Tema 4 - Clasificación, regresión y series temporales

- Aprendizaje supervisado
- ▼ Clasificación (problemas categóricos)
  - ▼ Criterios de evaluación de un clasificador
    - ▼ Medidas calculadas
      - ▼ Precisión/exactitud
        - Acierto → $s = \frac{VP+VN}{VP+FP+VN+FN}$
        - Error → $\varepsilon = 1 - s$
      - ▼ True Positive Rate (TPR)
        - Proporción de positivos predichos respecto al total de positivos reales

        $TPR = \frac{TP}{P} = \frac{TP}{TP+FN}$
      - ▼ True Negative Rate (TNR)
        - Proporción de negativos predichos respecto al total de negativos reales

        $TNR = \frac{TN}{N} = \frac{TN}{TN+FP}$
      - ▼ False Positive Rate (FPR)
        - Proporción de falsos positivos respecto al total de negativos reales

        $FPR = \frac{FP}{N} = \frac{FP}{TN+FP}$
      - ▼ False Negative Rate (FNR)
        - Proporción de falsos negativos respecto al total de positivos reales

        $FNR = \frac{FN}{P} = \frac{FN}{TP+FN}$
      - ▼ Positive Predictive Value (PPV)

- Proporción de verdaderos positivos respecto al total de positivos predichos

$$PPV = \frac{TP}{TP+FP}$$

▼ Area Under Curve (AUC)

$$AUC = \frac{1+TPR-FPR}{2}$$

▼ G-mean

$$G-mean = \sqrt{TPR \times TNR}$$

▼ G-measure

$$Gmeasure = \sqrt{PPV \times TPR}$$

▼ F1-score

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN}$$

- Velocidad
- Escalabilidad
- Interpretabilidad
- Complejidad

▼ Métodos de validación

▼ Hold-out

- BD de gran tamaño
- División en conjuntos de entrenamiento/test

▼ Validación cruzada

- BD de tamaño moderado
- K subconjuntos de igual tamaño
- K clasificadores, cada uno con su propio CE → Validación de cada clasificador con su propio CT

▼ Leaving-one-out

- BD de tamaño pequeño
- Validación cruzada con K = nº registros

▼ Clasificadores

▼ Basados en instancias

▼ Algoritmos → K-NN

- Conjunto de datos de entrenamiento → BD inicial (no hay entrenamiento)

▼ Cálculo de distancia euclídea de un punto a todos los ejemplos para quedarse con los N más cercanos

- Distancia euclídea → $d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
- Todas las variables deben ser numéricas → requiere selección de características
- Robusto frente a ruido, pero ineficiente en memoria

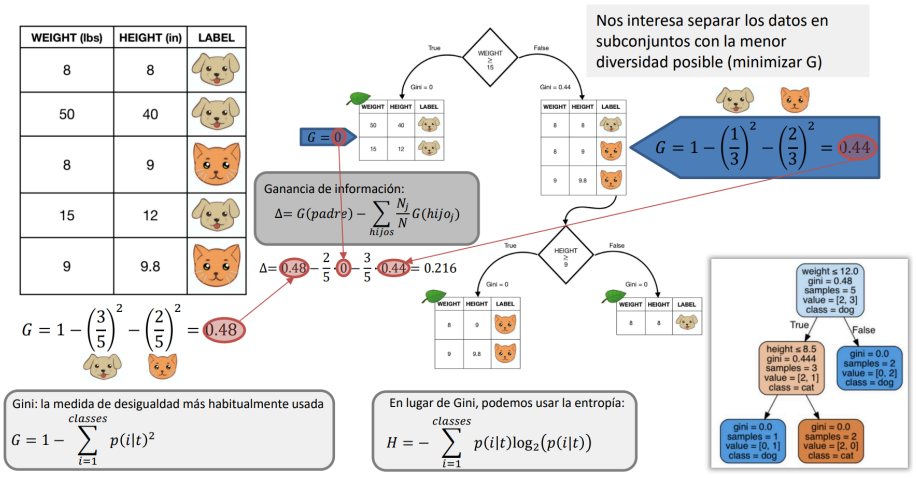
▼ Basados en árboles de decisión

▼ Características

▼ Estructura

- Dividen atributos en intervalos que se pueden recalcular en un siguiente nivel
- Prepoda hasta conseguir el conjunto de datos (también puede haber postpoda)
- Objetivo → mayor diversidad entre cada grupo generado al dividir el conjunto de datos

▼ Medidas



▼ Entropía (con K clases)

- $S = \sum_{i=1}^{k} p_i \times \log_2(\frac{1}{p_i})$

▼ Gini

- $G = 1 - \sum(\frac{resultado.parcial}{datos.totales})^2$

▼ Ganancia de información

- $\Delta = G(nodo.padre) - \sum_{hijos}(\frac{resultado.parcial.nodo}{resultado.total.nodo})^2$

▼ Razón de ganancia

- $GainRatio = \frac{\Delta}{S} = \frac{G(nodo.padre) - \sum_{hijos}(\frac{resultado.parcial.nodo}{resultado.total.nodo})^2}{\sum_{i=1}^{k} p_i \times \log_2(\frac{1}{p_i})}$

▼ Ventajas/desventajas

Ventajas	Desventajas
Eficiencia	Sobreaprendizaje
Escalabilida	Trato de valores perdidos
Trato de ruido	

▼ Algoritmos

▼ ID3

- No puede manejar variables continuas
- Elige el atributo con mayor ganancia de información de forma recursiva
- Entropía = $\frac{1}{pureza}$ → Algoritmo termina cuando Entropía = 0
- Al calcular la pureza/ganancia/diversidad se añade un peso normalizado → No es lo mismo conseguir pureza en un subconjunto mayoritario que en uno minoritario
- Preferible construir árboles cortos

▼ C4.5

▼ Refinamiento de ID3

        - Propone soluciones para sobreaprendizaje
        - Tratamiento de valores continuos y valores perdidos
      - ▼ Utiliza razón de ganancia en términos de cociente
        - Evita priorizar atributos que generan más categorías
      - Útil como base para otros algoritmos
      - No maneja bien clases desbalanceadas
      - Funciona bien para datos ruidosos
- ▼ Basados en reglas
  - ▼ Características
    - ▼ Intenta generar reglas que cubran exactamente una clase
      - Mejor regla posible → regla más simple/interpretable
    - No cualquier conjunto de reglas puede convertirse en un árbol de decisión → puede haber antecedentes que no estén en todas las reglas
    - No siempre tiene que haber una clase al final → puede haber un reparto de peso
  - ▼ Algoritmos → PRISM
    - Fija un consecuente (clase) y busca el atributo con el que se obtienen mejores reglas → mayor acierto
    - ▼ Analiza todas las clases
      - Cubre todos los ejemplos de entrenamiento
      - Genera reglas para todas las categorías
    - ▼ No se define una regla hasta que la exactitud es total
      - Valora mejor un 1/1 que un 19/20 → sobreajusta los datos
    - ▼ Atributos
      - Numerador → nº de casos del consecuente
      - Denominador → nº de casos del antecedente
    - Resultado independiente del orden de clases que se analizan

▼ Métodos bayesianos

- Supone que los atributos son independientes entre sí
- Basados en probabilidades

▼ Cálculo de probabilidades

▼ Estimación de máxima verosimilitud (EMV)

- Cociente entre el nº de instancias de la clase y el nº total de instancias
- $p(x|x_i) = \frac{n(x_i,x_j)}{n(x_j)}$

▼ Correlación de Laplace

- Suma 1 en el numerador y el nº de clases en el denominador
- $p(x|x_i) = \frac{n(x_i,x_j)+1}{n(x_j)+|\Omega_{x_i}|}$

▼ Ventajas/desventajas

Ventajas	Desventajas
Fácil de implementar	Falta de precisión por asumir que las variables son independientes
Buenos resultados	

▼ Redes neuronales

- Necesitan valores continuos
- Aprendizaje por refuerzo → suele acabar en sobreajuste
- Poca interpretabilidad

▼ Ensemble learning

▼ Características

- Combinar varios clasificadores del mismo tipo
- Cada modelo es débil para conseguir una especialización individual

▼ Técnicas

▼ Bagging

- Funciona bien para árboles de decisión → cada pequeño cambio en CE provoca grandes cambios
- No todos los clasificadores ven los mismos datos

▼ Boosting

- Muestreo ponderado → concretar aprendizaje en ejemplos más difíciles
- Voto ponderado → produce un clasificador más fuerte
- Presta más atención a ejemplos mal clasificados

▼ Estructura

▼ Generación de modelos

- Aprender un modelo y almacenarlo
- Calcular error
- Normalizar pesos de todos los ejemplos

▼ Clasificación

- Asignar peso 0 a todas las categorías de la variable clase
- Calcular peso de cada categoría según el error
- Devolver categoría con más peso

▼ Problemas multiclase

▼ One vs One

- Problema binario para cada par de clases

▼ One vs All

- Reduce nº de problemas binarios
- Frecuencia de datos baja respecto al “all” → clases desbalanceadas

▼ Regresión (problemas continuos)

▼ Características principales

- Objetivo → predecir el valor numérico para una variable a partir de los valores de otras

- Definición parecida a clasificación, aunque en regresión la mayoría de las variables son numéricas

▼ Validación de algoritmos de regresión

- Todas las técnicas de validación usadas en clasificación son validas para regresión, pero se debe medir el error de otra manera

▼ Técnicas de clasificación válidas para regresión

▼ Métodos basados en instancias

▼ KNN

▼ Valores a devolver para el objeto analizado

- Si todos los objetos cuentan igual → $v = \frac{\sum_{i=1}^{k} v_i}{k}$
- Si se hace un voto ponderado → $v = \frac{\sum_{i=1}^{k} w_i \times v_i}{\sum_{i=1}^{k} w_i}$

▼ Métodos basados en redes neuronales

- Capa de salida compuesta únicamente por una neurona
- Los pesos se adaptan en función del error cometido → basta con medir de forma adecuada el error

▼ Análisis de regresión

- Método más utilizado para predicción numérica
- Objetivo → estimar variable objetivo como una ecuación que contiene al resto de variables como incógnitas

▼ Regresión lineal ($y = a + b \times x$)

- Modelo más sencillo

▼ Obtención de coeficientes mediante método de mínimos cuadrados

- $b = \frac{\sum_{i=1}^{S}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{S}(x_i - \bar{x})^2}$
- $a = \bar{y} - b \times \bar{x}$

▼ Regresión lineal múltiple ($y = a + b_1 x_1 + b_2 x_2 + ... + b_n x_n$)

- Útil cuando hay más de una variable predictora

- Estimación de coeficientes más compleja → requiere operar con matrices

▼ Regresión exponencial ($y = a \times e^{bx}$)

- Útil para estimar curvas

▼ Obtención de coeficientes mediante logaritmos

- $ln(y) = ln(a \times e^{bx}) \rightarrow ln(y) = ln(a) + ln(e^{bx}) \rightarrow y^* = a_* + bx$

▼ Árboles de regresión y árboles de modelos

▼ En árboles de decisión, las hojas pueden ser...

▼ Números → Árboles de regresión

- Árbol de decisión cuyas hojas predicen una cantidad numérica
- Esa cantidad numérica se calcula como media del valor para la variable dependiente de todos los ejemplos que han llegado a esa hoja durante la construcción del árbol
- Evaluación de nuevo ejemplo → idéntica a los árboles de decisión
- Es posible utilizar suavizado de valores a tratar → salvar posibles discontinuidades presentes en los datos

▼ Criterio de selección de una variable

- Basado en reducción del error esperado
- Reducción de desviación/varianza en la variable objetivo
- $SDR = sd(T) - \sum_i \frac{|T_i|}{|T|} \times sd(T_i)$

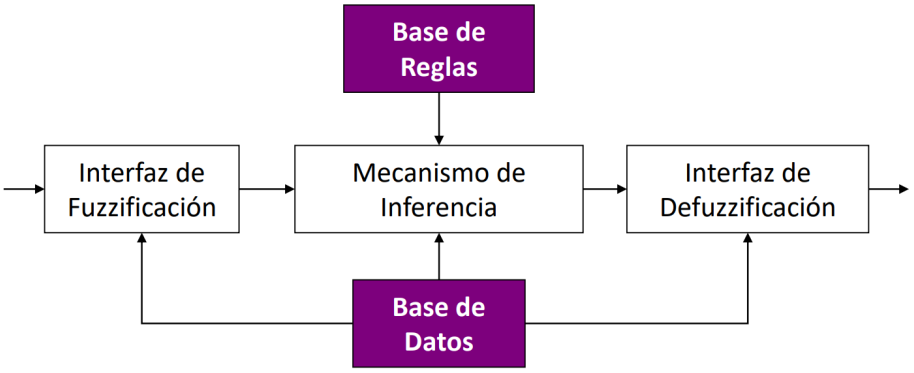
- Finalmente, el árbol se poda para evitar sobreajuste

▼ Combinaciones lineales o redes neuronales → Árboles de modelos

- Árboles de regresión en los que la poda se realiza en mayor medida
- En las hojas, en lugar de un valor numérico, contienen una ecuación de regresión local

▼ Sistemas basados en reglas difusas

▼ Esquema general



▼ Interfaz de fuzzificación

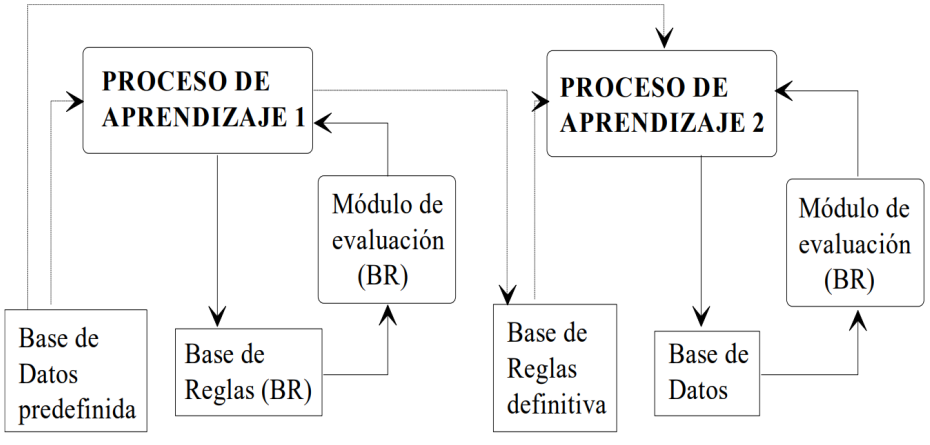
- Convierten un valor numérico normal a un valor difuso

▼ Interfaz de defuzzificación

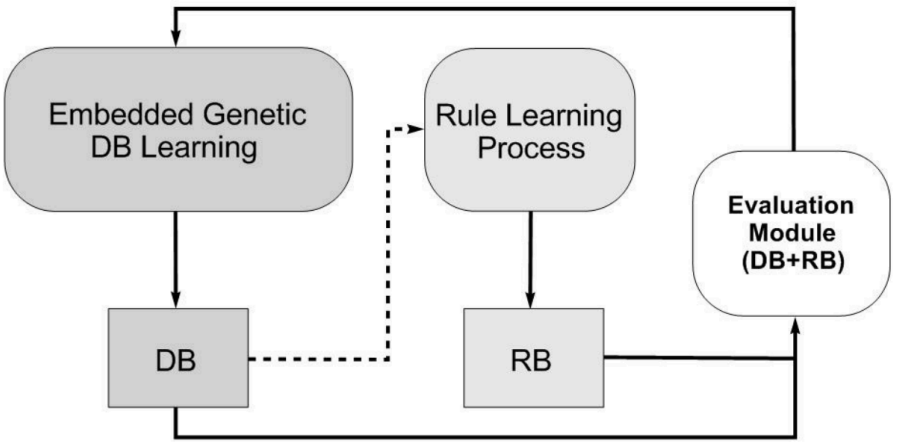
- Convierte el valor difuso obtenido a un valor numérico

▼ Enfoques adicionales

▼ Mejorar la definición de la BD una vez aprendida la base de reglas



▼ Esquema de meta-aprendizaje para aprender el número de términos lingüísticos



- Stacking → Agregación de distintos modelos de regresión

▼ Series temporales

- Objetivo → predecir el futuro en base a información histórica y a eventos futuros que puedan impactar en el resultado

▼ Componentes

▼ Tendencia (Trend → T)

- Representa el cambio a largo plazo en los datos

▼ Estacionalidad (Seasonal → S)

- Refleja patrones repetitivos y predecibles a intervalos regulares de tiempo

▼ Ciclo (Cycle → C)

- Variaciones que ocurren a lo largo de un período más prolongado que la estacionalidad
- Generalmente asociadas a factores económicos o sociales

▼ Ruido (Remainder → R)

- Fluctuaciones aleatorias e impredecibles que no pueden explicarse por los otros componentes
- Representan variabilidad no estructurada en los datos

▼ Descomposiciones de líneas temporales

- Aditiva → $y_i = S_t + Tt + Rt$
- Multiplicativa → $y_i = S_i \times T_t \times Rt$

▼ Factores que afectan a la predicción

- Tiempo horizonte → cuánto podemos predecir

▼ Tipos de patrones de datos

▼ Requisitos para aplicar predicción cuantitativa

- Disponer de datos numéricos pasados
- Poder asumir que algunos aspectos de los patrones del pasado pueden continuar en el futuro