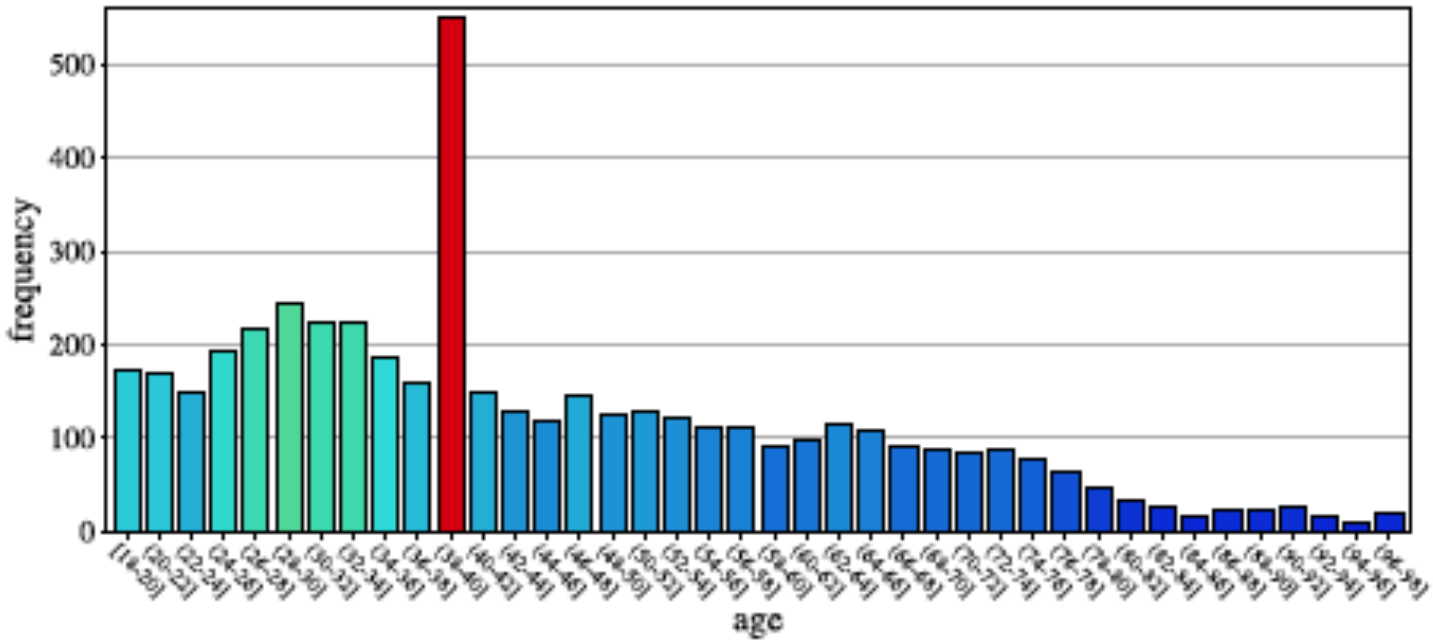
M.R. Berthold, C. Borgelt, F. Höppener, F. Klawonn. Guide to Intelligent Data Analysis. How to Intelligently Make Sense of Real Data. Springer, 2010

Analítica predictiva

- Motivación
- ¿Qué es predicción?
- Diseño del modelo de predicción
- Medidas de error
- **Preprocesamiento de datos**
- Métodos de analítica predictiva

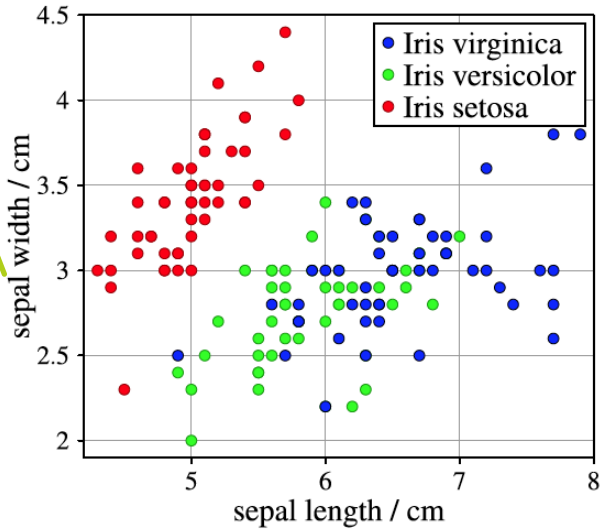
Preprocesamiento ► Observación de los datos

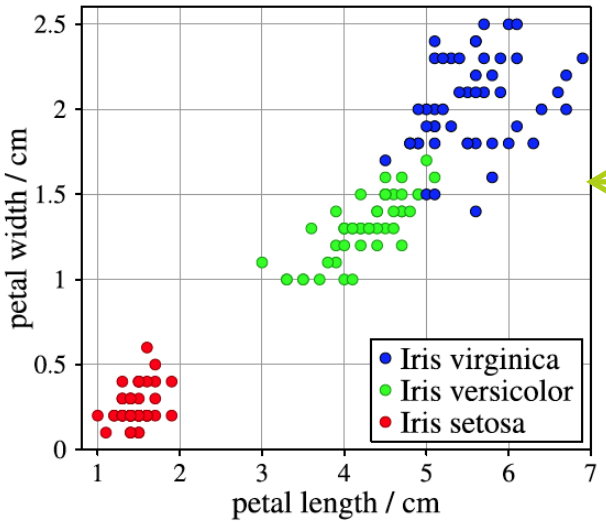
- Antes del pre-procesamiento es necesario analizar las características de los datos
  - Permitirá conocer mejor el problema, detectar posibles datos erróneos, dependencias, outliers, valores perdidos, etc.
- Uso de técnicas de visualización



Se detecta una frecuencia inusual que tras comprobar la recogida de datos corresponde a valores perdidos etiquetados incorrectamente

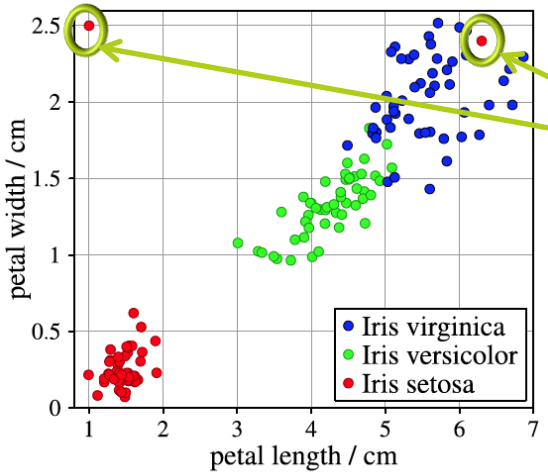
Preprocesamiento ► Observación de los datos





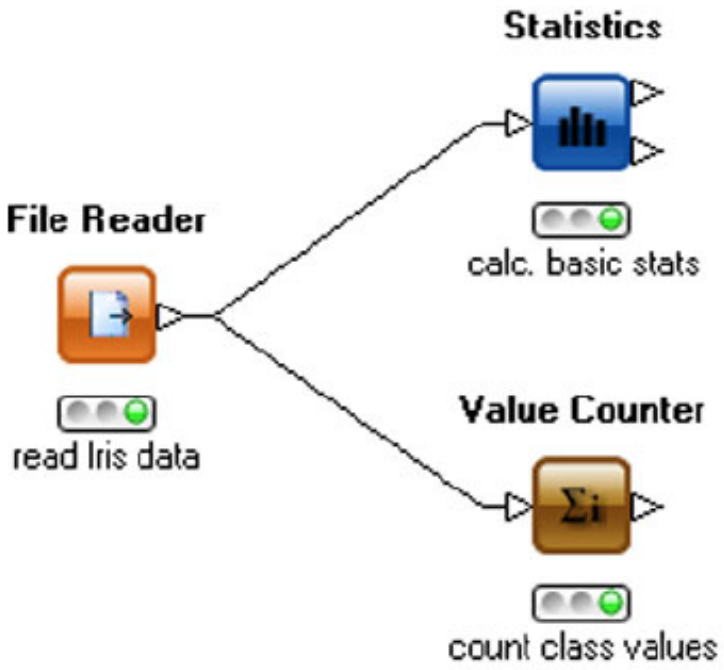
Es más fácil diferenciar las cases con estas variables

Con estas dos variables, es más difícil distinguir entre clases



outliers

Preprocesamiento ► Observación de los datos



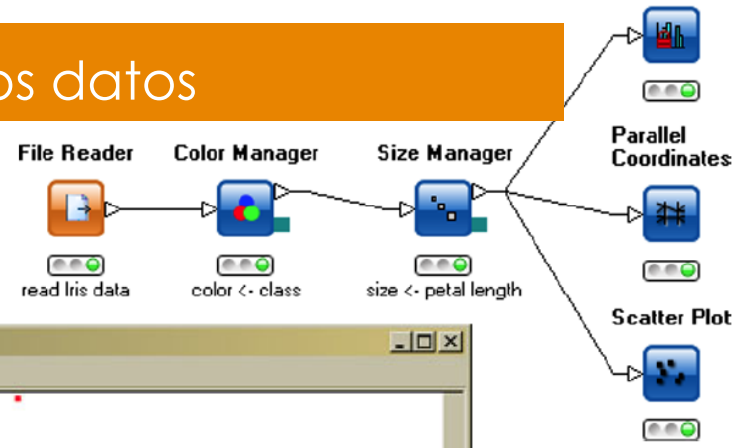
Statistics View - 0:6 - Statistics (calc. basic...

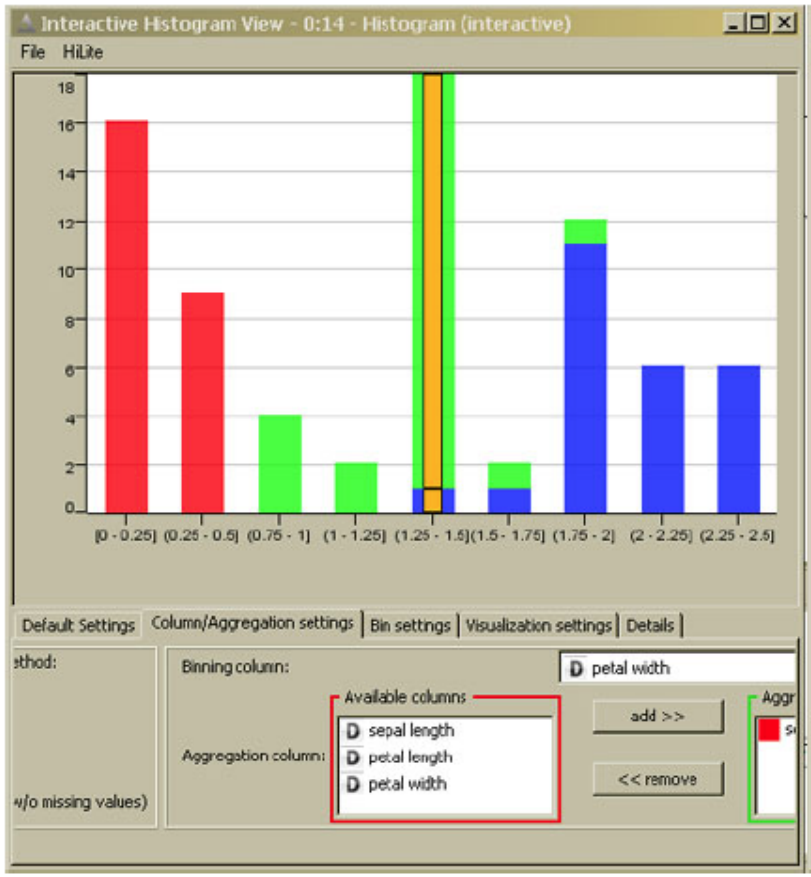
File

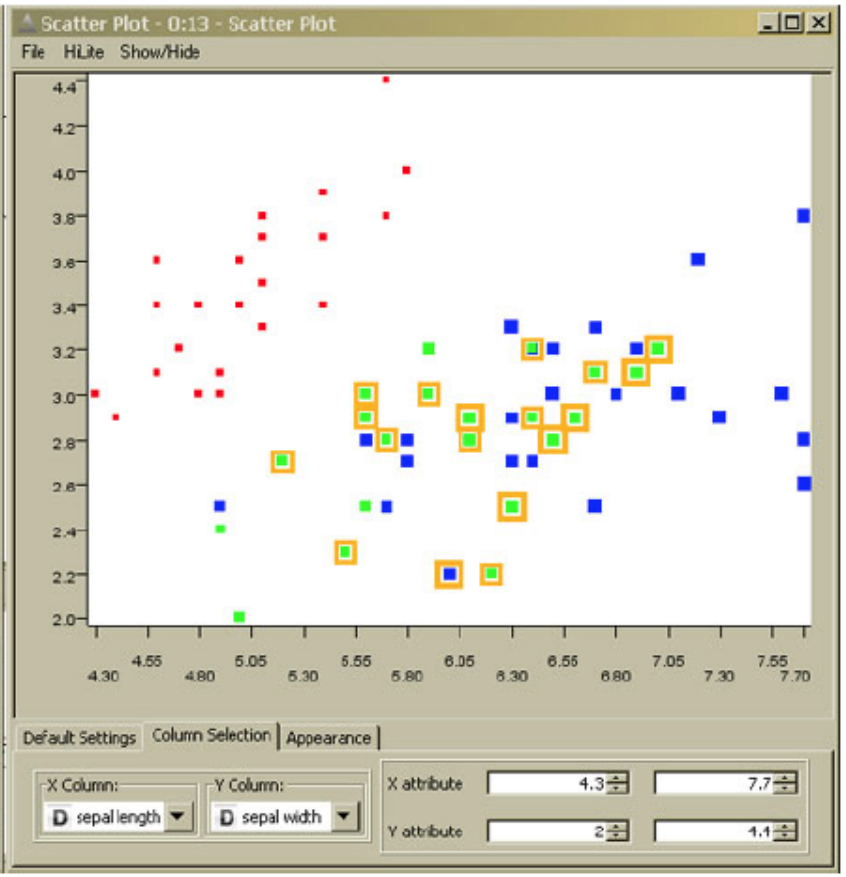
Numeric columns | Nominal columns

Row ID	D sepal le...	D sepal w...	D petal le...
Minimum	4.3	2	1
Maximum	7.7	4.4	6.9
Mean	5.872	3.061	3.804
Std. deviation	0.857	0.468	1.815
Variance	0.734	0.219	3.294
Overall sum	440.4	229.6	285.3
No. missings	0	0	0

Preprocesamiento ► Observación de los datos







Preprocesamiento de datos

Antes de aplicar los métodos de analítica predictiva es necesario preparar los datos

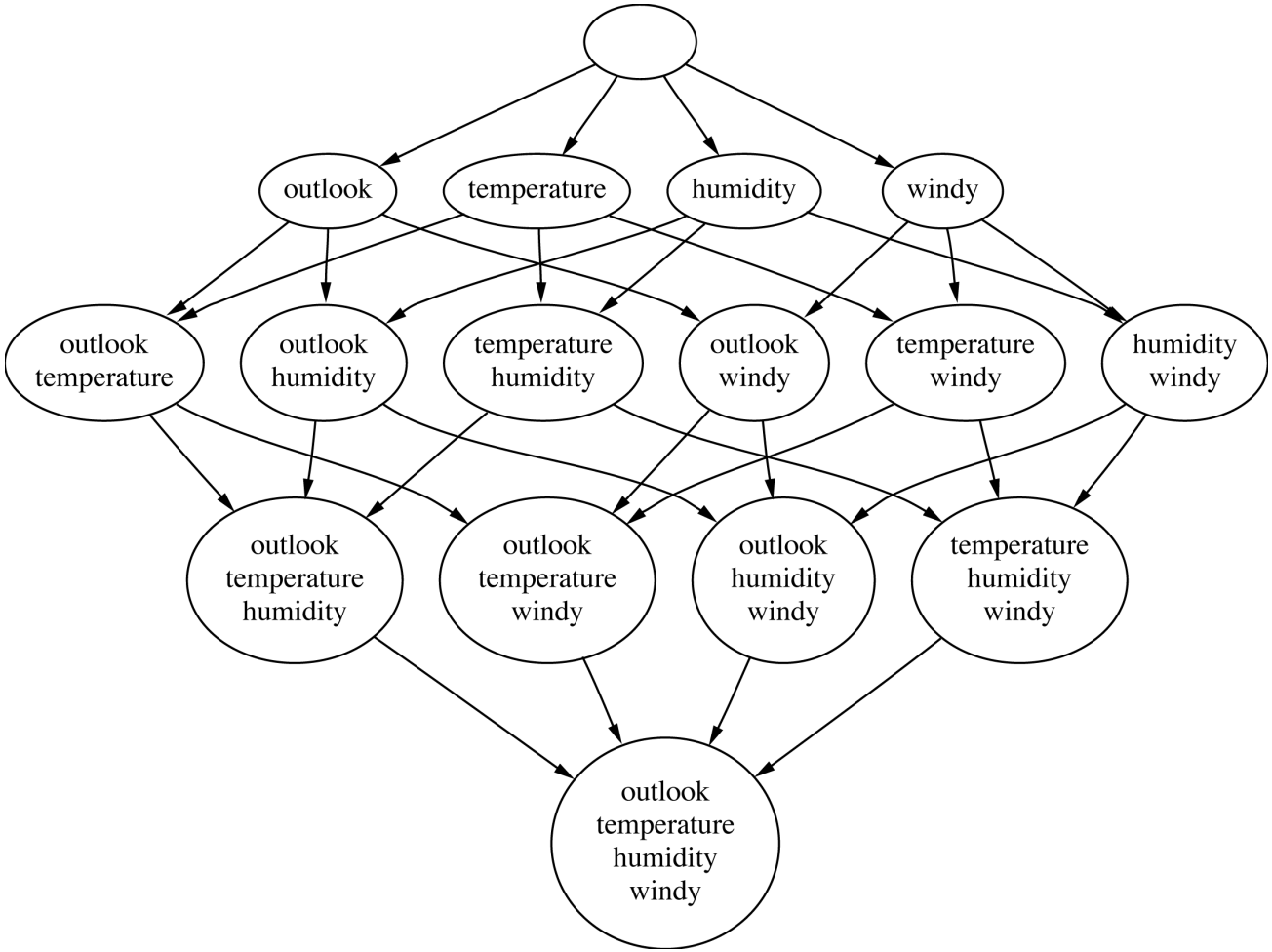
- Reducción de datos
  - Selección de variables
  - Selección de instancias
  - Generación de variables
  - Generación de instancias
- Datos imperfectos: missing values y noise data
- Transformación de datos: normalización, construcción de atributos

- **Objetivo**: seleccionar un subconjunto óptimo del conjunto total de variables disponibles

- **¿Por qué es necesario?**
  - Más atributos no significa más éxito en la clasificación
  - Trabajar con menos variables reduce la complejidad del problema y disminuye el tiempo de ejecución
  - Eliminando variables irrelevantes y redundantes se reduce a priori la varianza del modelo puesto que disminuye la posibilidad de sobreajuste

- **Componentes de un algoritmo de selección:**
  - Una función de evaluación que permita comparar dos subconjuntos de variables
  - Una estrategia que permita seleccionar subconjuntos


PATTERNS
DATA
MINING
LEARNING
ANALYSIS
CUSTOMER
CUSTOMERS
PROCESS


outlook
temperature
humidity
windy
outlook temperature
outlook humidity
temperature humidity
outlook windy
temperature windy
humidity windy
outlook temperature humidity
outlook temperature windy
outlook humidity windy
temperature humidity windy
outlook temperature humidity windy

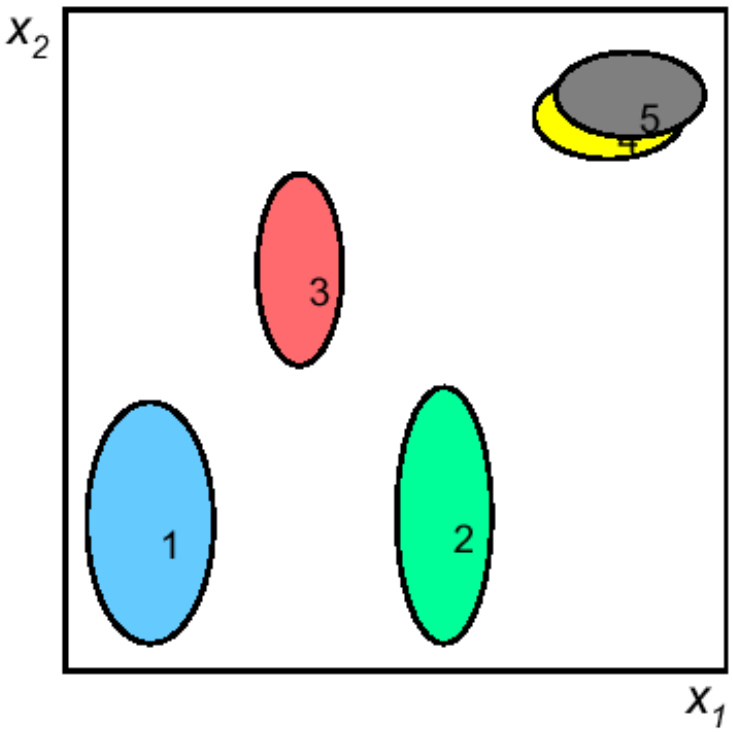
Selección de las k variables mejores (ranking)

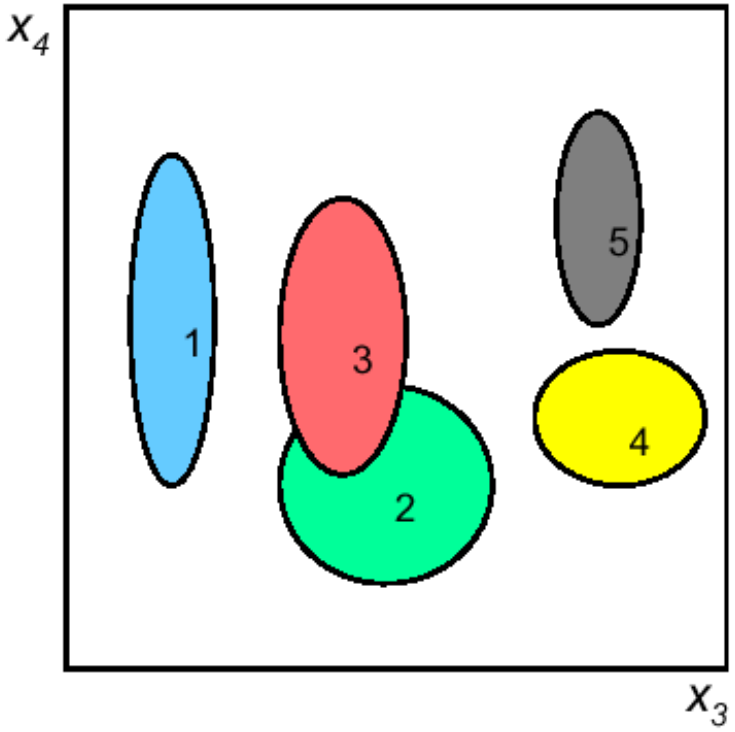
- Se asume que un conjunto de variables es tan bueno como lo sean *individualmente* las variables que lo componen
- Se utiliza una **función de evaluación** individual para cada variable
- **Estrategia de selección**: Elegir las k mejores

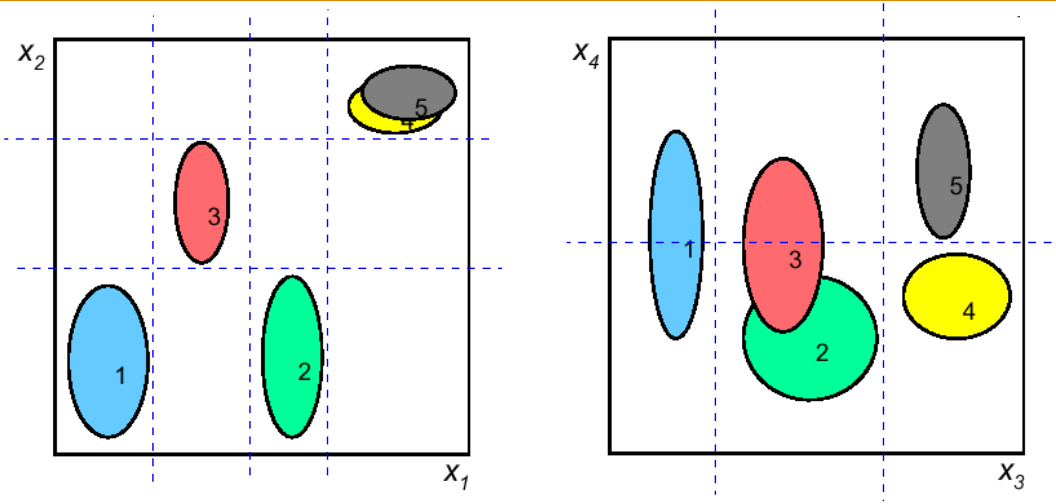
- **Función de evaluación:**
  - Correlación entre la variable evaluada y la de clase
  - Ganancia de información
  - Gain ratio, …

Ejemplo de selección de las 2 mejores variables

¿Qué subconjunto de 2 variables seleccionamos?







X1: [1,2,3,{4,5}]
X2: [{1,2},3,{4,5}]
X3: [1,{2,3},{4,5}]
X4: [{1,2,3},4,5]

Es razonable que la función objetivo ofrezca un resultado como
f(X1) >f(X2)≈f(X3)>f(X4)

→ Elegir {X1,X2} o {X1,X3}

La mejor elección es: {X1,X4}

Algoritmo Relief

- Estima la contribución de una variable A en la clasificación mediante un peso
  - Si el peso es grande, la variable es útil
- Los pesos se determinan de forma incremental y consideran la interacción entre variables

Dado un ejemplo x

- si hay dos ejemplos cerca (uno de la clase y otro no), las variables consideradas no son de ayuda en la predicción por lo que los pesos no se alteran
- Si el ejemplo más cercano es el de la clase y el de la clase contraria está más lejos, el peso se incrementa

Algorithm Relief($\mathcal{D}, \mathcal{A}, C$) $\rightarrow w[\cdot]$

input: data set $\mathcal{D}$, $|\mathcal{D}| = n$,
attribute set $\mathcal{A}$, target variable $C \in \mathcal{A}$
output: attribute weights $w[A]$, $A \in \mathcal{A}$

1 set all weights $w[A] = 0$
2 **for all** records $\mathbf{x} \in \mathcal{D}$
3 find nearest hit $\mathbf{h}$ (same class label $\mathbf{x}_C = \mathbf{h}_C$)
4 find nearest miss $\mathbf{m}$ (different class label)
5 **for all** $A \in \mathcal{A}$:
6 $w[A] = w[A] - \frac{\text{diff}(A,\mathbf{x},\mathbf{h})}{n} + \frac{\text{diff}(A,\mathbf{x},\mathbf{m})}{n}$

where

$$\text{diff}(A, \mathbf{x}, \mathbf{y}) = \begin{cases} 0 & \text{if } \mathbf{x}_A = \mathbf{y}_A \\ 1 & \text{otherwise} \end{cases}$$

for categorical attributes A and

$$\text{diff}(A, \mathbf{x}, \mathbf{y}) = \frac{|\mathbf{x}_A - \mathbf{y}_A|}{\max(A) - \min(A)}$$

for numerical attributes A.

Selección de mejor subconjunto

- **Función de evaluación:** determina la calidad de subconjuntos de variables
  - **Filtro**: Se evalúa mediante medidas de separabilidad de clases, consistencia, correlaciones, medidas basadas en teoría de la información, etc.
  - **Envolventes**: Se evalúa el subconjunto de variables en base a la calidad del modelo derivado a partir de ellas.

- **Estrategia de selección:**
  - Algoritmos secuenciales: Añaden o eliminan variables
    - Selección hacia delante, hacia atrás, MásMenos-r, bidireccional, secuencial flotante…
  - Exponenciales: branch and bound, beam search,…
  - Estocásticos: Ascensión de colinas con reinicios, enfriamiento simulado, algoritmos genéticos, …

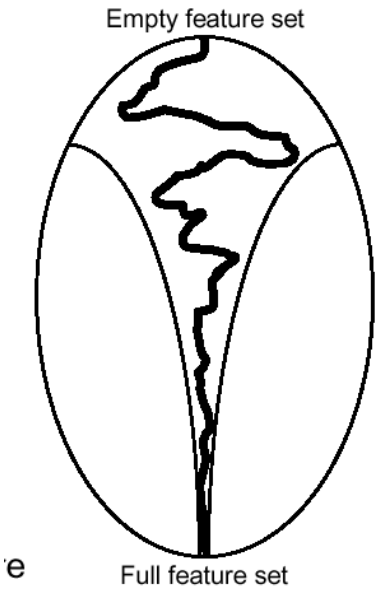
Selección hacia delante

La selección *forward* comienza con el conjunto vacío y de forma secuencial añade al subconjunto actual S el atributo X_i que maximiza $f(S,X_i)$

1. Comenzar con $S=\Phi$
2. Seleccionar la variable

$$X^+ = \arg\max_{X \in U-S} f(S \cup X)$$

3. $S=S\ U\ \{X^+\}$
4. Ir al paso 2



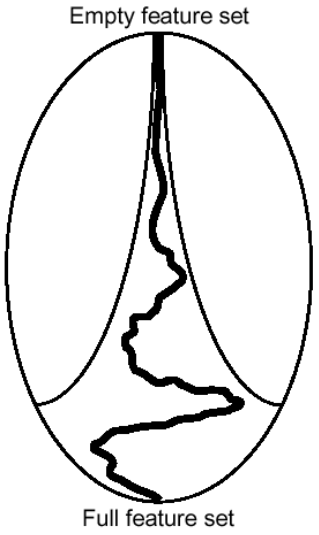
Selección hacia atrás

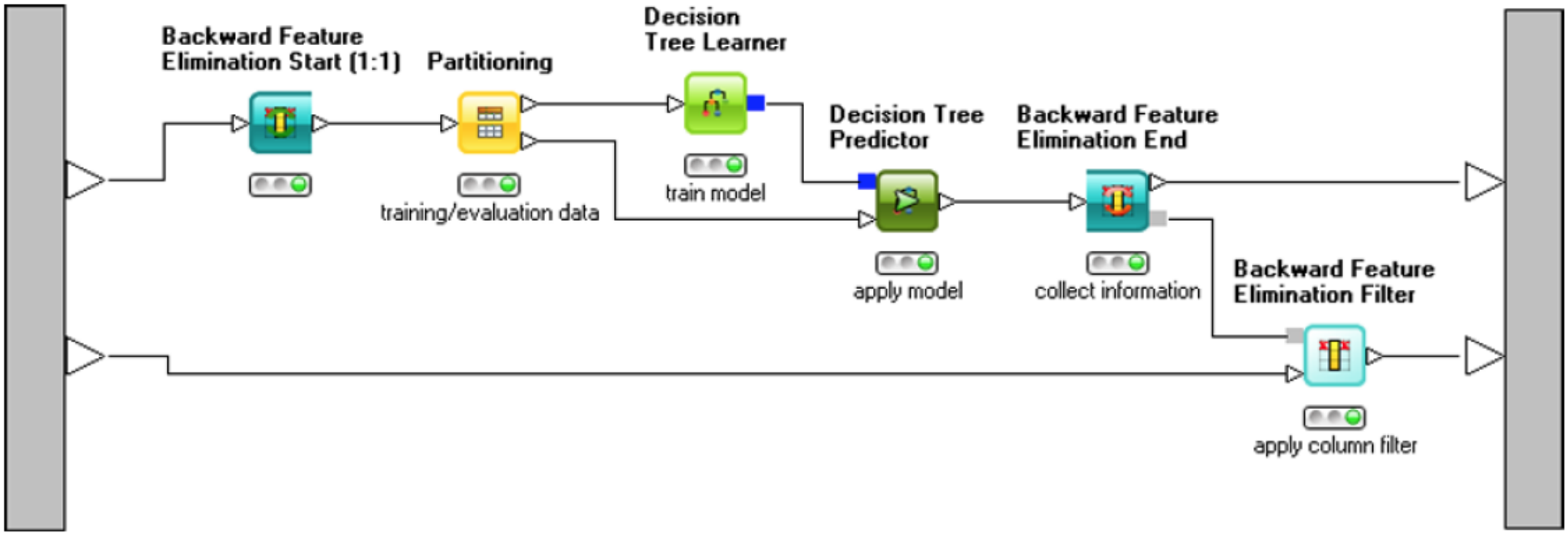
- La selección *backward* comienza con el conjunto completo U y de forma secuencial elimina del subconjunto actual S el atributo X que decrementa menos f(S-X)

1. Comenzar con $S=U$
2. Seleccionar la variable X^-

$$X^- = \arg\max_{X \in S} f(S - X)$$

3. $S=S-\{X^-\}$
4. Ir al paso 2





Selección de características envolvente con árboles de decisión

¿Por qué es necesaria la reducción del conjunto de ejemplos?

5 MB en 1956

… en la actualidad

¿Por qué es necesaria la reducción del conjunto de ejemplos?

- Para reducir el costo
- Asegurar la "**actualidad**" de los datos
  - Para predecir en entornos cambiantes se conveniente utilizar datos recientes
- **Representatividad:** En ocasiones la población de interés es diferente a la población disponible
  - Eligiendo con más frecuencia casos poco representados se compensa el desbalanceo y se incrementa la representatividad de la muestra
  - En casos de muestras pequeñas es importante la estratificación

¿De qué tamaño debe ser la muestra?

Aunque hay resultados teóricos para algunas situaciones (desigualdad de Hoeffding) no hay una respuesta

- Cuanto más pequeña sea la muestra, menos representativa
- Cuanto más complejo sea el modelo a aprender, más parámetros tendrá y necesitará más ejemplos para obtener una estimación robusta

Puesto que la elección de la técnica de análisis predictivo es un compromiso entre interpretabilidad, precisión, tiempo de ejecución, ..

¿por qué no reducir el dataset a aquellos ejemplos que sean importantes para la técnica?

Un ejemplo clásico: **Muestreo aleatorio**

Objetivo: seleccionar un conjunto *M* del total de los *N* casos presentes en la BD original (*M*<*N*)

Variantes:

- Muestreo simple con o sin reemplazamiento
- Muestreo estratificado: mantiene la distribución por clases deseada
  - La presente en el dataset original
  - Igual para todas las clases
- Adaptativo

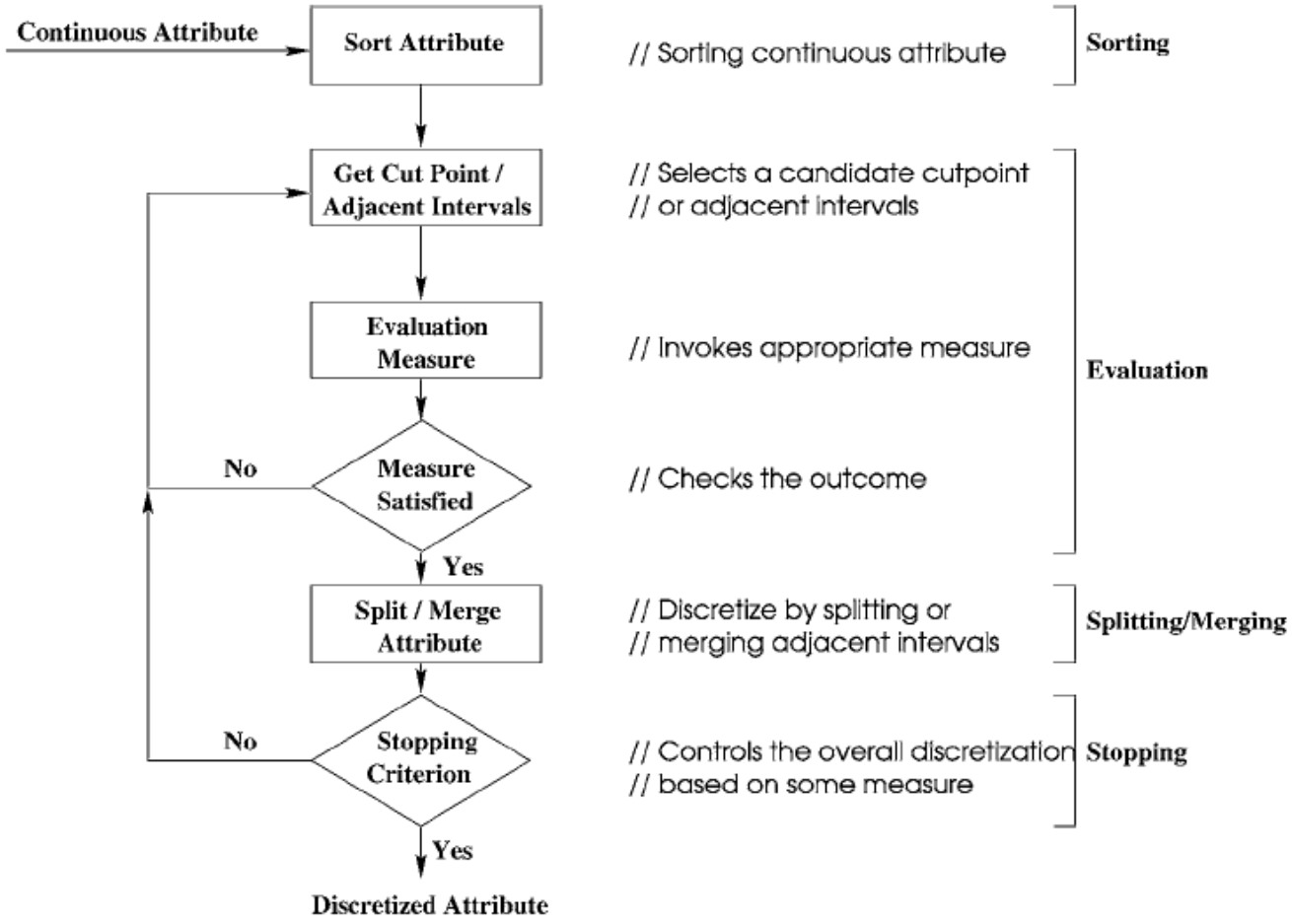
Preprocesamiento ▶ Reducción de datos ▶ **Discretización**

- Es una forma de transformar los datos que implica reducción
  - Permite representar la información de una forma más concisa

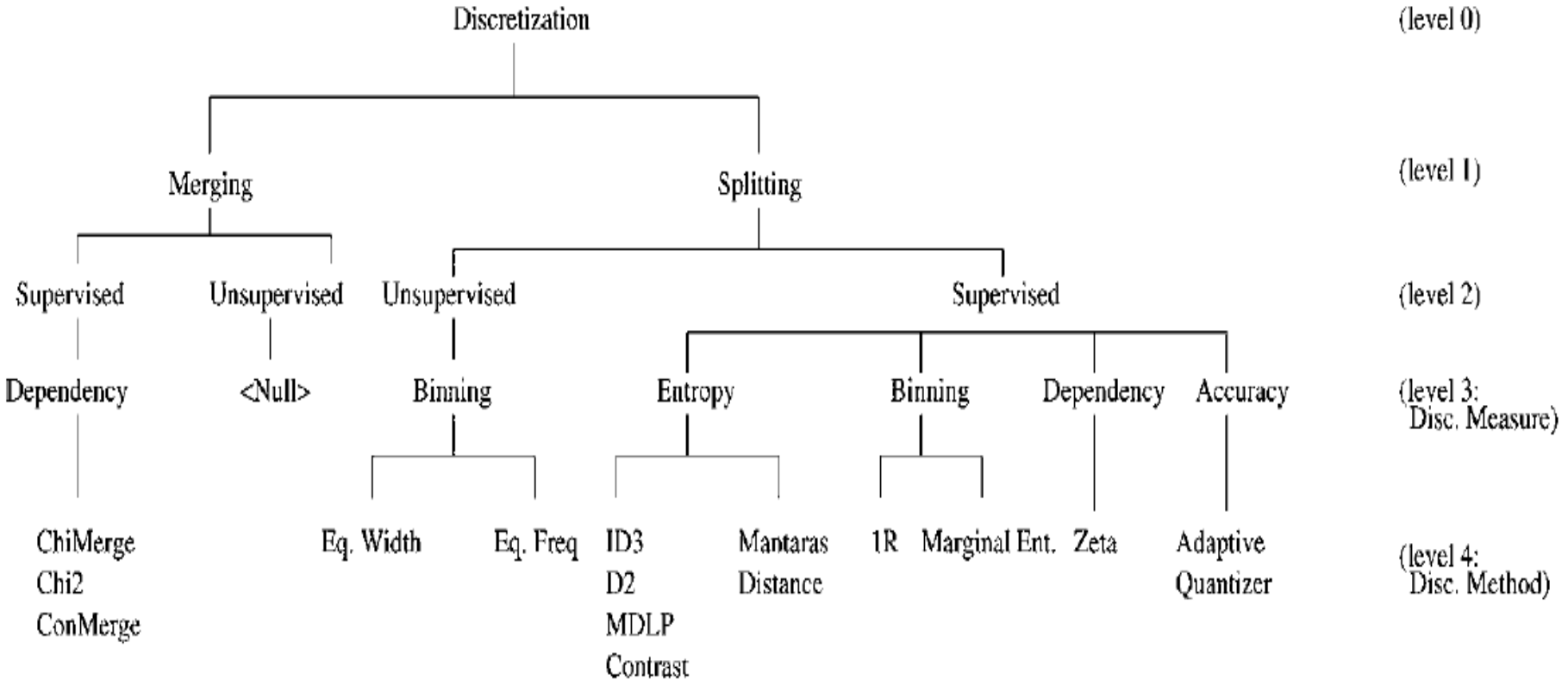
Algunas alternativas:

- **Supervisados vs. no supervisados**
- **Dinámicos vs. Estáticos**
- **Locales vs. globales:** Centrados en una subregión del espacio de instancias o considerando todas ellas
- **Top-down vs. bottom-up**: Empiezan con una lista vacía o llena de puntos de corte
- **Directos vs. Incrementales**: Usan o no un proceso de optimización posterior

Preprocesamiento ▶ Reducción de datos ▶ **Discretización**




PATTERNS
DATA
MINING


Discretization
(level 0)
Merging
Splitting
(level 1)
Supervised
Unsupervised
Unsupervised
Supervised
(level 2)
Dependency
<Null>
Binning
Entropy
Binning
Dependency
Accuracy
(level 3: Disc. Measure)
ChiMerge
Chi2
ConMerge
Eq. Width
Eq. Freq
ID3
D2
MDLP
Contrast
Mantaras Distance
1R
Marginal Ent.
Zeta
Adaptive Quantizer
(level 4: Disc. Method)

Ejemplo de discretizador sencillo: **Igual anchura o frecuencia**

Crea intervalos dada una aridad predefinida sin considerar nada más (***binning* no supervisado**).

- **En igual anchura**, los intervalos deben tener la misma anchura y no se preocupa del número de valores que caen en cada uno de ellos.
- **En igual frecuencia**, cada intervalo contiene exactamente el mismo número de valores continuos, por lo que los intervalos no resultan de la misma anchura.

Limitaciones: Sensibles a la aridad dada

Preprocesamiento de datos

- Reducción de datos
  - Selección de variables
  - Selección de instancias
  - Generación de variables
  - Generación de instancias
  - Discretización
- **Datos imperfectos: missing values y noise data**
- Transformación de datos: normalización, construcción de atributos

- **Objetivos**:
  - mejorar la calidad de los datos
  - detectar/completar **valores perdidos**
  - suavizar el **ruido** de los datos
  - identificar o eliminar **outliers**
- Algunos algoritmos de DM tienen métodos propios para tratar con datos imperfectos
  - Es habitual realizar previamente la limpieza de los datos

W. Kim, B. Choi, E.-D. Hong, S.-K. Kim. A taxonomy of dirty data. Data Mining and Knowledge Discovery 7, 81-99, 2003.

Previamente: corrección de inconsistencias, erratas, abreviaciones, formatos variantes, etc

- En caso de sensitividad a mayúsculas, transformar
- Eliminar espacios en blanco y caracteres no imprimibles
- Fijar el formato de números y fechas
  - Dividir variables con información mezclada (p.e. "mantequilla, 100g" en "mantequilla", 100.00)
  - Reemplazar abreviaturas
  - Normalizar la escritura de direcciones, nombres, enfermedades, síntomas, etc.
  - Uniformizar unidades de medida
  - …

¿Por qué necesitamos tratar los valores perdidos?

La mayoría de los métodos de MD no pueden trabajar con ellos

Opciones:

- Ignorar los ejemplos.
  - Sencillo pero elimina mucha informaciónRellenar manualmente los datos. En general es impracticable
- Utilizar una constante global para la sustitución. P.e. “desconocido”,”?”
- Imputación:
  - Rellenar utilizando la media/desviación del resto de las tuplas
  - Rellenar utilizando la media/desviación del resto de las tuplas pertenecientes a la misma clase
  - Rellenar utilizando kNN
  - Rellenar con el valor más probable

Posición	Valor original	Pos. 11 perdida	Preservar la media	Preservar la desviación
1	0.0886	0.0886	0.0886	0.0886
2	0.0684	0.0684	0.0684	0.0684
3	0.3515	0.3515	0.3515	0.3515
4	0.9875	0.9875	0.9875	0.9875
5	0.4713	0.4713	0.4713	0.4713
6	0.6115	0.6115	0.6115	0.6115
7	0.2573	0.2573	0.2573	0.2573
8	0.2914	0.2914	0.2914	0.2914
9	0.1662	0.1662	0.1662	0.1662
10	0.4400	0.4400	0.4400	0.4400
11	0.6939	????	0.3731	0.6622
Media	0.4023	0.3731	0.3731	
SD	0.2785	0.2753		0.2753
Error en la estimación			0.3208	0.0317

X	Y	Clase
a	a	+
a ?	n	+
n	a	-
n	n	-
n	a	+

- Estimar por el valor más probable (la moda)

X = n → error

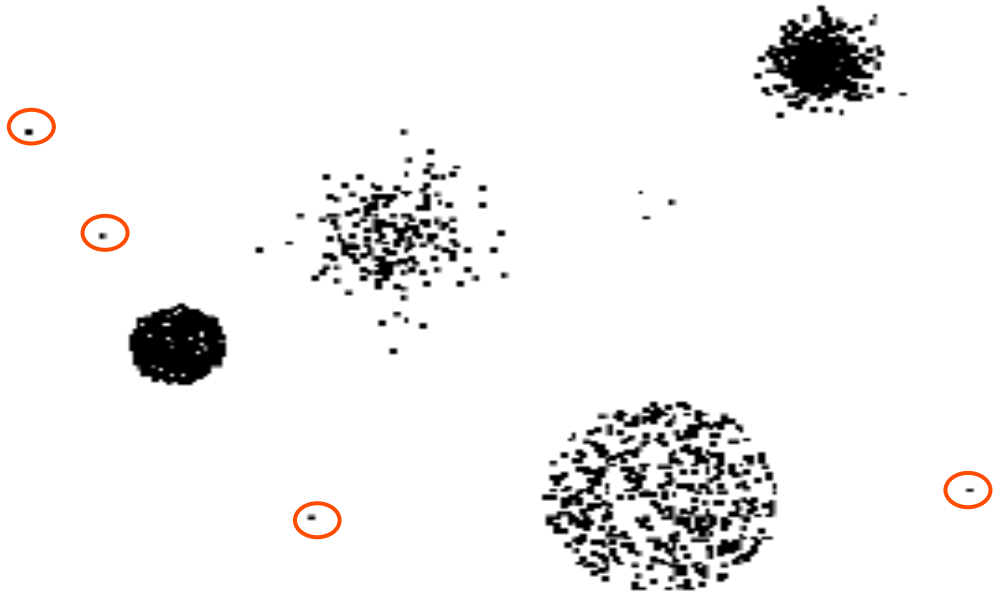
- Estimar por el valor más probable (la moda) dentro de la clase (+)

X = a (prob. 0.5) ó X=n (prob. 0.5)

→ No resuelve nada

- **Valor erróneo <> valor anómalo**
- El **ruido** es un **error** o varianza aleatoria en la medición de una variable
- ¿Cómo se detectan datos erróneos?
  - Nominales → valor fuera del formato o rango.
  - Numéricos → Buscar datos anómalos y estudiar si son realmente anómalos o erróneos.
- ¿Cómo se elimina el ruido? Mediante técnicas de suavizado.
  - Binning: Se suavizan valores ordenados consultando sus vecinos
  - Regresión: Los datos se suavizan ajustándolos a una función con técnicas de regresión.

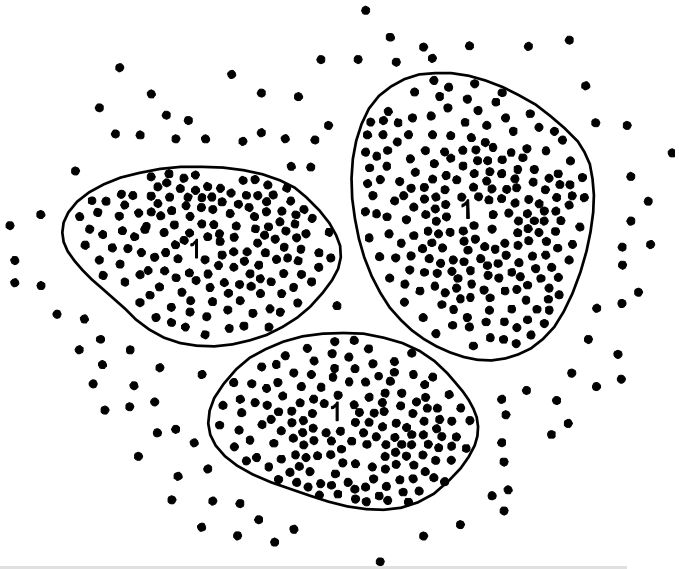
- **Outliers:** Datos con características considerablemente diferentes de la mayoría de los datos del conjunto



- **Outliers**: correctos aunque sean anómalos estadísticamente
- Pueden ser un inconveniente para métodos basados en ajuste de pesos (p.e. AANN).

- **Técnicas de detección**:
  - Definir una distancia y ver los individuos con mayor distancia media al resto de individuos.

  - Clustering parcial: los datos se agrupan en clusters y los datos que queden fuera pueden considerarse outliers



- Combinación de inspección humana y automática.

  Utilizar técnicas automatizadas (p.e. basadas en la teoría de la información) para identificar casos "extraños" y el experto humano trabaja sólo sobre estos datos

- La no detección de un dato anómalo puede ser un problema importante si el atributo se normaliza posteriormente, ya que la mayoría de datos estarán en un rango pequeño y puede haber poca precisión o sensibilidad para algunos métodos de DM

Preprocesamiento de datos

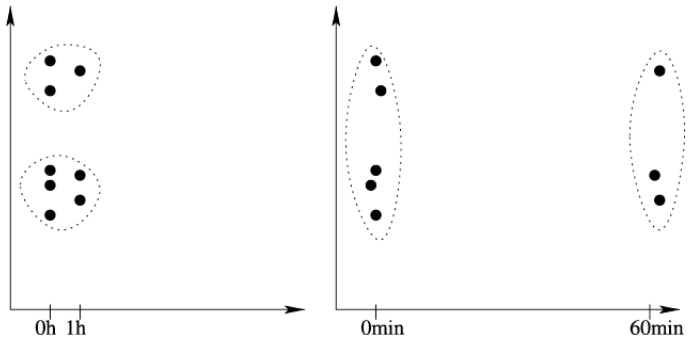
- Reducción de datos
  - Selección de variables
  - Selección de instancias
  - Generación de variables
  - Generación de instancias
  - Discretización
- Datos imperfectos: missing values y noise data
- **Transformación de datos: normalización, construcción de atributos**

Preprocesamiento de datos ▶ Transformación de datos

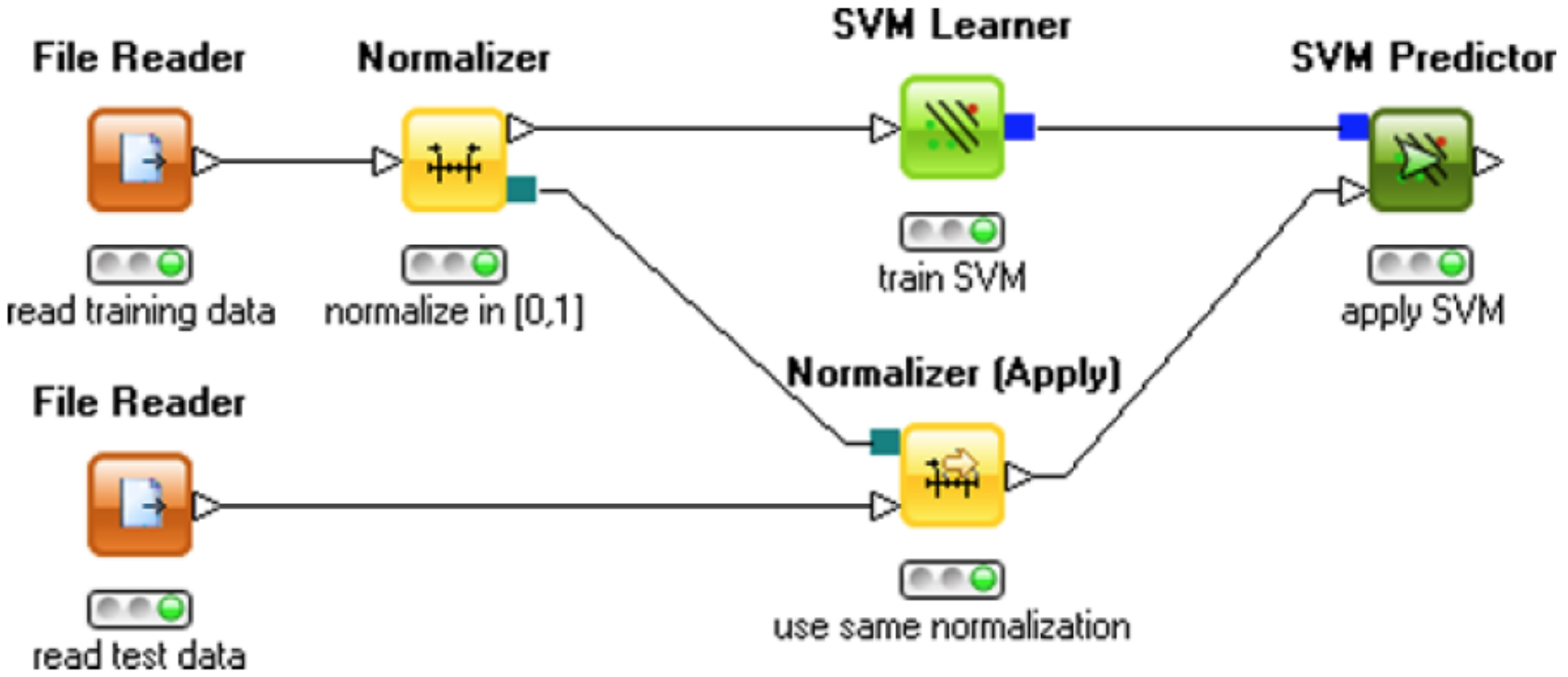
Objetivo: poner los datos de la mejor forma posible para la aplicación de los algoritmos de DM

Parte de las técnicas de transf. se consideran técnicas de reducción

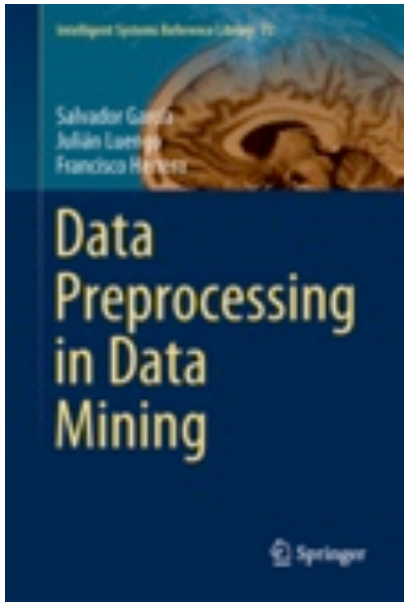
- Agregación
- Generalización: Obtener datos de más alto nivel a partir de los actuales con jerarquías de conceptos
  - Calles → ciudades
  - Edad numérica → {joven, adulto, mediana-edad, anciano}
- Normalización: Necesaria para técnicas basadas en distancia
- Construcción de atributos
- Discretización



Preprocesamiento de datos ▶ Transformación de datos



Preprocesamiento de datos

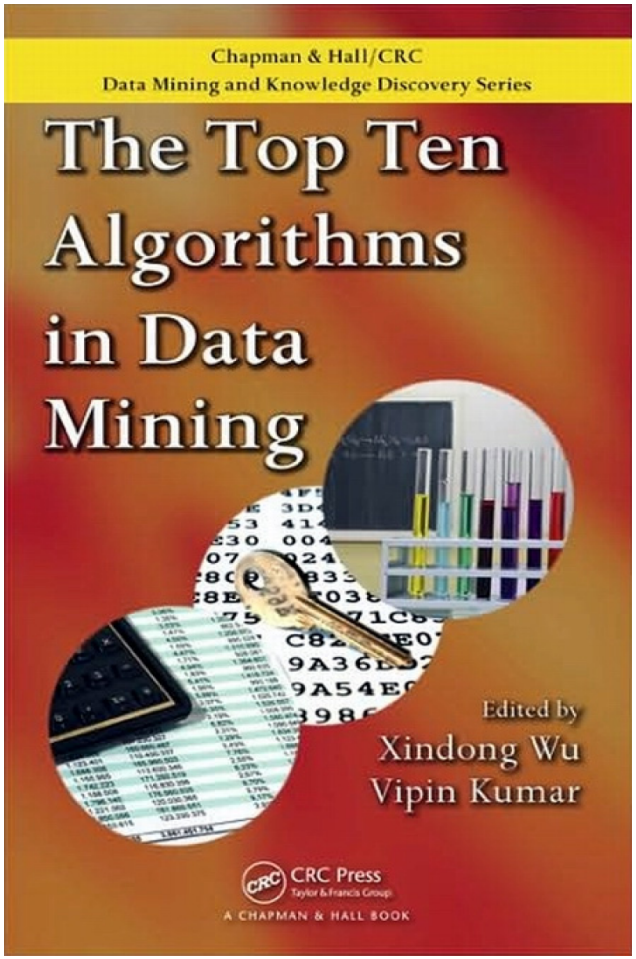


S. García, J. Luengo, F. Herrera. Data Preprocessing in Data Mining. Springer, 2014

Analítica predictiva

- Motivación
- ¿Qué es predicción?
- Diseño del modelo de predicción
- Medidas de error
- Preprocesamiento de datos
- **Métodos de analítica predictiva**

Métodos de Minería de Datos más utilizados



1. C4.5
2. K-Means
3. SVM: Support Vector Machines
4. A priori
5. EM
6. PageRank
7. Adaboost
8. kNN: k-Nearest Neighbors
9. Naïve-Bayes
10. CART: Classification and Regression Trees

Métodos de analítica predictiva

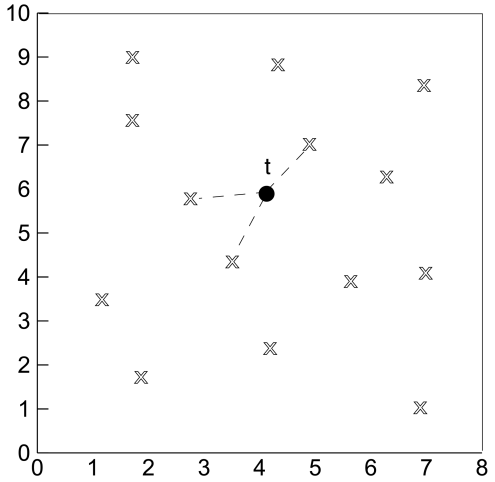
- kNN
- C4.5
- Naïve-Bayes
- Adaboost
- Random Rorests

k-Nearest neighbors (kNN)

- Clasificador basado en instancias: aprendizaje por analogía
- Paradigma perezoso de aprendizaje: el trabajo se retrasa todo lo posible
- No se construye ningún modelo, el modelo es el conjunto de entrenamiento
- Se trabaja cuando llega un nuevo caso a clasificar:
  - Se buscan los casos más parecidos y la clasificación se construye en función de la clase a la que dichos casos pertenecen

k-NN ►1-Nearest Neighbor (1NN)

- Si tenemos *m* instancias $\{e_1, \ldots, e_m\}$ para clasificar un nuevo ejemplo *e'* se hará lo siguiente:
  1. c_{min} = clase (e_1)
  2. d_{min} = d (e_1, e')
  3. Para *i*=2 hasta *m* hacer
     $d=d(e_i,e')$
     Si $(d<d_{min})$
     Entonces c_{min} = clase (e_i), $d_{min} = d$
  4. Devolver c_{min} como clasificación de *e'*



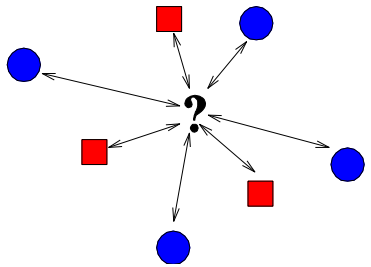
- d(·, ·) es una función de distancia

k-NN ►Extensión a los k vecinos más próximos

Dado el ejemplo e a clasificar

1. Seleccionar los k ejemplos más cercanos
2. Devolver la clase que más se repite en el conjunto {clase(e_1), …, clase (e_k)}

Por ejemplo, si k=7 el siguiente caso (?) se clasificaría como ●



- Se podría tratar de forma diferente a los k-vecinos, p.e., dependiendo de la distancia al objeto a clasificar:
  - Voto por la mayoría → Clasificación ●
  - Voto con pesos en función de la distancia → Clasificación □

k-NN ►k-Nearest Neighbors (kNN)

Algorithm 8.1 Basic *k*NN Algorithm

Input : D, the set of training objects, the test object, $\mathbf{z}$, which is a vector of attribute values, and L, the set of classes used to label the objects

Output : $c_z \in L$, the class of z

foreach object $\mathbf{y} \in D$ **do**

| Compute $d(\mathbf{z}, \mathbf{y})$, the distance between $\mathbf{z}$ and $\mathbf{y}$;

end

Select $N \subseteq D$, the set (neighborhood) of k closest training objects for z;

$c_z = \underset{v \in L}{\operatorname{argmax}} \sum_{y \in N} I(v = class(c_y))$;

where $I(\cdot)$ is an indicator function that returns the value 1 if its argument is true and 0 otherwise.

k-NN ►Funciones de distancia

Variables categóricas: Distancia de Hamming

$$d_h(a,b) = \begin{cases} 0, & si\ a = b \\ 1, & si\ a \neq b \end{cases}$$

Variables numéricas: se suelen normalizar al intervalo [0,1]

- Euclídea $d_e(e_1,e_2) = \sqrt{\sum_{i=1}^{n}(e_1^i - e_2^i)}$
- Manhattan $d_m(e_1,e_2) = \sum_{i=1}^{n}\left|e_1^i - e_2^i\right|$
- Minkowski $d_m^k(e_1,e_2) = \left(\sum_{i=1}^{n}\left|e_1^i - e_2^i\right|^k\right)^{1/k}$

$d_m^1 = d_m$ y $d_m^2 = d_e$

k-NN ▶ Funciones de distancia

- Por tanto, la distancia entre dos instancias e_1 y e_2, utilizando p.e. d_e para las variables numéricas sería

$$d_e(e_1, e_2) = \sqrt{\sum_i (e_1^i - e_2^i)^2 + \sum_j d_h(e_1^j, e_2^j)}$$

- Tratamiento de **valores desconocidos**: si e_{j1}=? y e_{j2}=? Entonces la distancia asociada a la j-ésima componente es la máxima (1)

- Se pueden ponderar las variables:

$$d_e(e_1, e_2) = \sqrt{\sum_i w_i \cdot (e_1^i - e_2^i)^2 + \sum_j w_j \cdot d_h(e_1^j, e_2^j)}$$

k-NN ▶ Consideraciones finales

- Robusto frente al ruido con valores de k moderados (k>1)
- Eficaz: utiliza varias funciones lineales locales para aproximar la función objetivo
- Válido para clasificación y regresión (devolviendo la media o la media ponderada por la distancia)
- Ineficiente en memoria: hay que almacenar toda la BD
- La distancia entre vecinos puede estar dominada por variables irrelevantes
- Complejidad $O(dn^2)$, siendo $O(d)$ la complejidad de la distancia utilizada
  - Para reducir esta complejidad: uso de prototipos

Clasificación con árboles de decisión

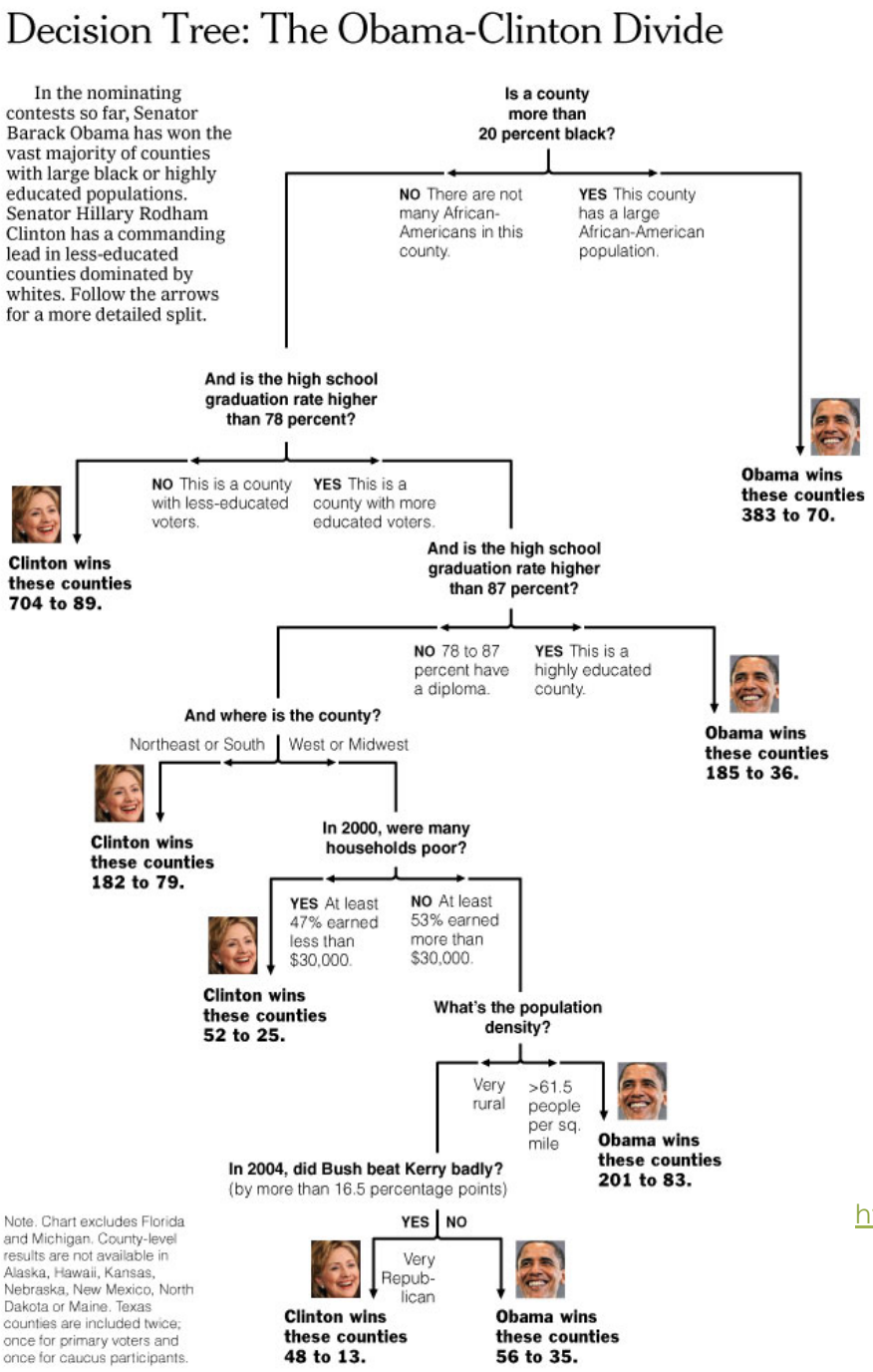
- ¿Qué es un árbol de decisión?
- Construcción de árboles de decisión
- Criterios de selección de variables
- Particionamiento del espacio con árbol de decisión
- Algoritmo C4.5

Clasificación con árboles ► ¿Qué es un árbol de decisión?

- Un árbol de decisión es un clasificador que en función de un conjunto de atributos permite determinar a que clase pertenece el caso objeto de estudio

- La estructura de un árbol de decisión es:
  - Cada hoja es una categoría (clase) de la variable objeto de la clasificación.
  - Cada nodo es un nodo de decisión que especifica una prueba simple a realizar.
  - Los descendientes de cada nodo son los posibles resultados de la prueba del nodo.

¿Qué es un árbol de decisión?



http://graphics8.nytimes.com/images/2008/04/16/us/0416-nat-subOBAMA.jpg

Proceso de **generación de un árbol de decisión**

1. Construcción del árbol
   - Al inicio todos los ejemplos de entrenamiento están en el nodo raíz
   - Dividir recursivamente los ejemplos en base a los atributos seleccionados
2. Poda del árbol
   - Identificar y quitar ramas que describen ruido o datos anómalos

Uso del árbol de decisión: Clasificar un ejemplo desconocido

- Comprobar los valores de los atributos del ejemplo contra el árbol de decisión

Algoritmo básico

Se construye el árbol mediante la técnica divide y vencerás

- Todos los ejemplos están en el nodo raíz
- Encontrar la variable que mejor separe los ejemplos
- Dividir los ejemplos basándose en el/los atributos seleccionados
- Continuar recursivamente el proceso hasta que los grupos de ejemplos sean muy pequeños o suficientemente "puros"

Los atributos de test se seleccionan en base a una medida heurística o estadística

Condiciones para terminar

- Todos los ejemplos para un nodo dado pertenecen a la misma clase
- No quedan más atributos para seguir particionando: voto de la mayoría
- NO quedan ejemplos

Clasificación con árboles ▶ Construcción de árboles de decisión

Problema de asignación de crédito

crédito	ingresos	propietario	Gastos-mensuales
N	Bajos	N	Altos
N	Bajos	S	Altos
N	Medios	S	Altos
N	Medios	N	Altos
N	Altos	N	Altos
S	Altos	S	Altos
N	Bajos	N	Bajos
N	Medios	N	Bajos
N	Altos	N	Bajos
s	Medios	S	Bajos

1. Se llama al algoritmo sobre el nodo raíz

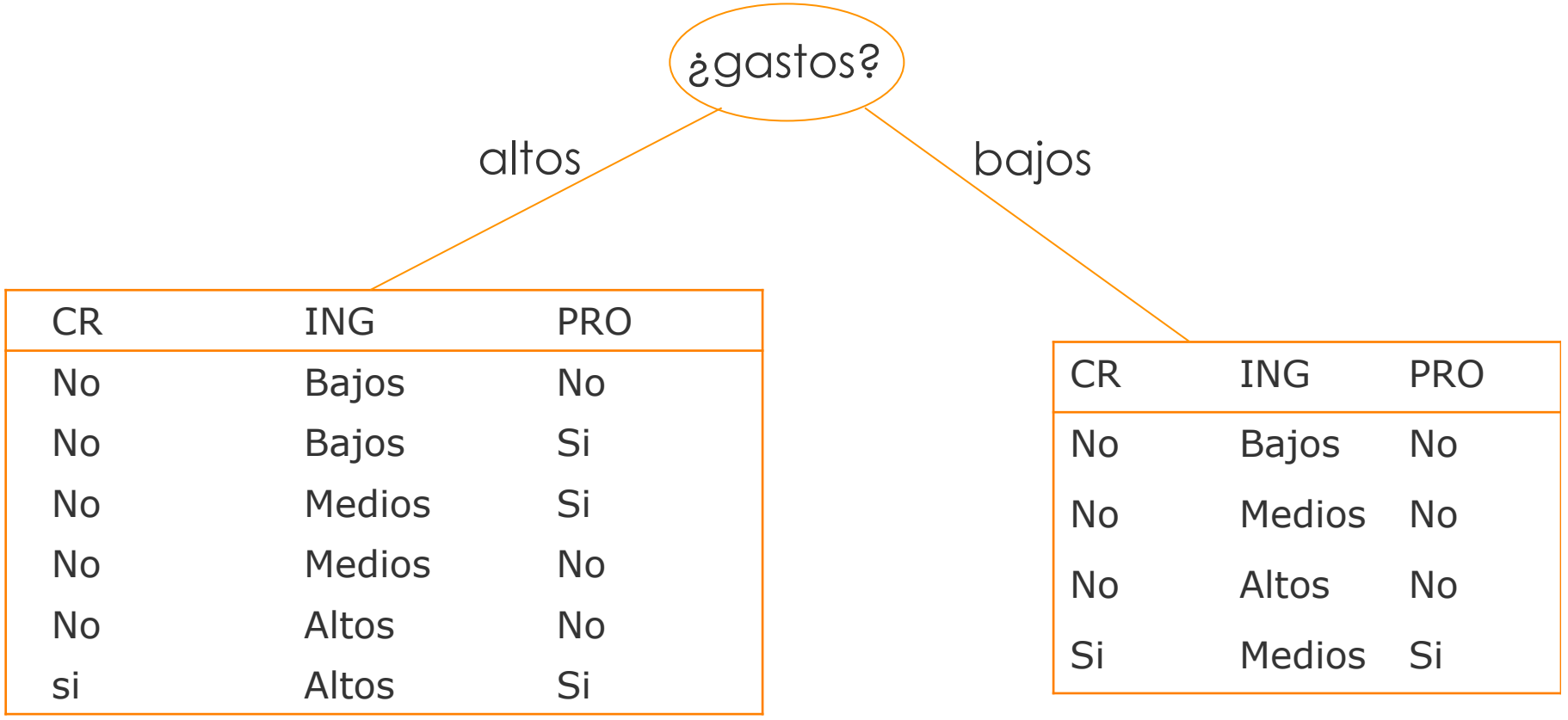
crédito	ingresos	propietario	Gastos-mensuales
N	Bajos	N	Altos
N	Bajos	S	Altos
N	Medios	S	Altos
N	Medios	N	Altos
N	Altos	N	Altos
S	Altos	S	Altos
N	Bajos	N	Bajos
N	Medios	N	Bajos
N	Altos	N	Bajos
s	Medios	S	Bajos

2. Seleccionamos gastos como variable test

CR	ING	PRO	gastos
No	Bajos	No	Altos
No	Bajos	Si	Altos
No	Medios	Si	Altos
No	Medios	No	Altos
No	Altos	No	Altos
si	Altos	Si	Altos

CR	ING	PRO	gastos
No	Bajos	No	Bajos
No	Medios	No	Bajos
No	Altos	No	Bajos
Si	Medios	Si	Bajos

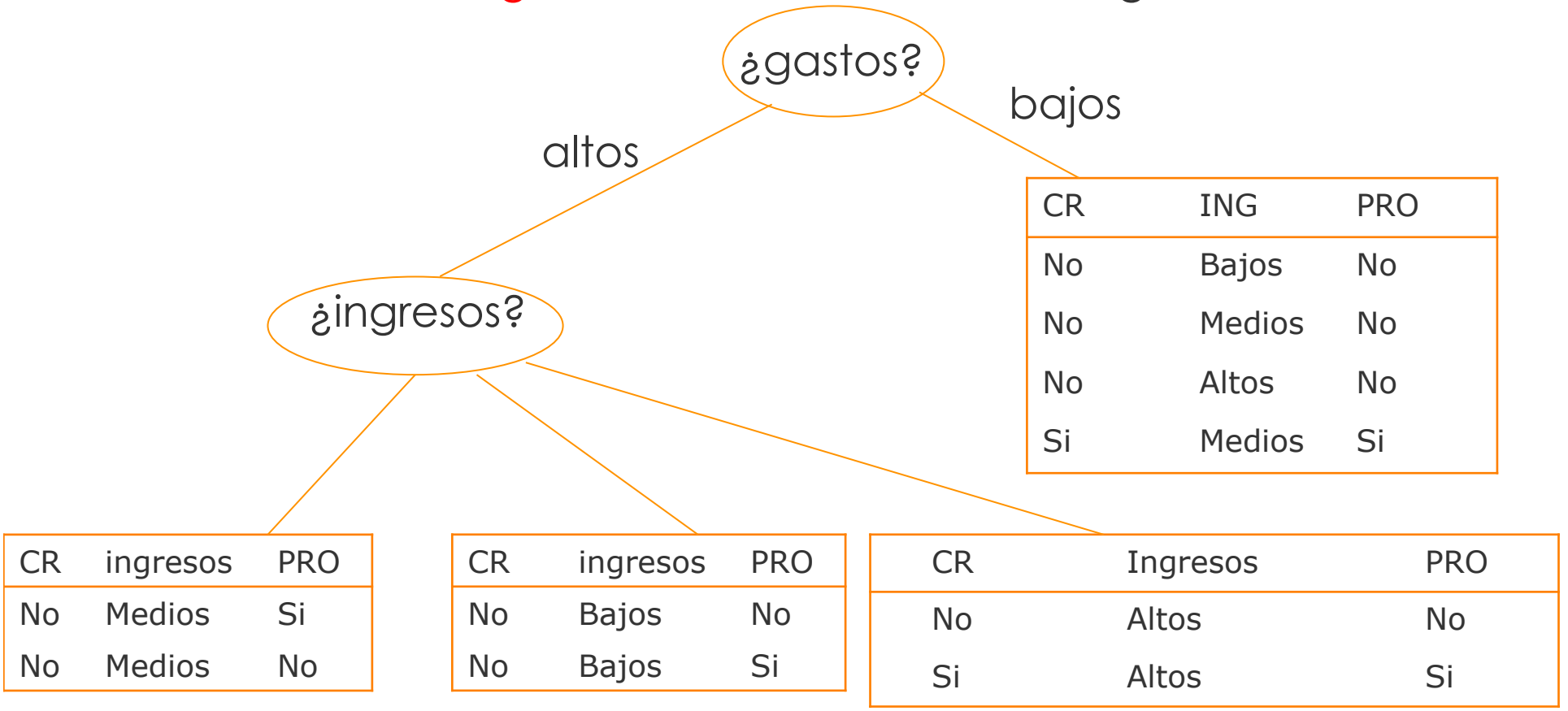
3. Preparamos los nodos para las llamadas recursivas



CR	ING	PRO
No	Bajos	No
No	Bajos	Si
No	Medios	Si
No	Medios	No
No	Altos	No
si	Altos	Si

CR	ING	PRO
No	Bajos	No
No	Medios	No
No	Altos	No
Si	Medios	Si

4. Seleccionamos ingresos como variable test en gastos=altos



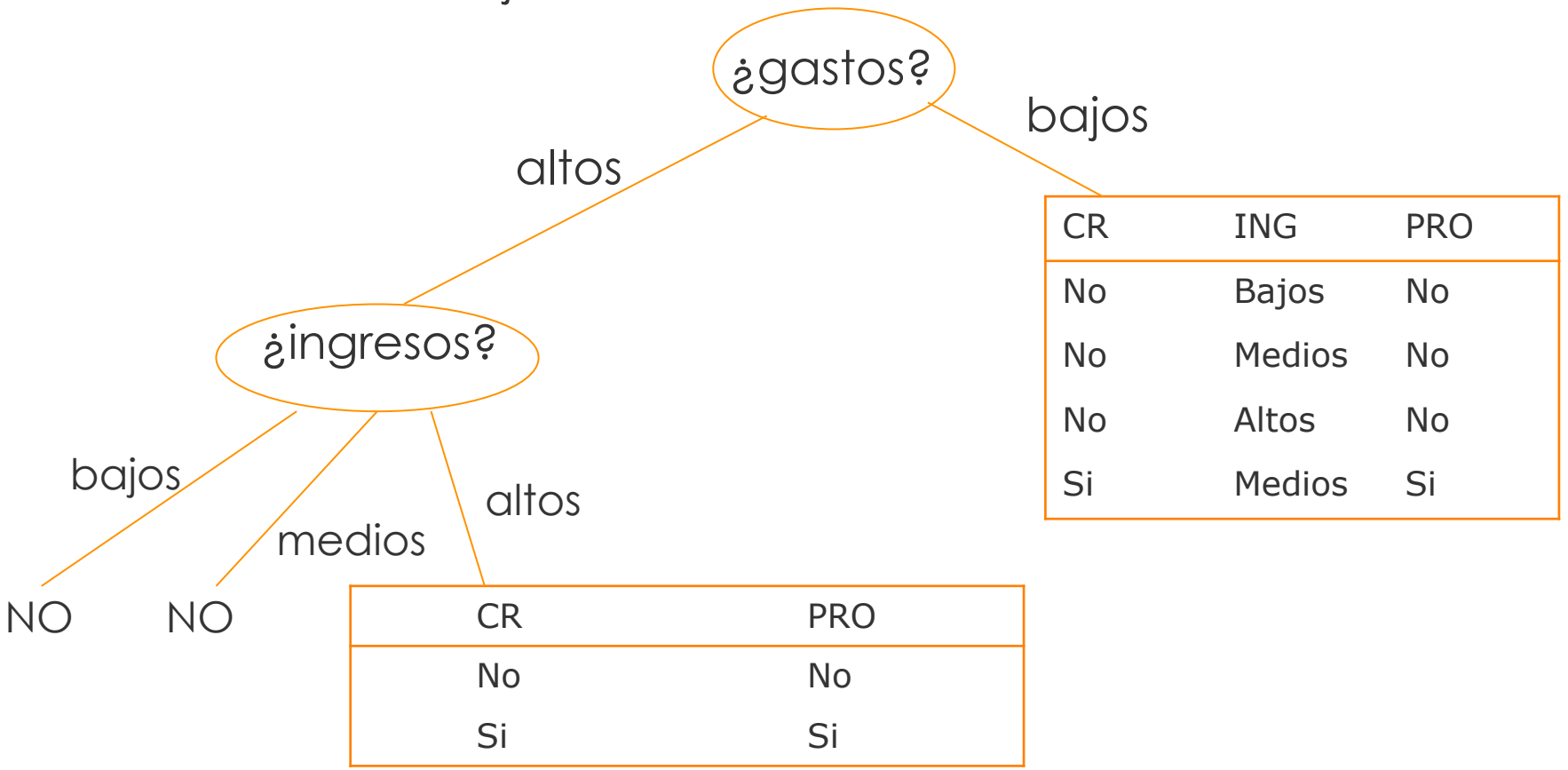
CR	ING	PRO
No	Bajos	No
No	Medios	No
No	Altos	No
Si	Medios	Si

CR	ingresos	PRO
No	Medios	Si
No	Medios	No

CR	ingresos	PRO
No	Bajos	No
No	Bajos	Si

CR	Ingresos	PRO
No	Altos	No
Si	Altos	Si

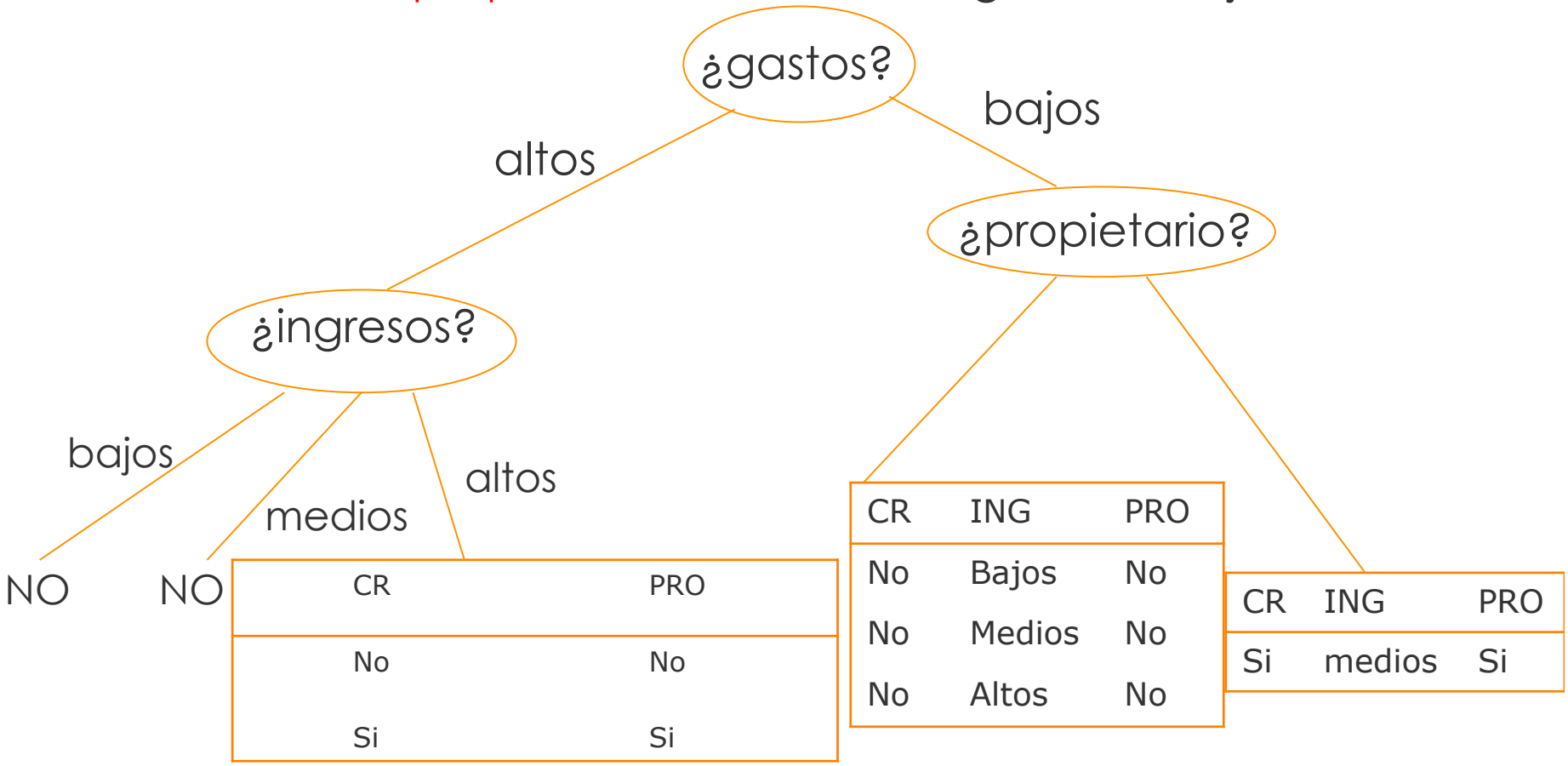
5. Creamos nodos hoja



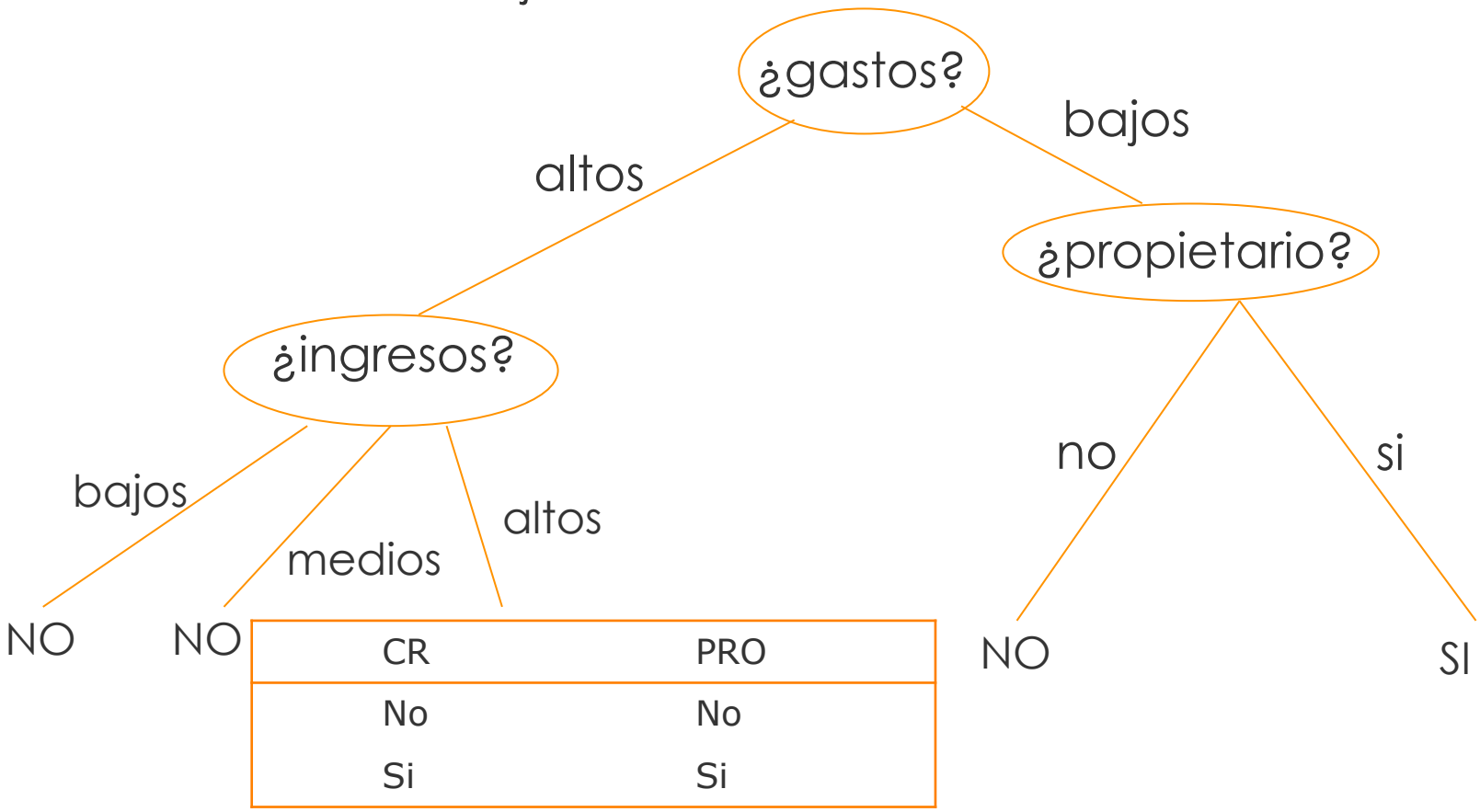
CR	ING	PRO
No	Bajos	No
No	Medios	No
No	Altos	No
Si	Medios	Si

CR	PRO
No	No
Si	Si

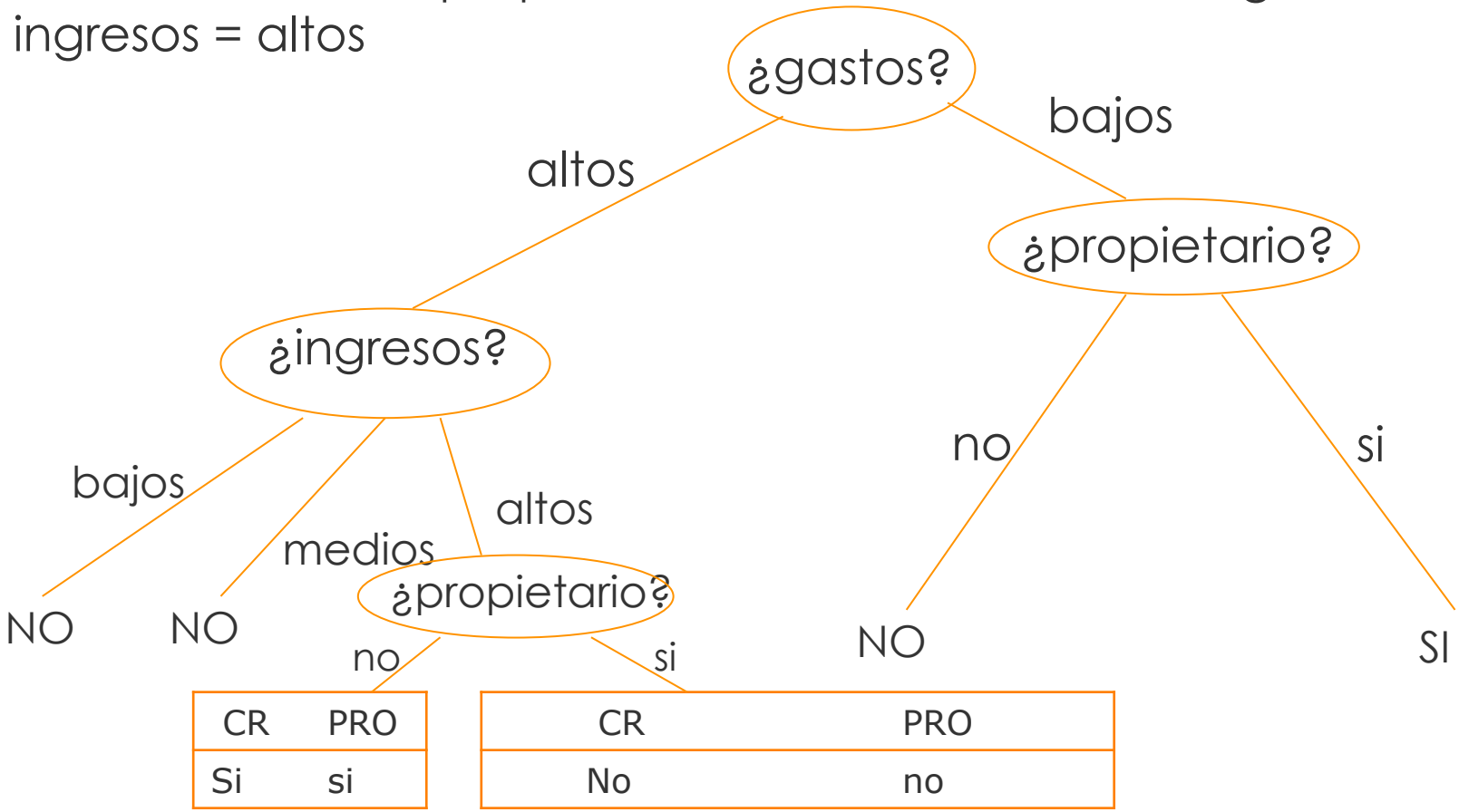
6. Seleccionamos propietario como test en gastos = bajos



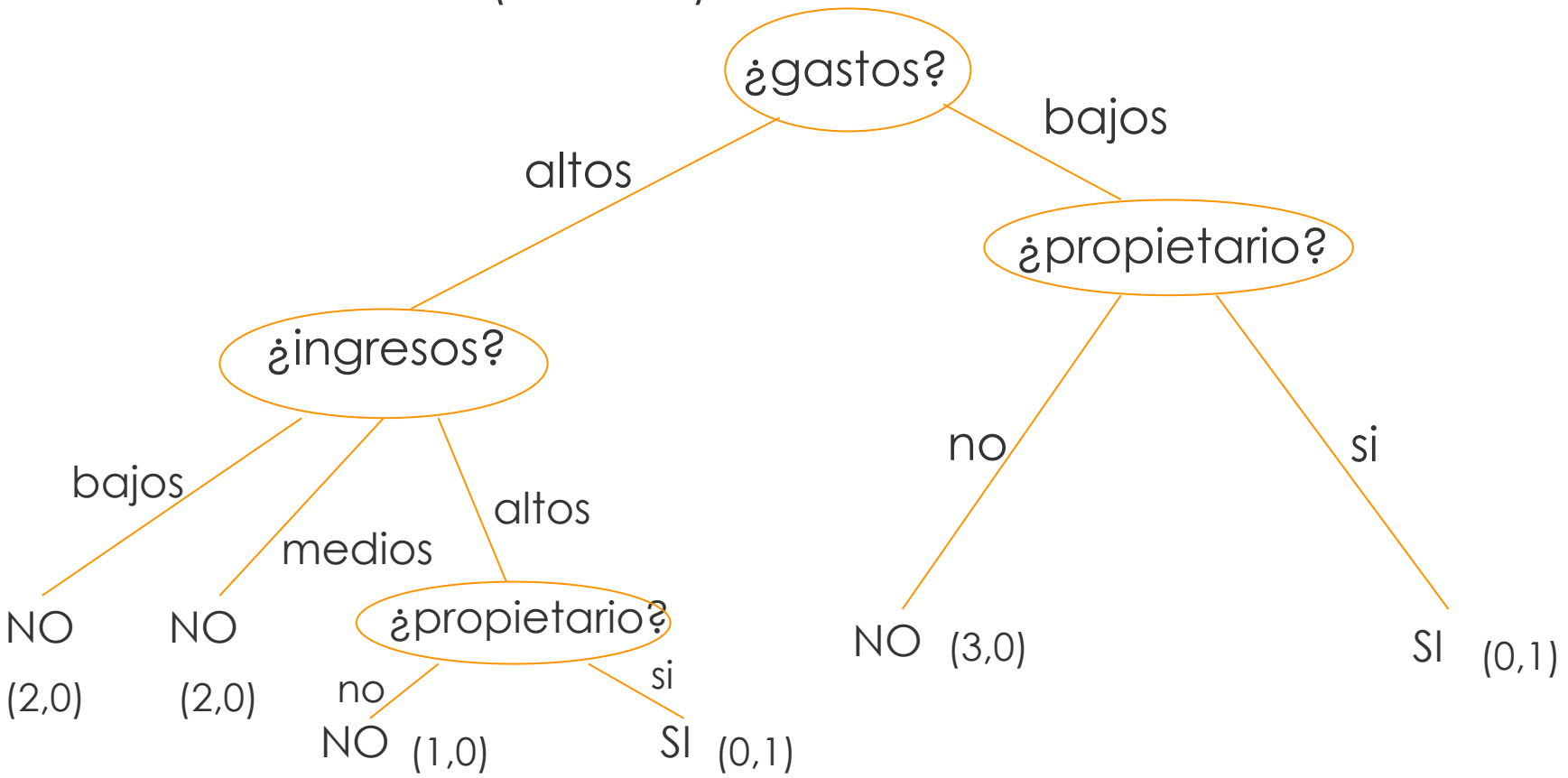
7. Creamos nodos hoja



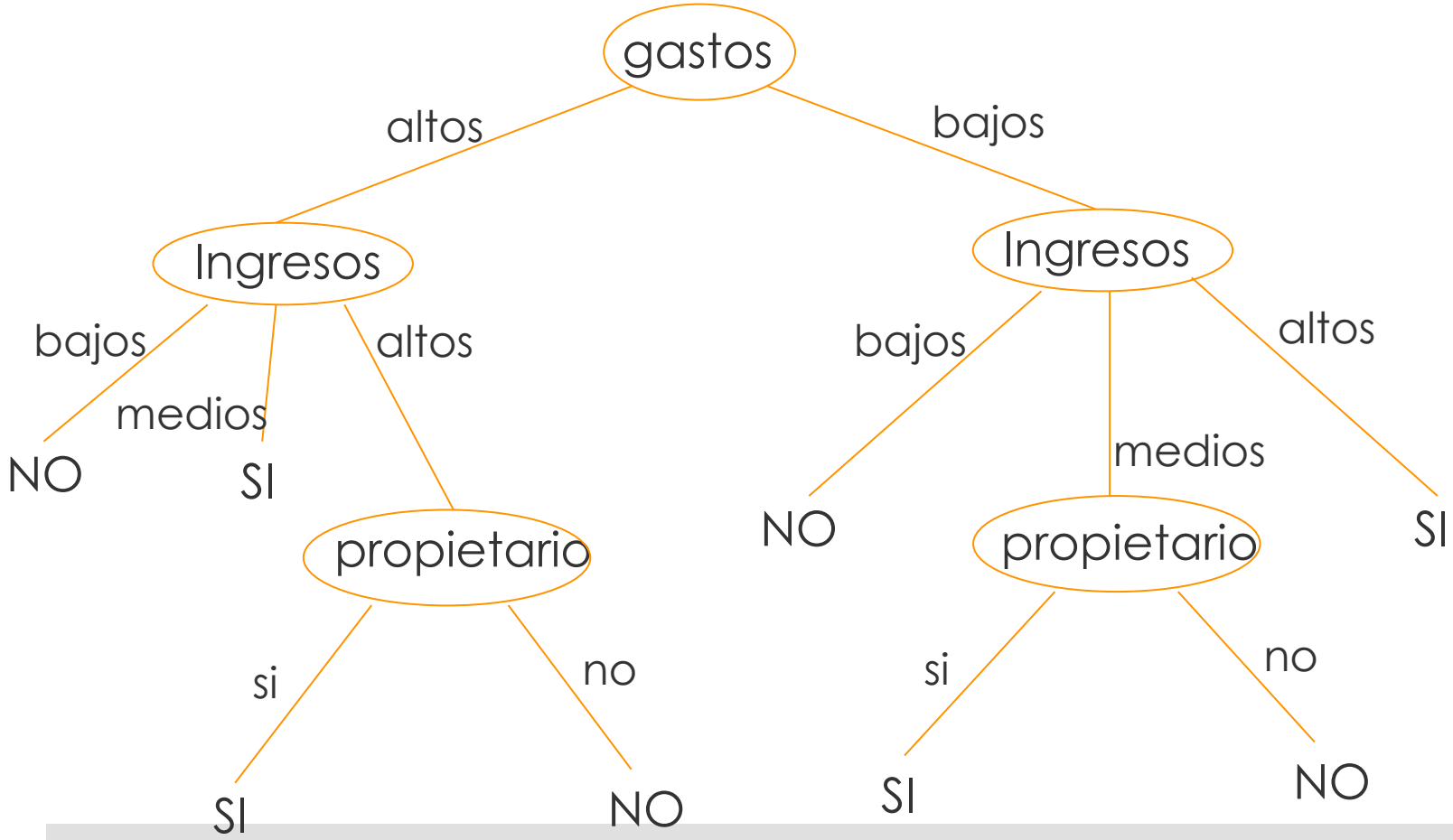
8. Seleccionamos propietario como variable test en gastos = altos e ingresos = altos



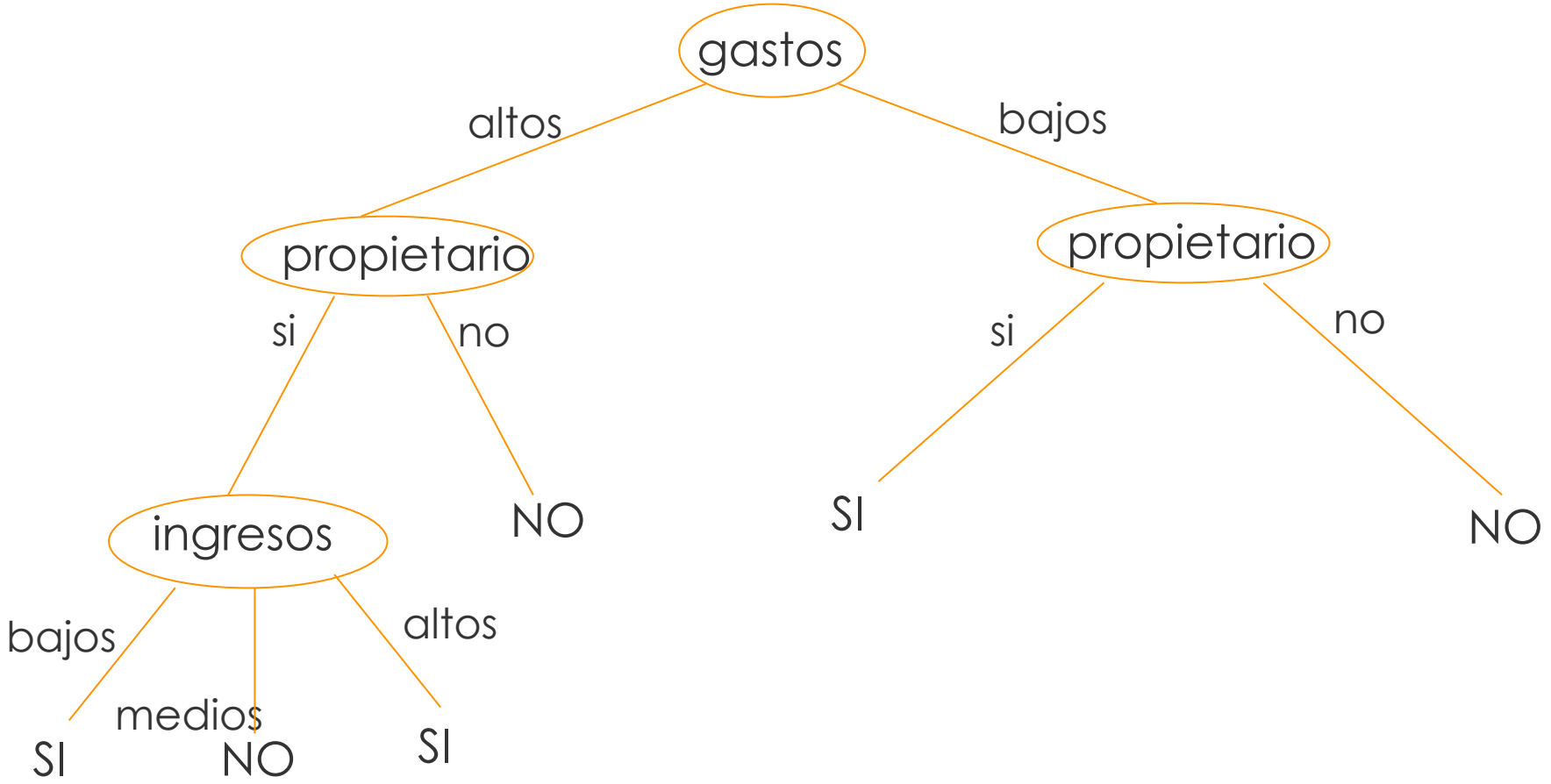
9. Árbol de decisión (#no, #si)



- Otro árbol de decisión para crédito



- Otro árbol de decisión para crédito



Clasificación con árboles ▶ Criterios de selección de variables

- 1er. árbol: 6 reglas (2.33 premisas por regla)
- 2do. árbol: 8 reglas (2.5 premisas por regla)
- 3er. árbol: 6 reglas (2.5 premisas por regla)

- Dependiendo del orden en el que se van tomando los atributos obtenemos clasificadores de distinta complejidad
- Lo ideal sería tomar en todo momento el atributo que mejor clasifica

¿Cómo decidir qué atributo es el mejor?

Clasificación con árboles ► Criterios de selección de variables

Criterios para seleccionar la variable X* de test:

- InfoGain: Ganancia de información (ID3)

$$X^* = \max_X (H(C) - H(C|X))$$

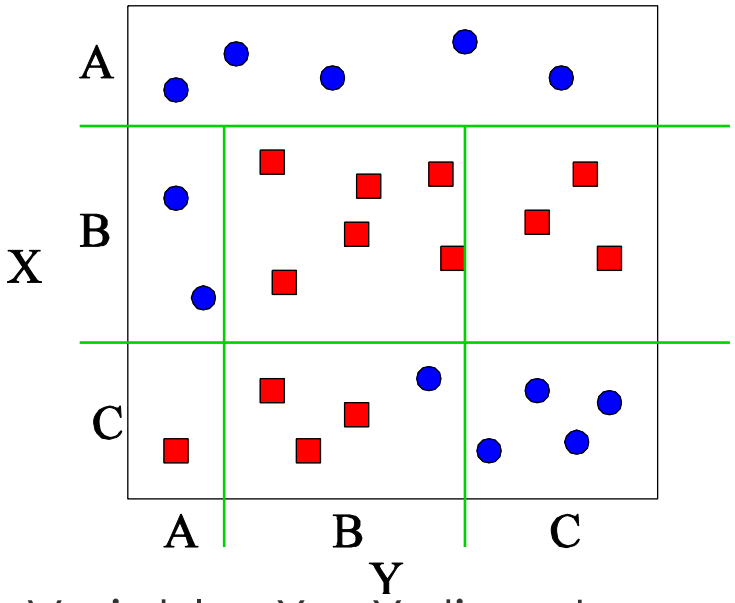
- GainRatio: Ganancia de información modificada (C4.5)

$$X^* = \max_X \frac{H(C) - H(C|X)}{H(X)}$$

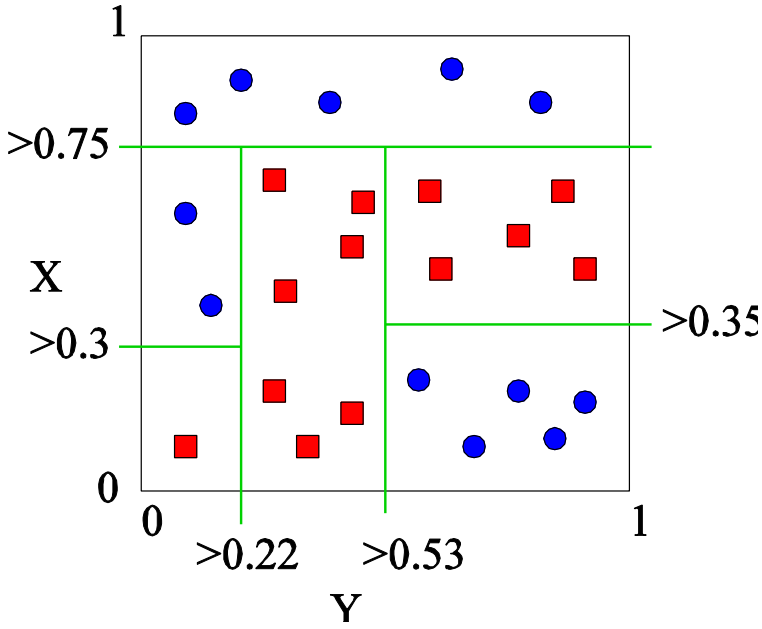
- GINI(CART) $X^* = \max_X (G(C) - G(C|X)) \quad con \quad G = 1 - \sum_{i=1}^{n} p_i^2$

- Los árboles particionan el espacio de forma exhaustiva

  Ejemplos para variables discretas y continuas



Variables X e Y discretas



Variables X e Y continuas

- J.H. Friedman. *A Recursive Partitioning Decision Rule for Nonparametric Classification*. IEEE Transactions on Computers, 26(4):404-408, 1977.
- L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen. *Classification and Regression Trees*. Chapman & Hall/ CRC, 1984. **(CART)**
- J.R. Quinlan. *Induction of Decision Trees*. Machine Learning, 1(1): 81-106, 1986. **(ID3)**
- J.R. Quinlan. C4.5: *Programs for Machine Learning*. Mrogan Kaufmann, 1993. **(C4.5)**

Clasificación con árboles ► El algoritmo C4.5

J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993

- Conjunto de algoritmos para problemas de clasificación
  - C4.5
  - C4.5-sin poda
  - C4.5-reglas
- Aprendizaje supervisado
- Desciende de ID3
- Actualmente tiene una versión comercial superior:

  See5/c5.0 systems de *Rulequest Research, Inc*.

Clasificación con árboles ► El algoritmo C4.5

Algorithm 1.1 C4.5(D)

Input: an attribute-valued dataset D

1: Tree = {}
2: **if** D is "pure" OR other stopping criteria met **then**
3: terminate
4: **end if**
5: **for all** attribute $a \in D$ **do**
6: Compute information-theoretic criteria if we split on a
7: **end for**
8: a_{best} = Best attribute according to above computed criteria
9: Tree = Create a decision node that tests a_{best} in the root
10: D_v = Induced sub-datasets from D based on a_{best}
11: **for all** D_v **do**
12: Tree_v = C4.5(D_v)
13: Attach Tree_v to the corresponding branch of Tree
14: **end for**
15: **return** Tree

Clasificación con árboles ► El algoritmo C4.5

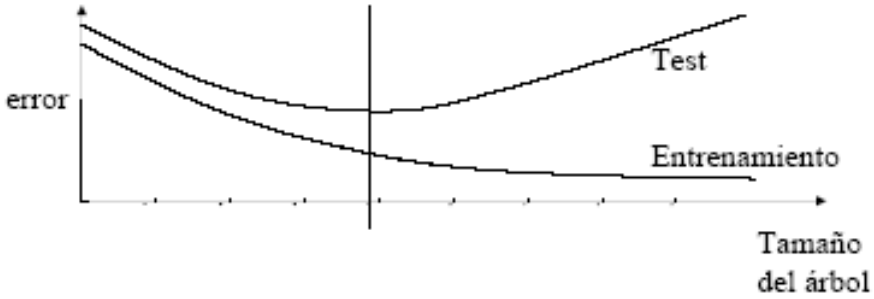
- **¿Qué tipos de tests se utilizan en los nodos no hoja?**
  - Var. booleana: se generan dos ramas
  - Var. categórica: se generan varias ramas, pero si un grupo de valores predice una única clase se agrupan en una única rama
  - Var. numérica: se generan dos ramas $\{\leq \Theta, \geq \Theta\}$
- **¿Qué variables test se eligen?** Basados en la ganancia de información y el ratio de ganancia
- **¿Qué umbrales se eligen?** El umbral se obtiene ordenando los ejemplos en base a esa variable y eligiendo el corte que maximiza el criterio elegido
- **¿Cuándo finaliza la construcción del árbol?**
  - no quedan variables o ejemplos
  - todos los ejemplos pertenecen a la misma clase o
  - el número de ejemplos está por debajo de un umbral.
- **¿Con qué clase se etiquetan las hojas?** Con la mayoritaria para los ejemplos de la rama

Clasificación con árboles ► El algoritmo C4.5

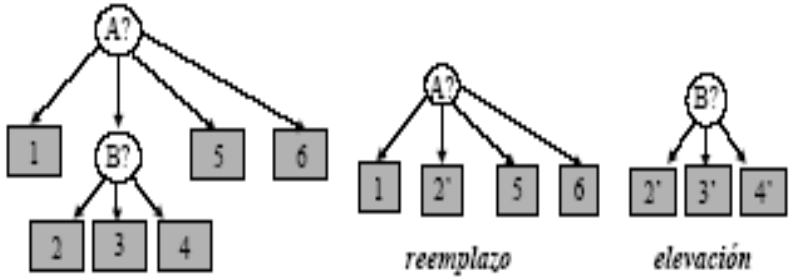
- **Datos perdidos**:
  - Cuando se construye el árbol, los datos perdidos se ignoran
  - Para clasificar un ejemplo con valor perdido, éste se predice en base a lo que se sabe sobre los valores del atributo para otros registros
- Propone soluciones para el **sobreaprendizaje**. Posibilidades
  - pre-poda: se decide cuándo dejar de subdividir el árbol
  - post-poda: se construye el árbol y después se poda

Clasificación con árboles ► El algoritmo C4.5

- Pre-poda:
  - no se divide un nodo si se tiene pocc confianza en él (no es significativa la diferencia de clases), o
  - se valida con un conjunto de test independiente y se para cuando la curva del conjunto de test empieza a subir



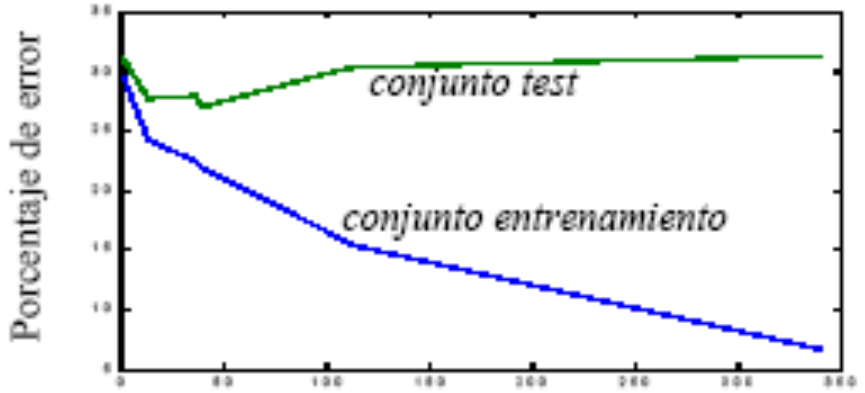
- Post- poda
  - Reemplazamiento de subárboles: Se reemplaza un subárbol por una hoja si al hacerlo el error es similar al original
  - Elevación de subárbol: Reemplaza un subárbol por su subárbol más utilizado



Clasificación con árboles ▶ El algoritmo C4.5

Efecto del tamaño del árbol

Tamaño del árbol	Conjunto de entrenamiento		Conjunto de test	
	instancias incorrectas	porcentaje de error	instancias incorrectas	porcentaje de error
1	207	29.57 %	93	31%
13	170	24.29 %	83	27.67 %
36	157	22.43%	84	28%
39	154	22%	81	27%
95	119	17%	89	29.67%
113	108	15.43%	91	30.3%
340	47	6.71%	94	31.3%



Naïve-Bayes

Los métodos probabilísticos/bayesianos representan la incertidumbre asociada a los procesos de forma natural

Ejemplo: Supongamos que consultamos un sistema de recomendaciones (SR) para invertir en bolsa sobre P1 y P2

- Si el modelo utilizado por el SR no trata la incertidumbre (p.e. un árbol de decisión), podríamos obtener:

  (P1, invertir) (P2, invertir)

- Si el modelo utilizado por el SR trata la incertidumbre (p.e. una red bayesiana) podríamos obtener:

(P1, invertir, prob=0.9) (P2, invertir, prob=0.52)

(P1, no invertir, prob=0.1) (P2, no invertir, prob=0.48)

Naïve-Bayes ► Teorema de Bayes e hipótesis MAP

- **Teorema de Bayes** en un problema de clasificación con *n* variables

$$P(C|A_1,\ldots,A_n) = \frac{P(A_1,\ldots,A_n|C)P(C)}{P(A_1,\ldots,A_n)}$$

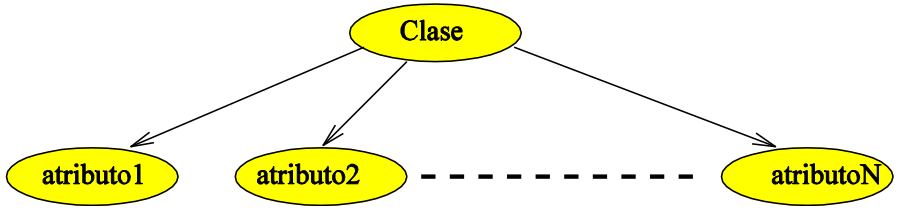
- **Hipótesis MAP** (máxima a posteriori): Si queremos clasificar un nuevo caso $(a_1,\ldots a_n)$ y la variable clase C tiene *k* posibles categorías $\Omega_C=\{c_1,\ldots,c_k\}$, se identifica la más probable y se devuelve como clasificación

$$\begin{aligned} c_{MAP} &= \arg\max_{c\in\Omega_C} P(c|a_1,\ldots,a_n) \\ &= \arg\max_{c\in\Omega_C} \frac{P(a_1,\ldots,a_n|c)P(c)}{P(a_1,\ldots,a_n)} \\ &= \arg\max_{c\in\Omega_C} P(a_1,\ldots,a_n|c)P(c) \end{aligned}$$

- Problema: Trabajar con la distribución conjunta

Naïve-Bayes ▶ Clasificador Naïve-Bayes

- El modelo de red bayesiana orientada a clasificación más simple
- Supone que todos los atributos son independientes conocida la variable clase



- Hipótesis MAP en un Naïve BAyes (NB):

$$c_{MAP} = \arg\max_{c \in \Omega_C} P(c|a_1,\ldots,a_n) = \arg\max_{c \in \Omega_C} P(c)\prod_{i=1}^{n} P(a_i|c)$$

- A pesar de la suposición poco realista realizada en el NB sus resultados son competitivos con la mayoría de los clasificadores

¿Cómo se estiman estas probabilidades?

- **Variables discretas**
  - $P(x|c_i)$ se estima como la frecuencia relativa de ejemplos que teniendo un determinado valor de x pertenecen a la clase c_i
  - Estimación por máxima verisimilitud (EMV)

$$p(x_i|x_j) = \frac{n(x_i, x_j)}{n(x_j)}$$

    - $n(x_i)$: nº de veces que aparece $X_i=x_i$ en la BD
    - $n(x_i,x_j)$: nº de veces que aparece el par $(X_i=x_i, X_j=x_j)$ en la BD

  - Suavizando por la corrección de Laplace:

$$p(x_i|x_j) = \frac{n(x_i, x_j)+1}{n(x_j)+|\Omega_{X_i}|}$$

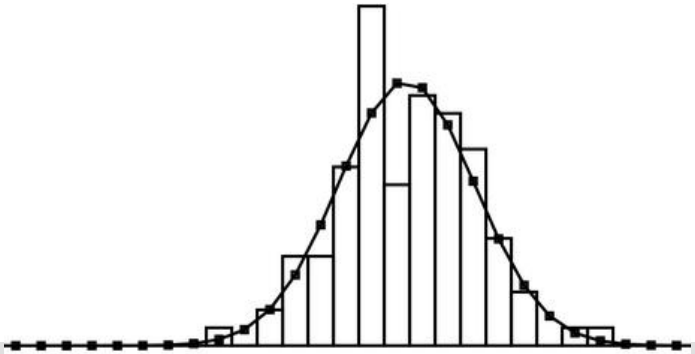
Naïve-Bayes ▶ Naïve-Bayes

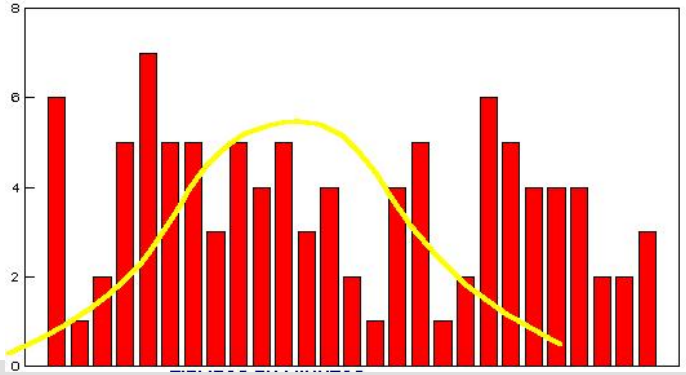
- **Variables numéricas**: P(x | ci) se estima mediante una función de densidad gaussiana (se asume que los valores numéricos siguen una distribución normal)

$$P(x|c_i) = N(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{(x-\mu)^2}{2\sigma^2}}$$

para cada categoría de la variable clase se estima una distribución normal (de media μ y desviación estándar σ)

En unos casos la aproximación realizada será mejor que en otros





Naïve-Bayes ▶ Naïve-Bayes

- **Ventajas:**
  - Es fácil de implementar
  - Obtiene buenos resultados en gran parte de los casos
- **Desventajas:**
  - Asumir que las variables tienen independencia condicional respecto a la clase lleva a una falta de precisión
  - En la práctica, existen dependencias entre las variables. P.e.: en datos hospitalarios:
    - Perfil: edad, historia familiar, etc.
    - Síntomas: fiebre, tos, etc.
    - Enfermedad: cáncer de pulmón, diabetes, etc.

    Con un clasificador Naïve Bayes no se pueden modelar estas dependencias.
    - Solución: Redes de creencia bayesianas

Ensemble learning

Aprendizaje tradicional: Generar un predictor a partir de los ejemplos

Ensemble learning: Generar un conjunto de predictores base y combinarlos

- Clasificación: combinación de la salida de cada clasificador
- Los clasificadores pueden estar basados en diferentes técnicas (árboles, reglas, instancias, etc.)
  - En general se aplican sobre una única técnica
- Pueden estar basados en diferentes conjuntos de ejemplos y/o variables

- **Bagging**: cada clasificador se induce independientemente

- **Boosting**: cada clasificador tiene en cuenta los fallos del anterior
  - Uno de los más conocido es Adaboost

Y. Freund, R.E. Schapire. A decisión-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sicences, 55(1):11-9-139,1997.

Y. Freund, R.E. Schapier. A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence, 14(5): 771-780, 1999.

Ensemble learning ►Boosting►Idea básica

¿De qué forma?

- Obtener un predictor a partir de un subconjunto de los ejemplos
- Obtener otro predictor a partir de un segundo conjunto de ejemplos
- …
- Repetir este proceso un determinado número de veces

¿Cómo se eligen los ejemplos en cada etapa? Concentrándonos en los ejemplos más difíciles

- Los ejemplos mal clasificados por predictores obtenidos antes

¿Cómo se combinan los predictores en uno? Considerando el voto ponderado de cada uno ellos

Ensemble learning ►Boosting►Procedimiento general

Input: Instance distribution $\mathcal{D}$;
Base learning algorithm L;
Number of learning rounds T.

Process:

1. $\mathcal{D}_1 = \mathcal{D}$. % Initialize distribution
2. **for** $t = 1, \cdots, T$:
3. $h_t = L(\mathcal{D}_t)$; % Train a weak learner from distribution $\mathcal{D}_t$
4. $\epsilon_t = \Pr_{\boldsymbol{x}\sim\mathcal{D}_t, y}\, \boldsymbol{I}[h_t(\boldsymbol{x}) \neq y]$; % Measure the error of h_t
5. $\mathcal{D}_{t+1} = AdjustDistribution\ (\mathcal{D}_t, \epsilon_t)$
6. **end**

Output: $H(\boldsymbol{x}) = CombineOutputs(\{h_t(\boldsymbol{x})\})$

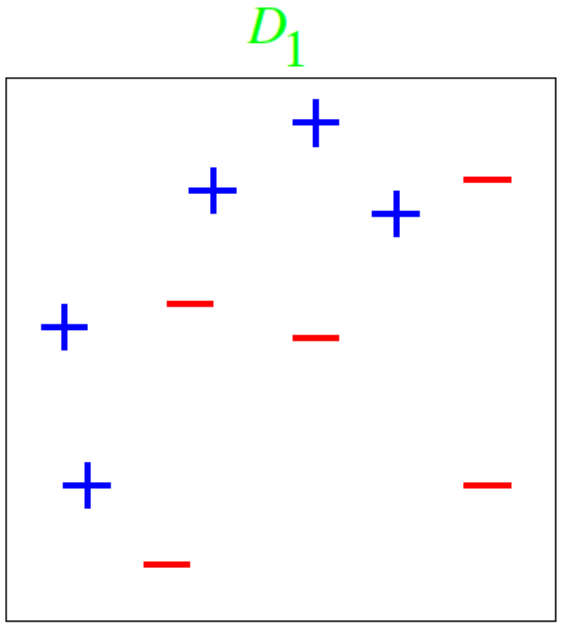
Ensemble learning ▶ Boosting ▶ **Adaboost**

Input: Data set $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
Base learning algorithm L;
Number of learning rounds T.

Process:

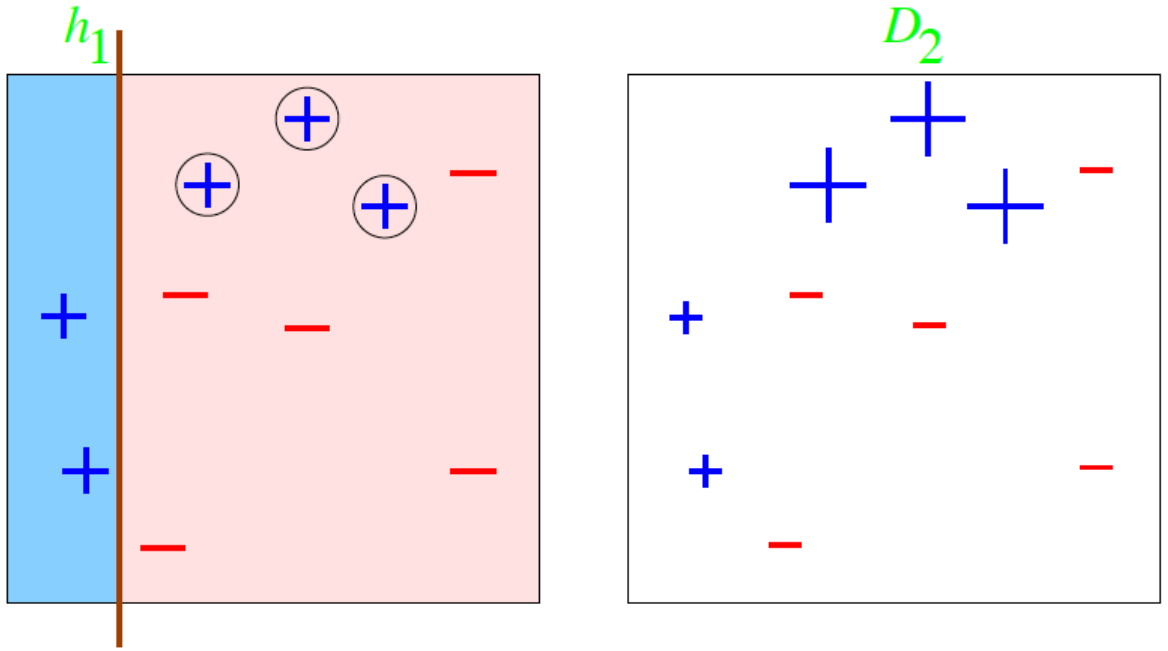
1. $\mathcal{D}_1(i) = 1/m$. % Initialize the weight distribution
2. **for** $t = 1, \cdots, T$:
3. $h_t = L(D, \mathcal{D}_t)$; % Train a learner h_t from D using distribution $\mathcal{D}_t$
4. $\epsilon_t = \Pr_{\boldsymbol{x} \sim \mathcal{D}_t, y} \boldsymbol{I}[h_t(\boldsymbol{x}) \neq y]$; % Measure the error of h_t
5. **if** $\epsilon_t > 0.5$ **then break**
6. $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$; % Determine the weight of h_t
7. $\mathcal{D}_{t+1}(i) = \frac{\mathcal{D}_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t) & \text{if } h_t(\boldsymbol{x}_i) = y_i \\ \exp(\alpha_t) & \text{if } h_t(\boldsymbol{x}_i) \neq y_i \end{cases}$
$= \frac{\mathcal{D}_t(i) \exp(-\alpha_t y_i h_t(\boldsymbol{x}_i))}{Z_t}$ % Update the distribution, where
% Z_t is a normalization factor which
% enables $\mathcal{D}_{t+1}$ to be distribution
8. **end**

Output: $H(\boldsymbol{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})\right)$



http://webee.technion.ac.il/people/rmeir/BoostingTutorial.pdf

Round 1

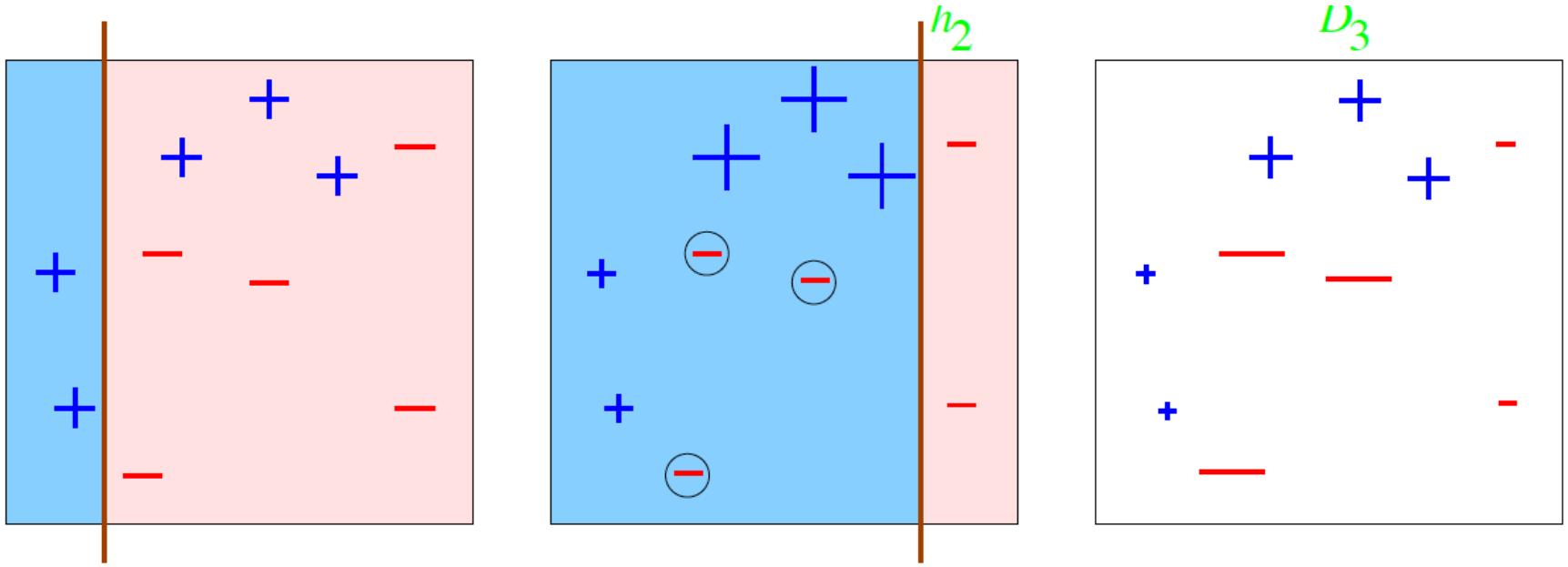


$\varepsilon_1 = 0.30$

$\alpha_1 = 0.42$

http://webee.technion.ac.il/people/rmeir/BoostingTutorial.pdf

Round 2

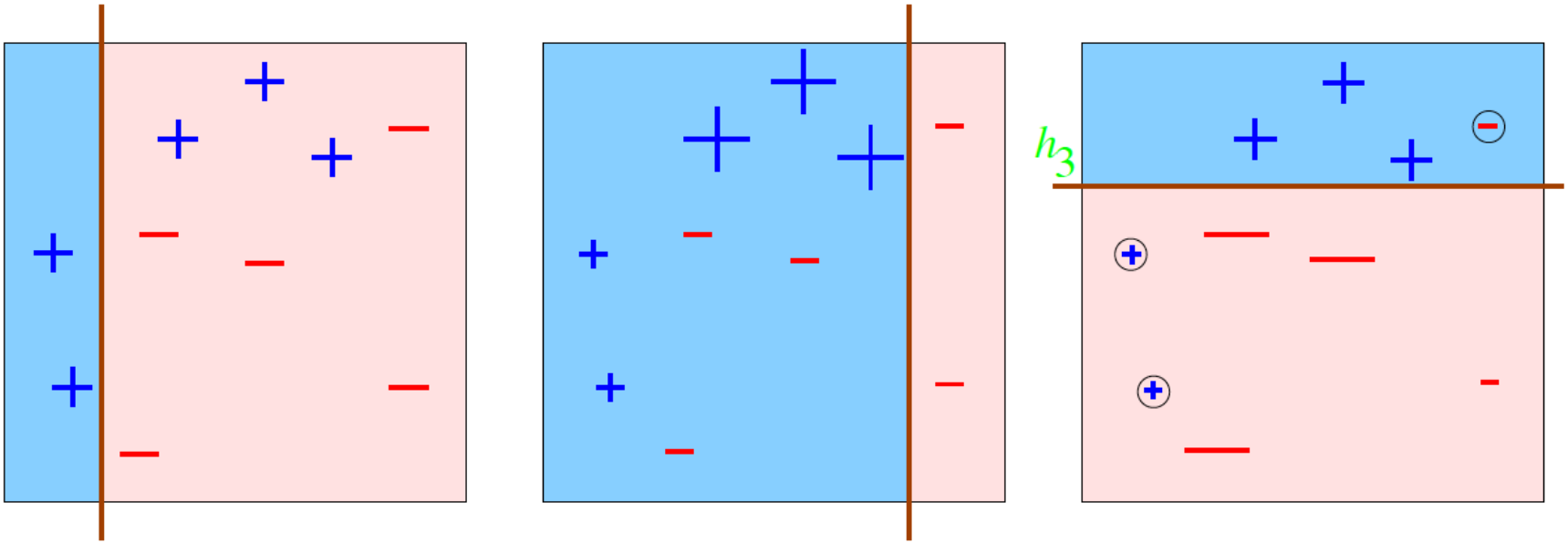


$\varepsilon_2 = 0.21$

$\alpha_2 = 0.65$

http://webee.technion.ac.il/people/rmeir/BoostingTutorial.pdf

Round 3

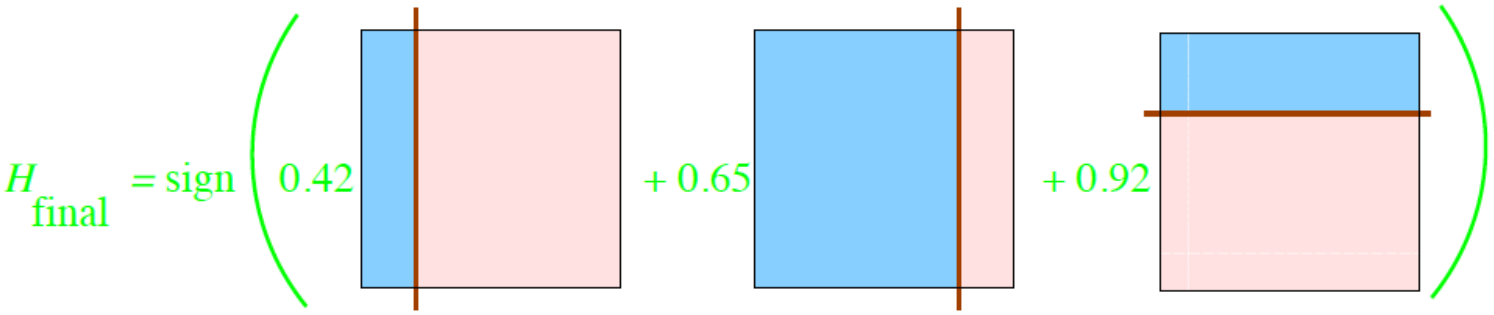


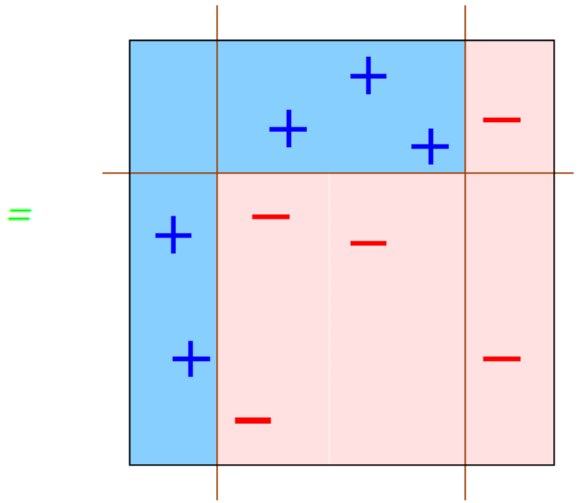
$\varepsilon_3 = 0.14$

$\alpha_3 = 0.92$

http://webee.technion.ac.il/people/rmeir/BoostingTutorial.pdf

Predictor final





Ensemble learning ►Boosting►**AdaBoost**

Ventajas:

- Rápido
- Sencillo y fácil de programar
- No tiene demasiados parámetros para ajustar
- Flexible (puede combinar cualquier algoritmo de aprendizaje)
- Efectivo
- Versátil: Tiene extensiones para clasificación multiclase y regresión

Limitaciones:

- El rendimiento depende de los datos y del clasificador base
  - Si los clasificadores base son muy complejos → overtfitting
  - Si son muy sencillos → poca precisión

Bibliografía adicional

- R. Meir, G. Rätsch. An Introduction to Boosting and Leveraging. In Advanced Lectures on Machine Learning (LNAI 2600) 2003.
- R.E. Schapire. The Boosting Approach to Machine Learning: An Overview. In MSRI Workshop on Nonlinear Estimation and Classification, 2002.

Ensemble learning ►Bagging

Bagging (Bootstrap Aggregating)

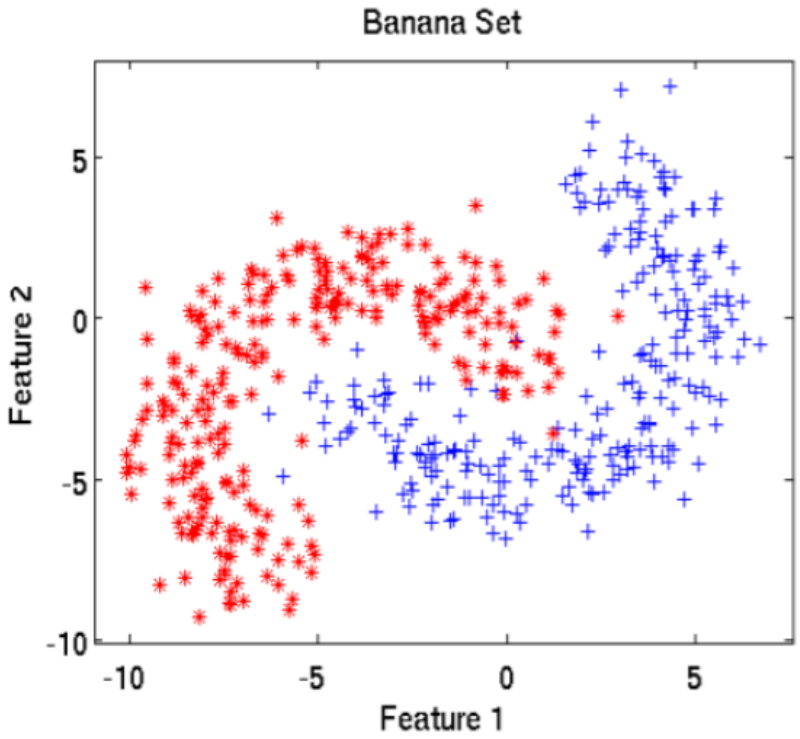
Dado un conjunto de datos y un tipo de algoritmos y modelos de aprendizaje (árboles, reglas, redes neuronales,…)

- Idea:
  - Obtener diferentes modelos en diferentes subconjuntos de datos
  - Predecir promediando las salidas de los diferentes modelos
- Objetivo:
  - Mejorar la precisión de cada modelo
  - La media de los errores en diferentes conjuntos da una estimación mejor de la habilidad predictiva de un método de aprendizaje

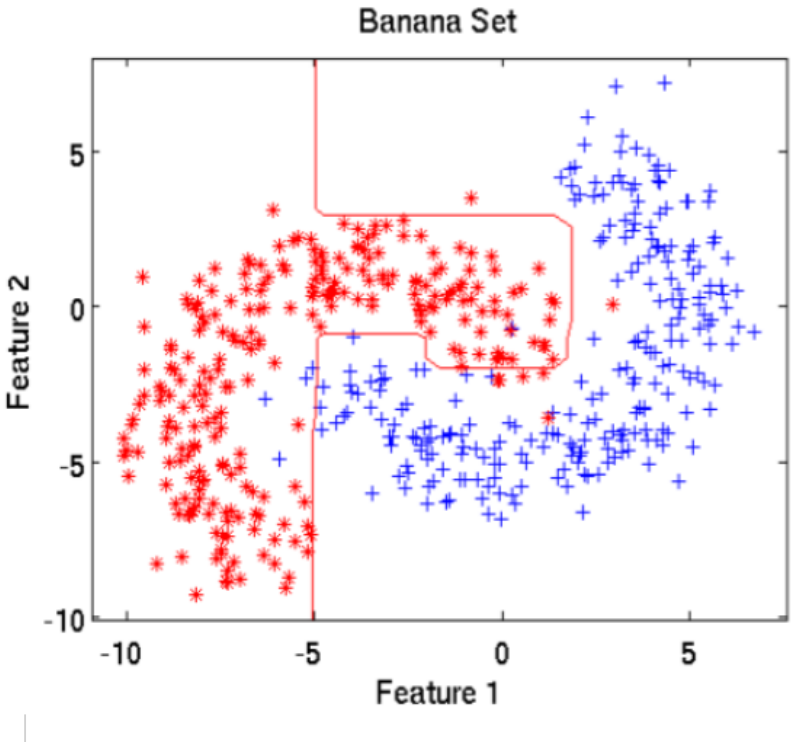
Ensemble learning ▶ Bagging

- Fase 1: Generación de modelos
  1. Sea *n* el número de ejemplos en la BD y *m* el número de los modelos a utilizar
  2. Para *i*=1,…,*T* hacer
     - Muestrear con reemplazo *n* ejemplos de la BD
     - Aprender un modelo con ese conjunto de entrenamiento
     - Almacenarlo en modelos[*i*]
- Fase 2: Clasificación
  1. Para cada ejemplo de test
     - Para i=1,…,T hacer
       - Predecir la clase utilizando modelos[i]
     - Regresión: devolver la media
     - Clasificación: devolver el voto de la mayoría

Ensemble learning ▶ Bagging ▶ Ejemplo

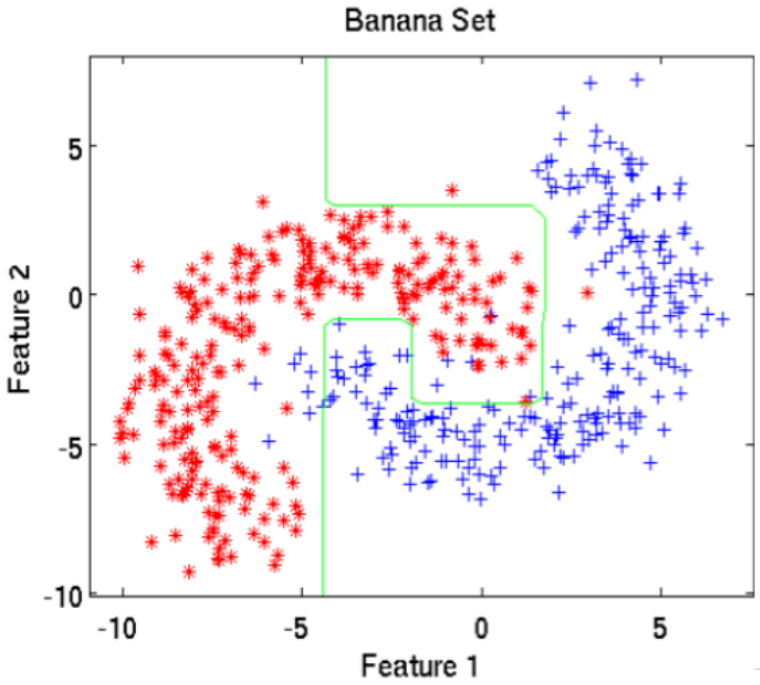


Conjunto de datos

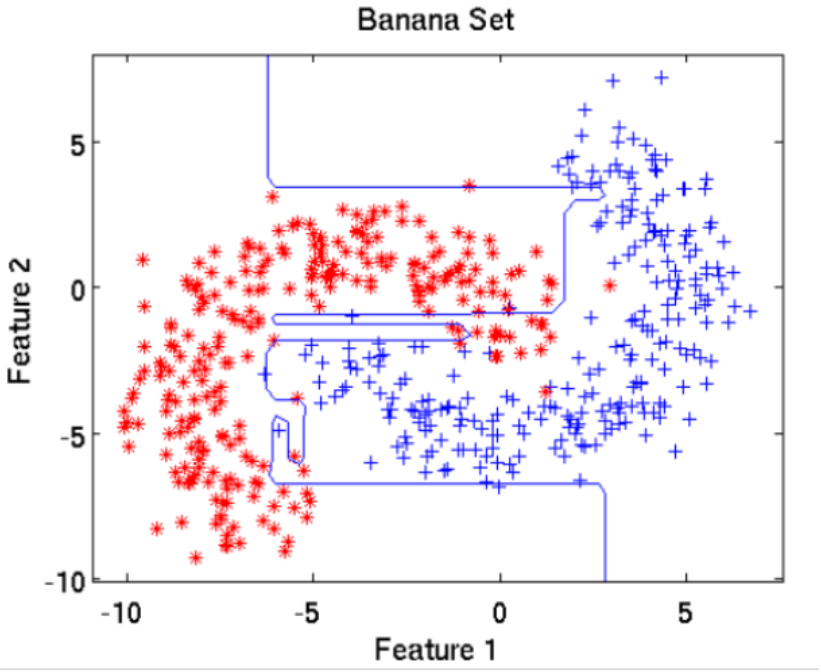


Frontera de decisión generada por árbol 1

Ensemble learning ▶ Bagging ▶ Ejemplo

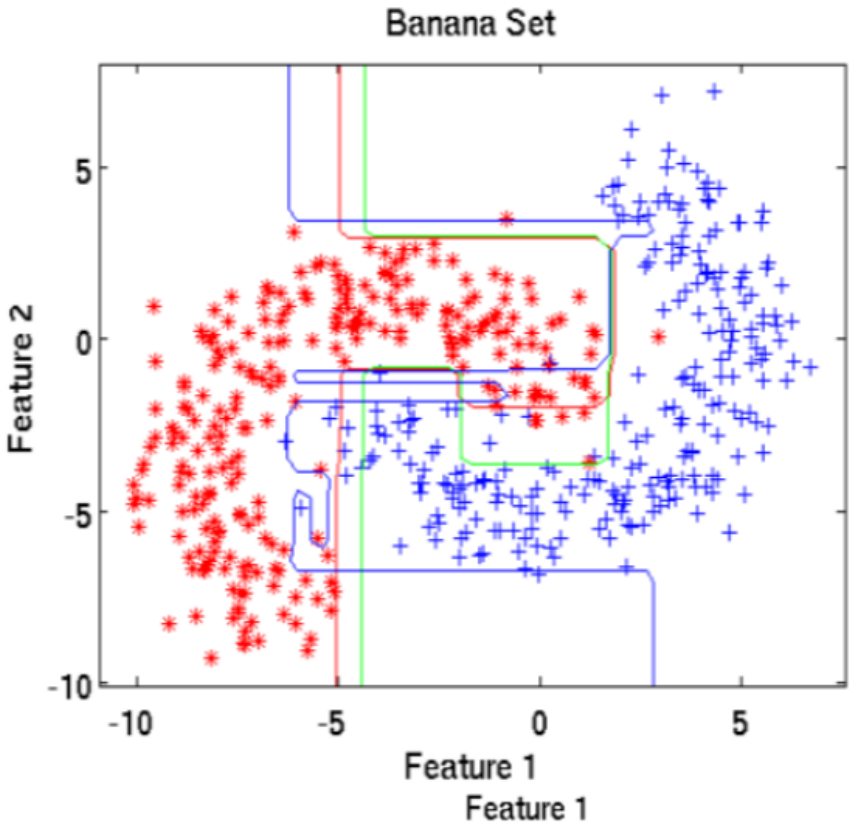


Frontera de decisión generada por árbol 2

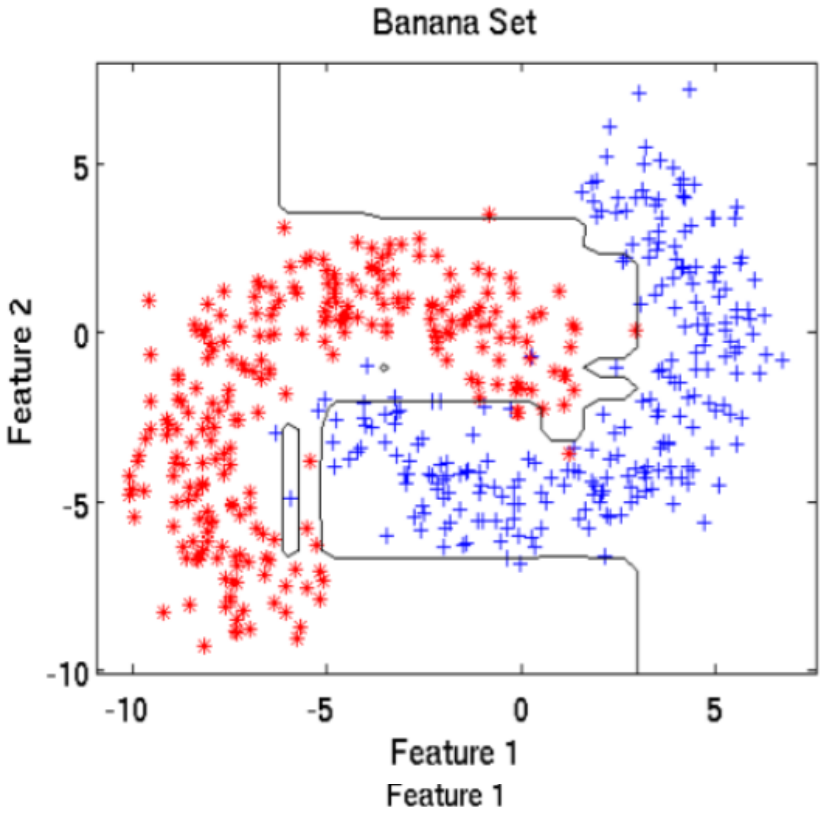


Frontera de decisión generada por árbol 3

Ensemble learning ▶ Bagging ▶ Ejemplo

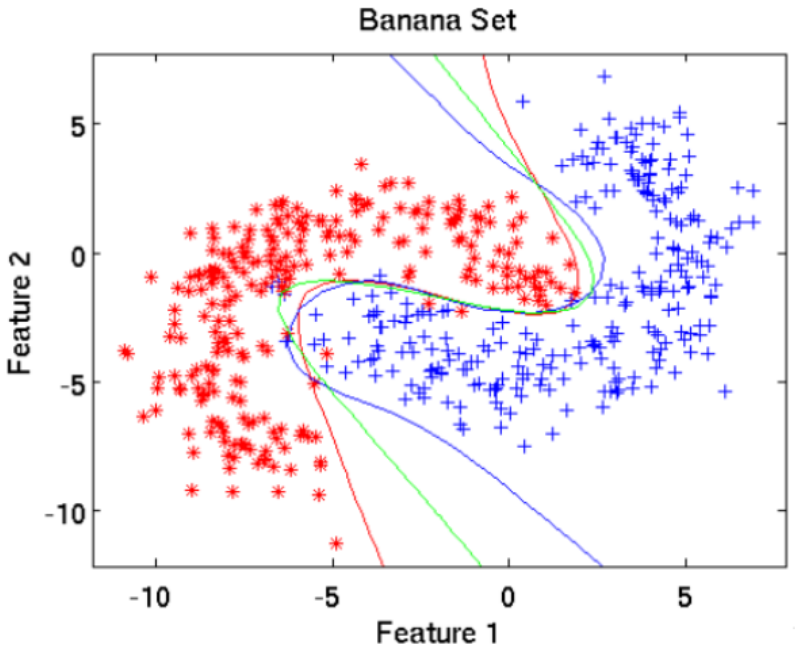


Frontera de decisión de todos los arboles

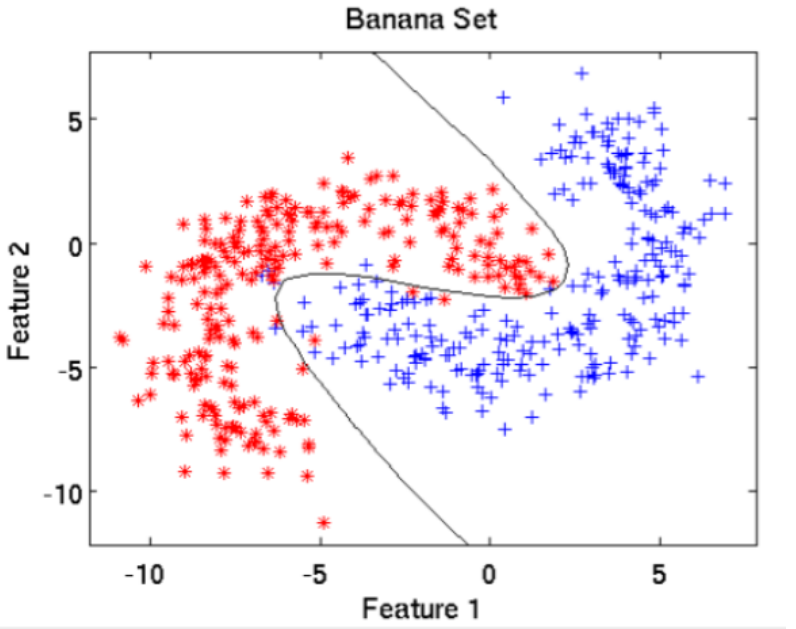


Frontera de decisión resultante del bagging de los cuatro modelos

Ensemble learning ▶ Bagging ▶ Ejemplo



Frontera de decisión generada por tres redes neuronales



Frontera de decisión resultante del bagging de los tres modelos neuronales

Ensemble learning ▶ Bagging

¿Por qué funciona?

- El error en aprendizaje se debe a ruido, sesgo y varianza
  - Ruido: error en la función objetivo
  - Sesgo: donde el algoritmo no puede aprender el objetivo
  - Varianza: viene dada por el muestreo y la forma en que éste afecta al algoritmo de aprendizaje
- Bagging minimiza especialmente el error de varianza

Ensemble learning ► Bagging ► **Random forest**

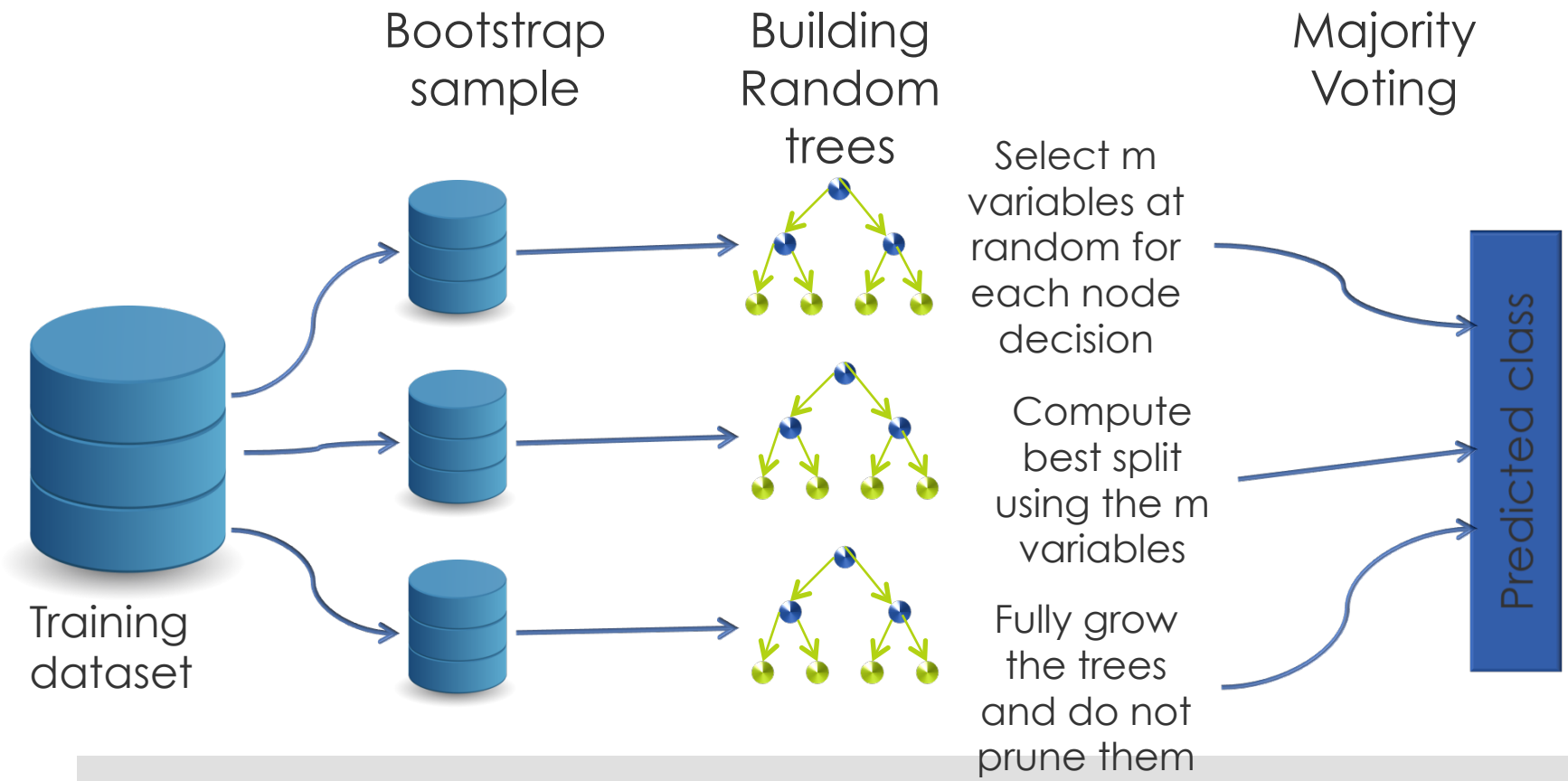
- Uno de los métodos basados en bagging más utilizado

  L. Breiman. Random Forests. Machine Learning 45: 5-32. 2001

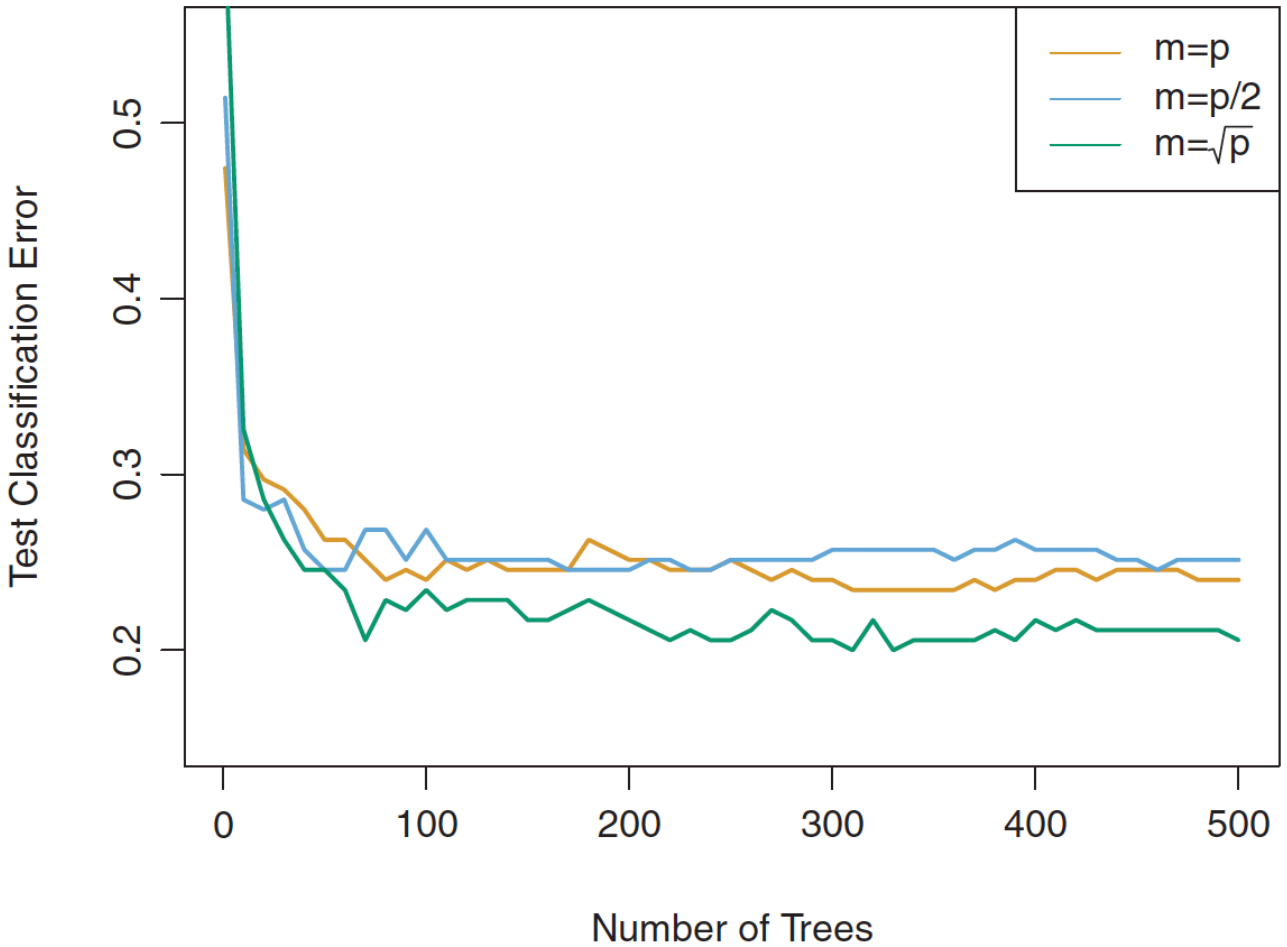
- **Idea básica**
  - Hacer boostrap sobre los ejemplos
  - Para cada subconjunto de ejemplos, construir un árbol pero sólo con un subconjunto (aleatorio) de las variables predictoras
  - Predecir la salida de un nuevo ejemplo
    - Clasificación: voto de la mayoría
    - Regresión: la media
  - Si p = nº total de variables predictoras,
    - m= $\sqrt{p}$ → Clasificación
    - m=1/2 p; m= 1/3 p → Regresión
    - m= ½ $\sqrt{p}$; m= 2$\sqrt{p}$

  Si m=p, Random Forests = bagging con árboles

Ensemble learning ► Bagging ► **Random forest**



Ensemble learning ►Bagging ► Random forest



Resultados para el dataset gene expression (15 clases, 500 variables) con RandomForest

An Introduction to Statistical Learning with Applications in R. Springer, 2013.

Ensemble learning ► Bagging ► Random forests

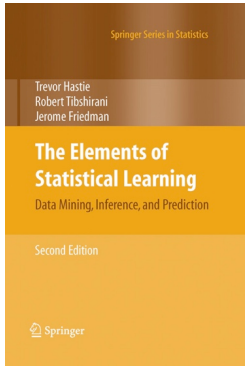
Ventajas:

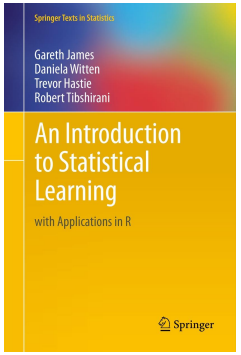
- Al forzar que cada árbol considere sólo un conjunto de variables, el ensemble estará formado por árboles menos correlados y con mayor varianza
  - El ensemble funciona mejor
- Junto con métodos basados en boosting (Adaboost) es uno de los algoritmos de analítica predictiva con mejores resultados

Inconvenientes:

- Dificultad para interpretar los resultados
- Se debe evitar el sobreaprendizaje

Bibliografía





- The Elements of Statistical Learning: Data Mining, Inference, and Prediction, by T. Hastie, R. Tibshirani and J. Friedman. 2009.

- An Introduction to Statistical Learning with Applications in R, by G. James, D. Witten, T. Hastie. Springer. Springer. 2013.

Más información

- Machine Learning Resources from Science

http://www.sciencemag.org/site/feature/data/compsci/machine_learning.xhtml