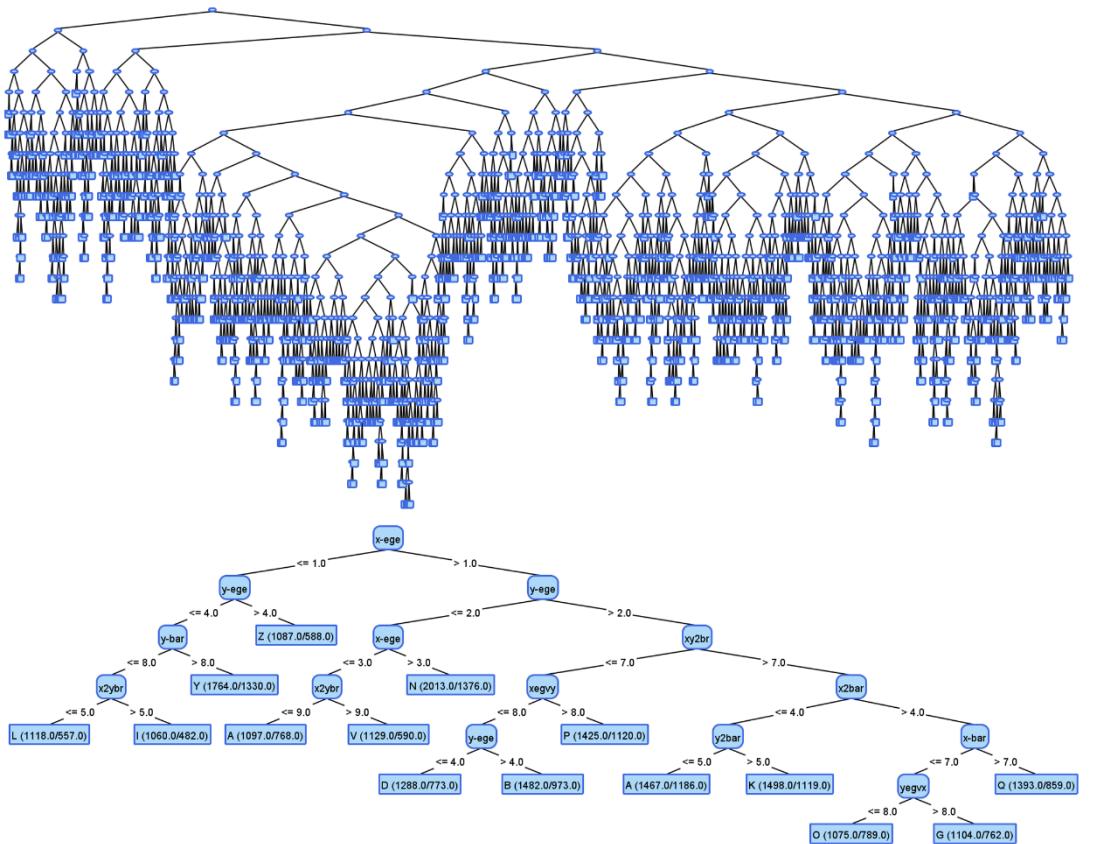


CLASIFICADORES BASADOS EN ÁRBOLES



Aprendizaje Maquinal

Segundo Cuatrimestre - 2022 -

ALGORITMOS DE CLASIFICACIÓN

Perezosos

- IB1, IBk (KNN), KStar, LBR, LWL

Bayesianos

- Naive Bayes, Bayesian Logistic Regression, BayesNet, Redes Bayesianas

Funciones

- SVM, Regresión Lineal/ Logística, MLP, RBF, SGD

Árboles

- Id3, J48 (**C4.5**), Random Forest, XGBoost

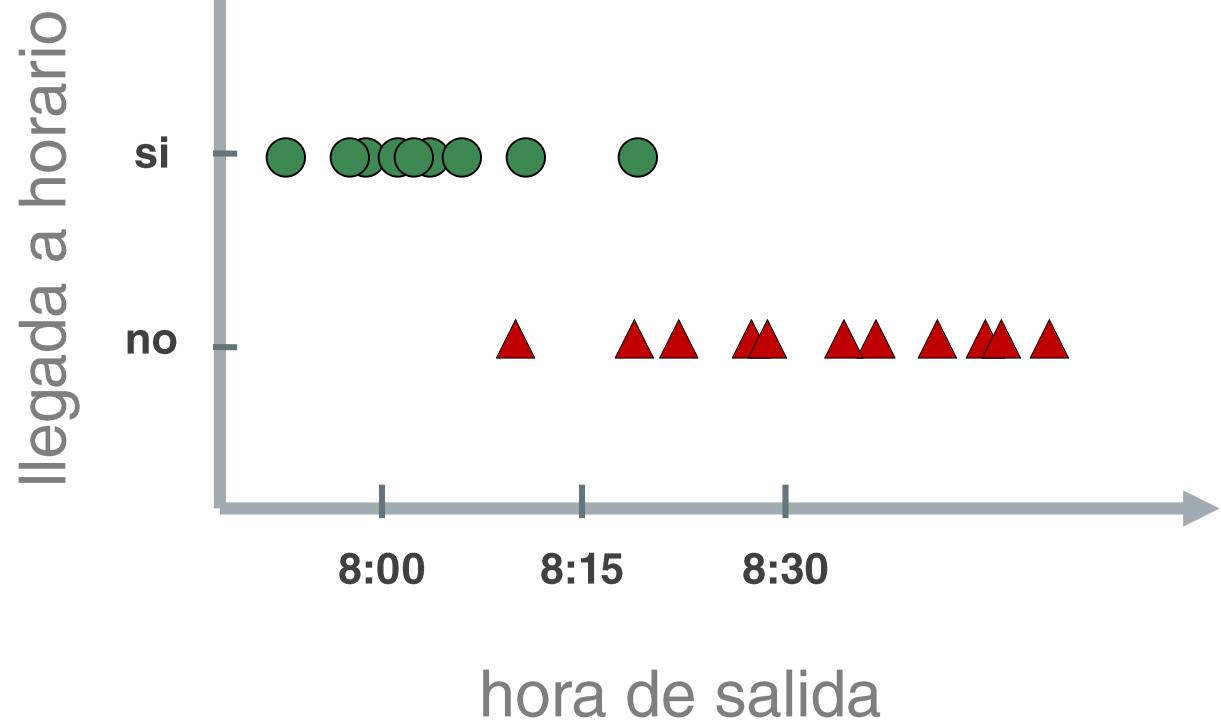
Reglas

- Conjunctive Rule, Decision Table, DTNB, FURIA, JRip, M5Rules, NNge, OneR, PART, Prism, Ridor, ZeroR

Meta

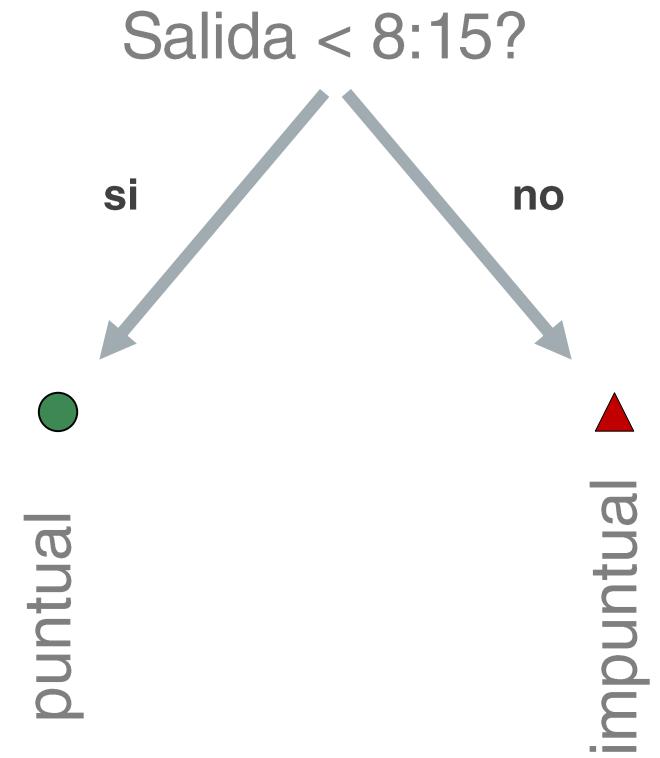
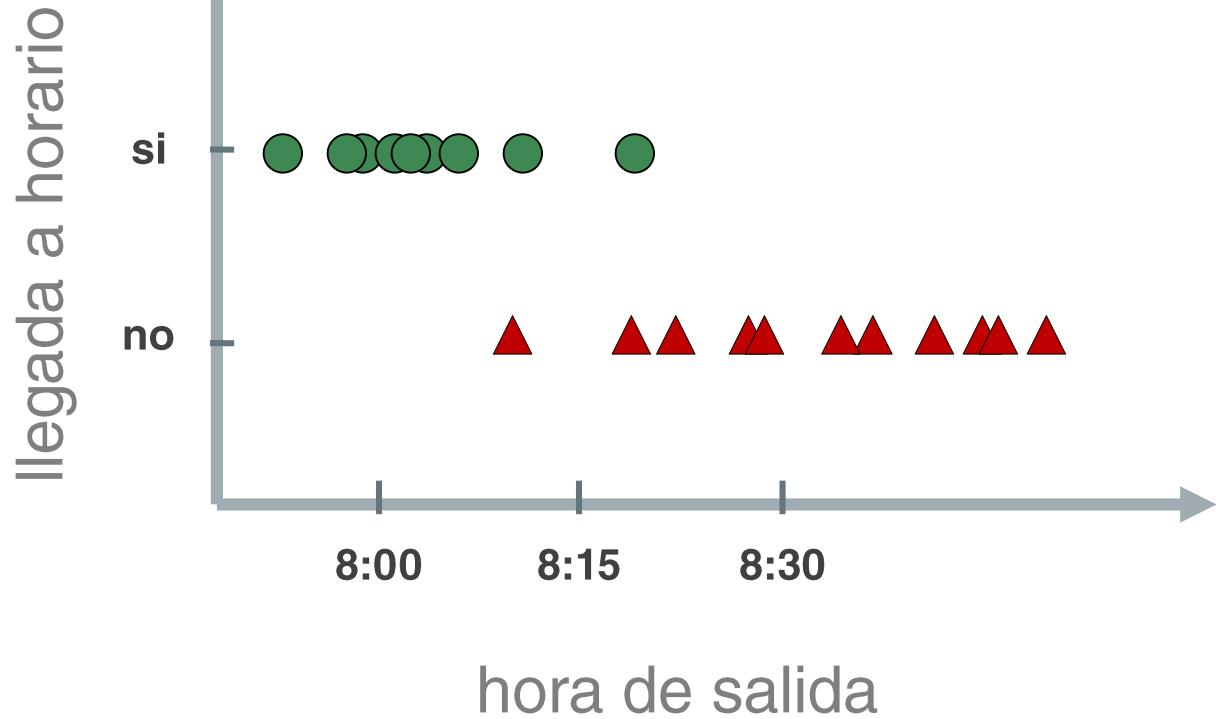
- AdaBoost, Additive Regression, Bagging, Dagging, GridSearch, Random Committee, Rotation Forest, Stacking

EJEMPLO

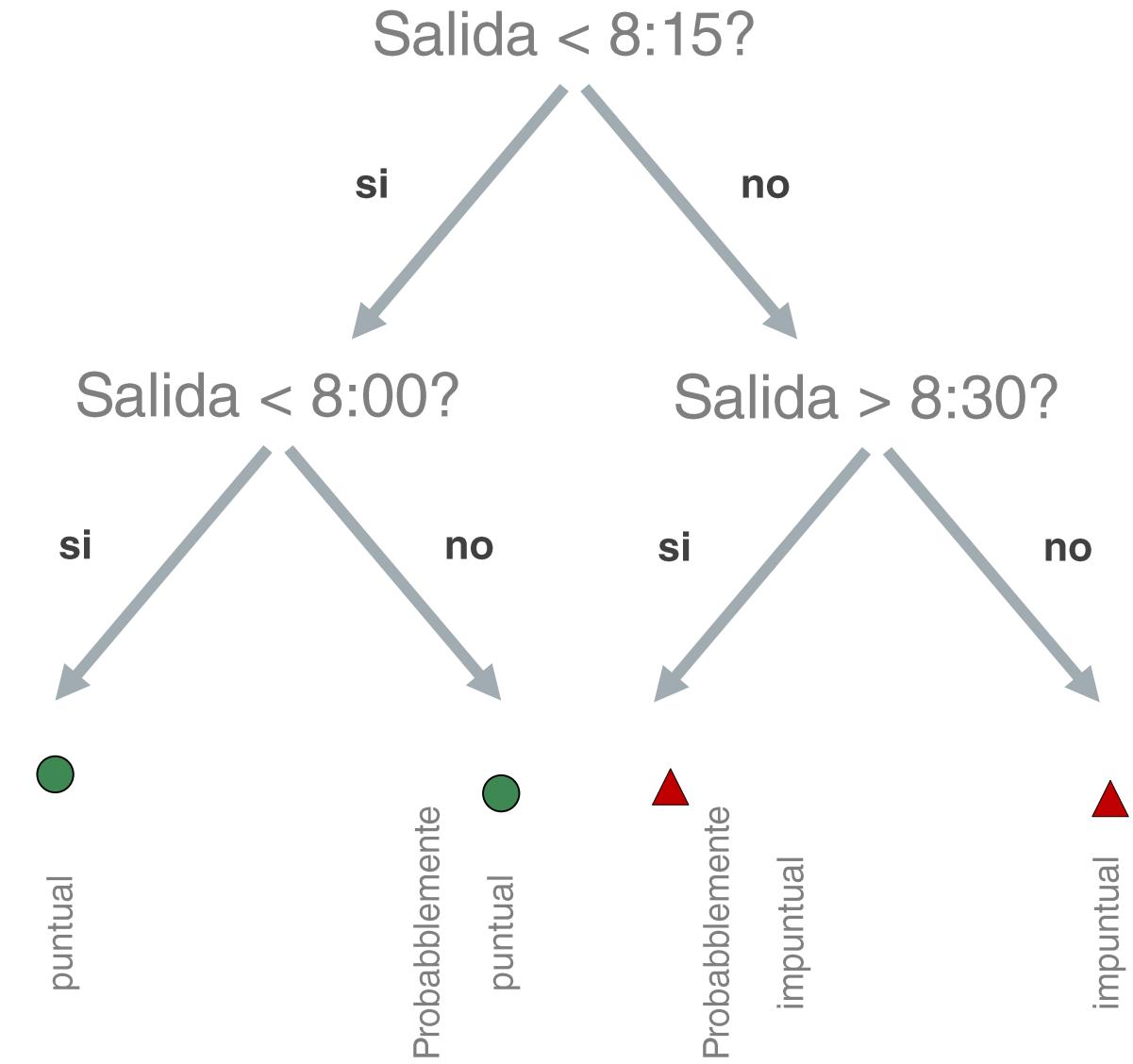
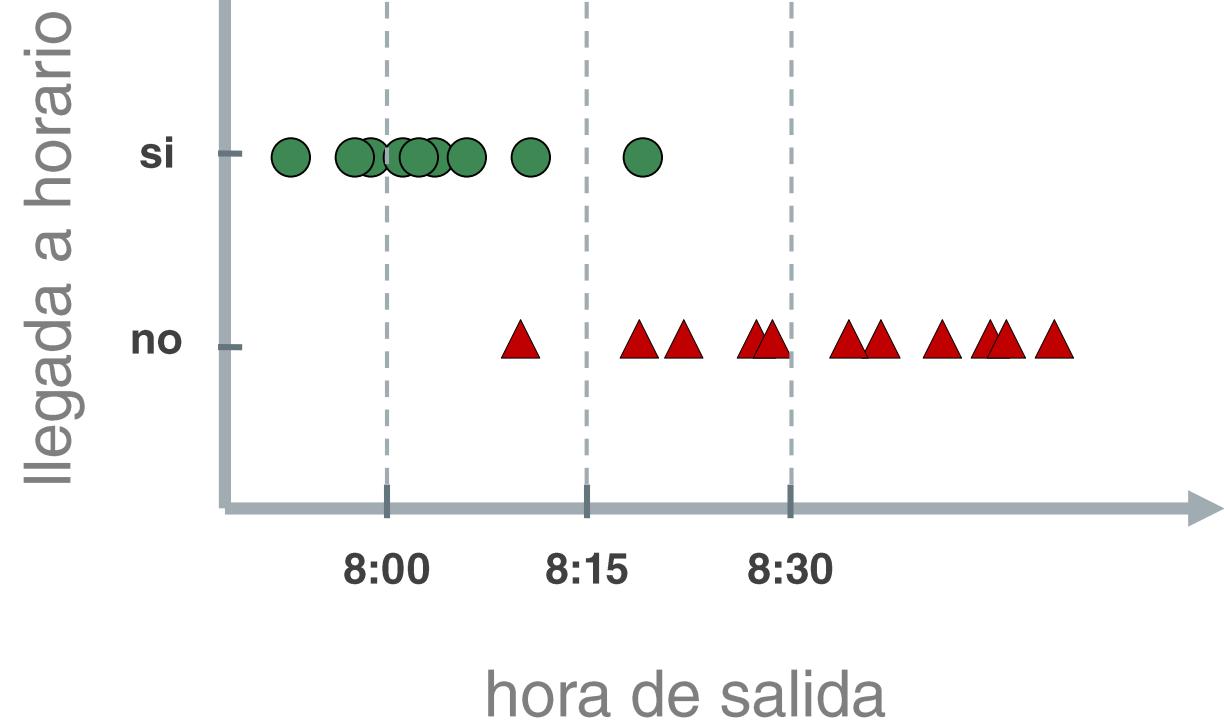


Hora salida	Llegada a horario
8:21	Si
8:05	Si
7:48	Si
8:41	No
7:53	Si
8:18	No
...	...

EJEMPLO

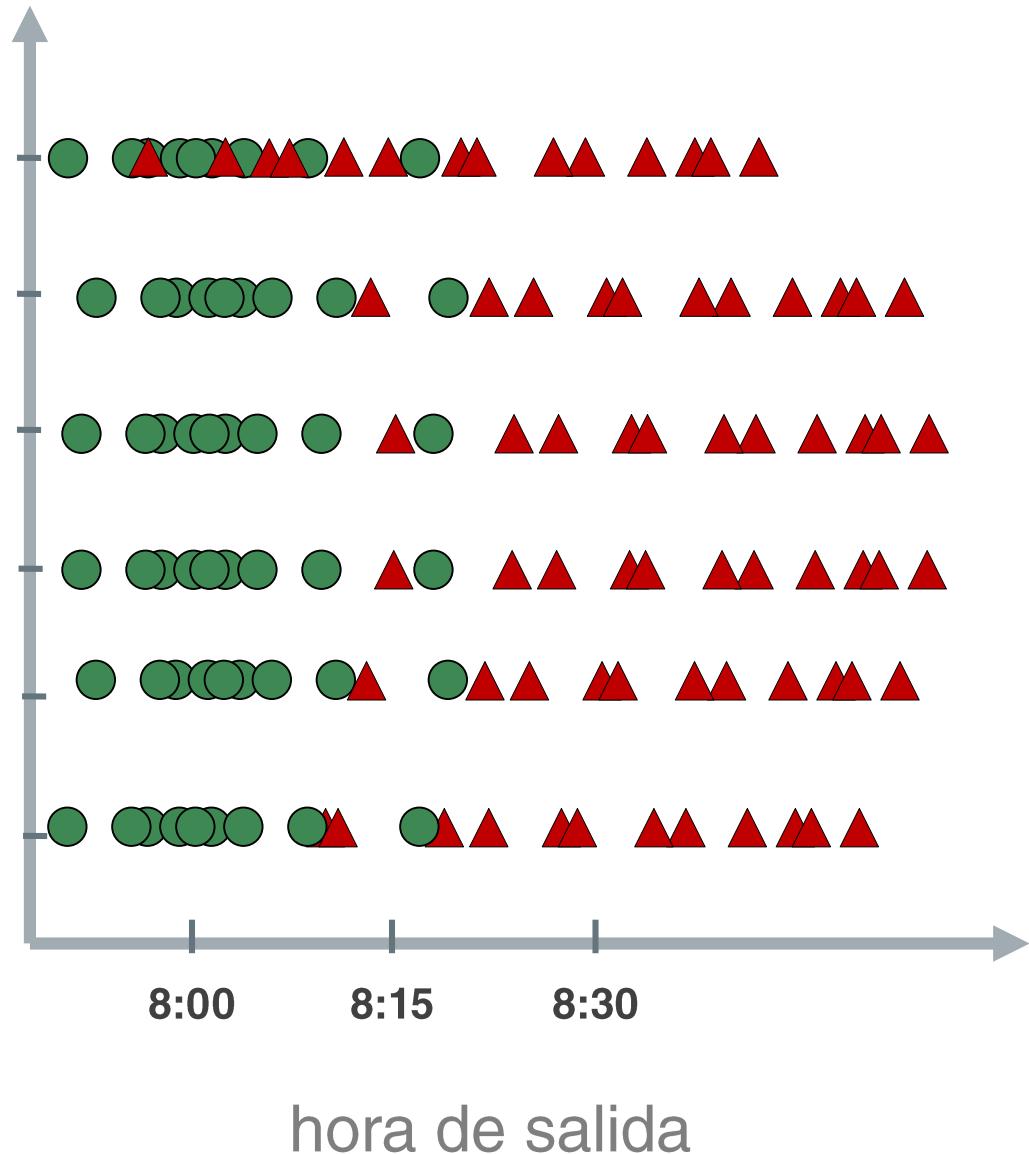


EJEMPLO



EJEMPLO

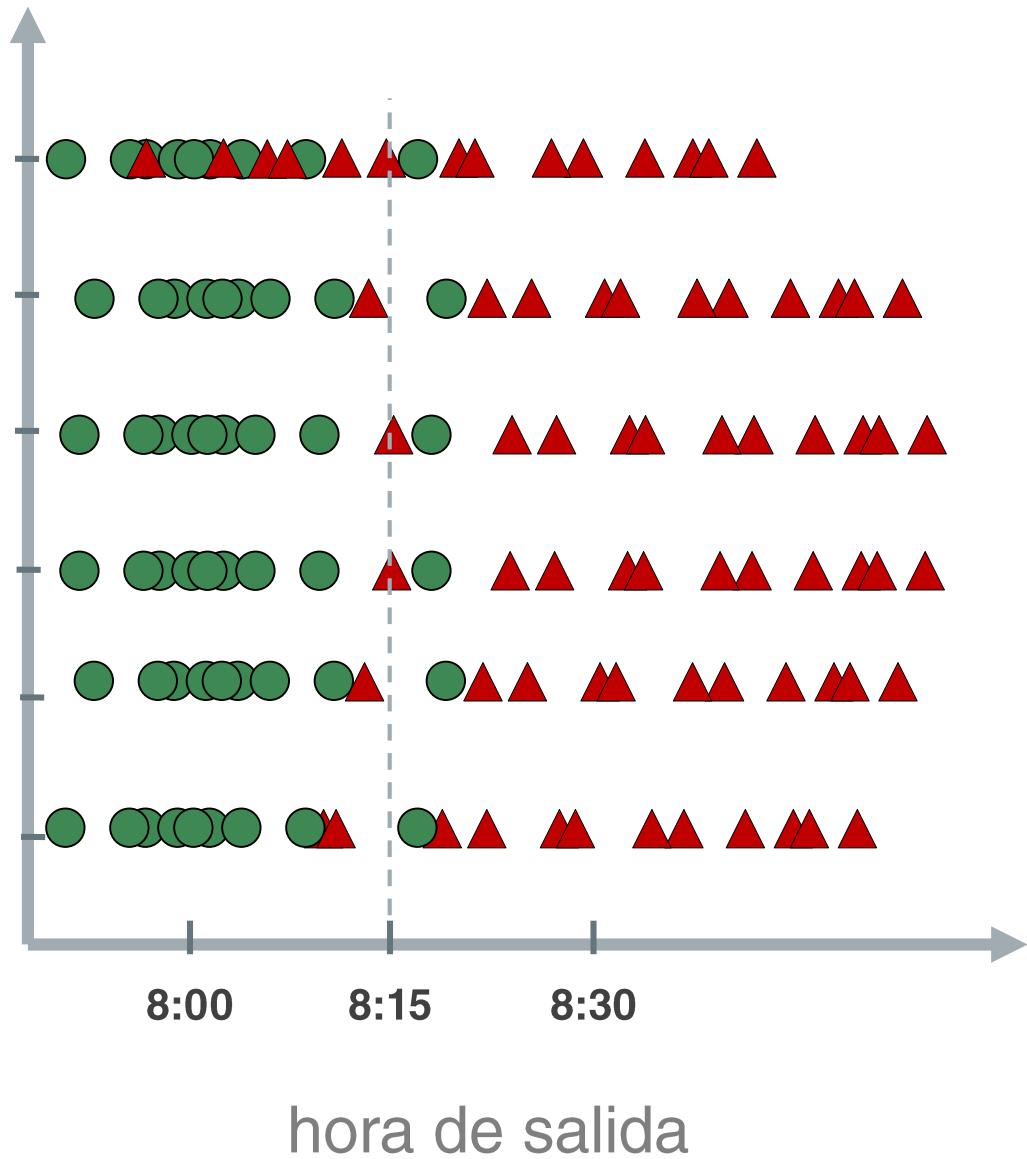
día de la semana



Día	Hora salida	Llegada a horario
1	8:21	Si
1	8:05	Si
5	7:48	Si
6	8:41	No
4	7:53	Si
3	8:18	No
...

EJEMPLO

día de la semana



Salida < 8:15?

si

puntual

no

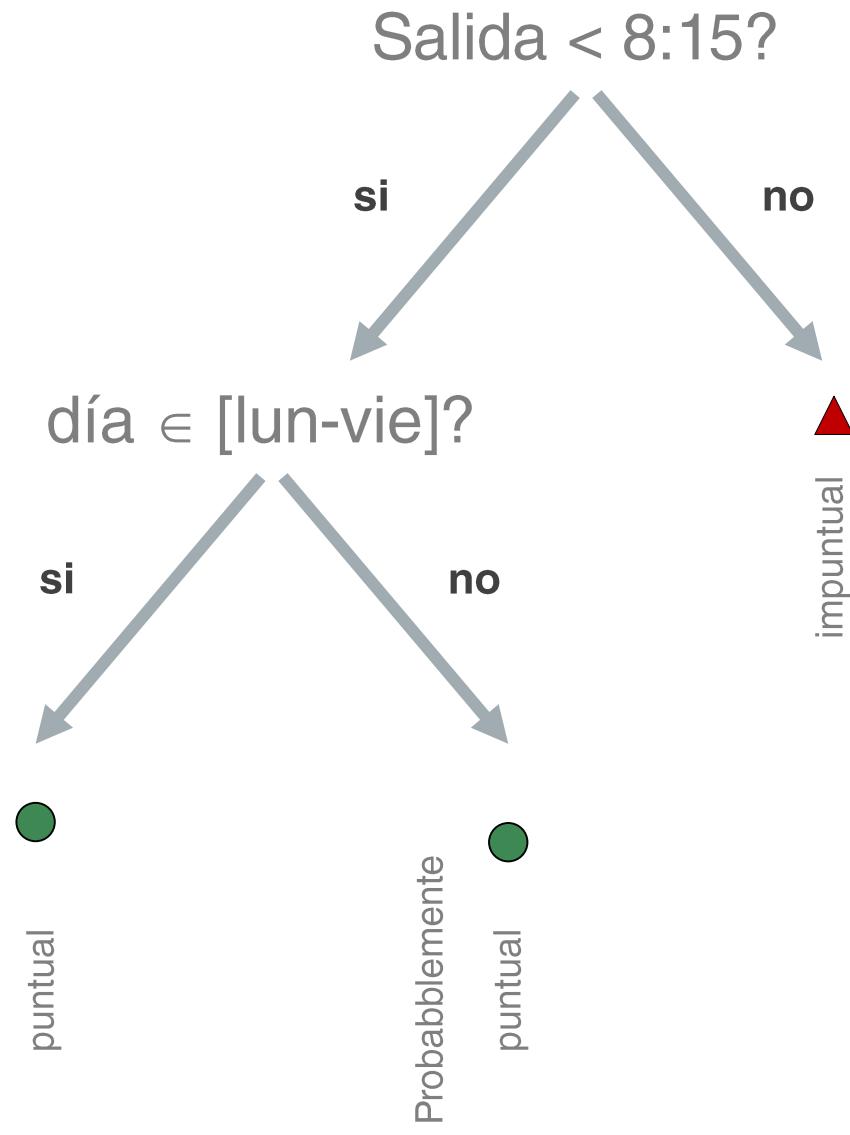
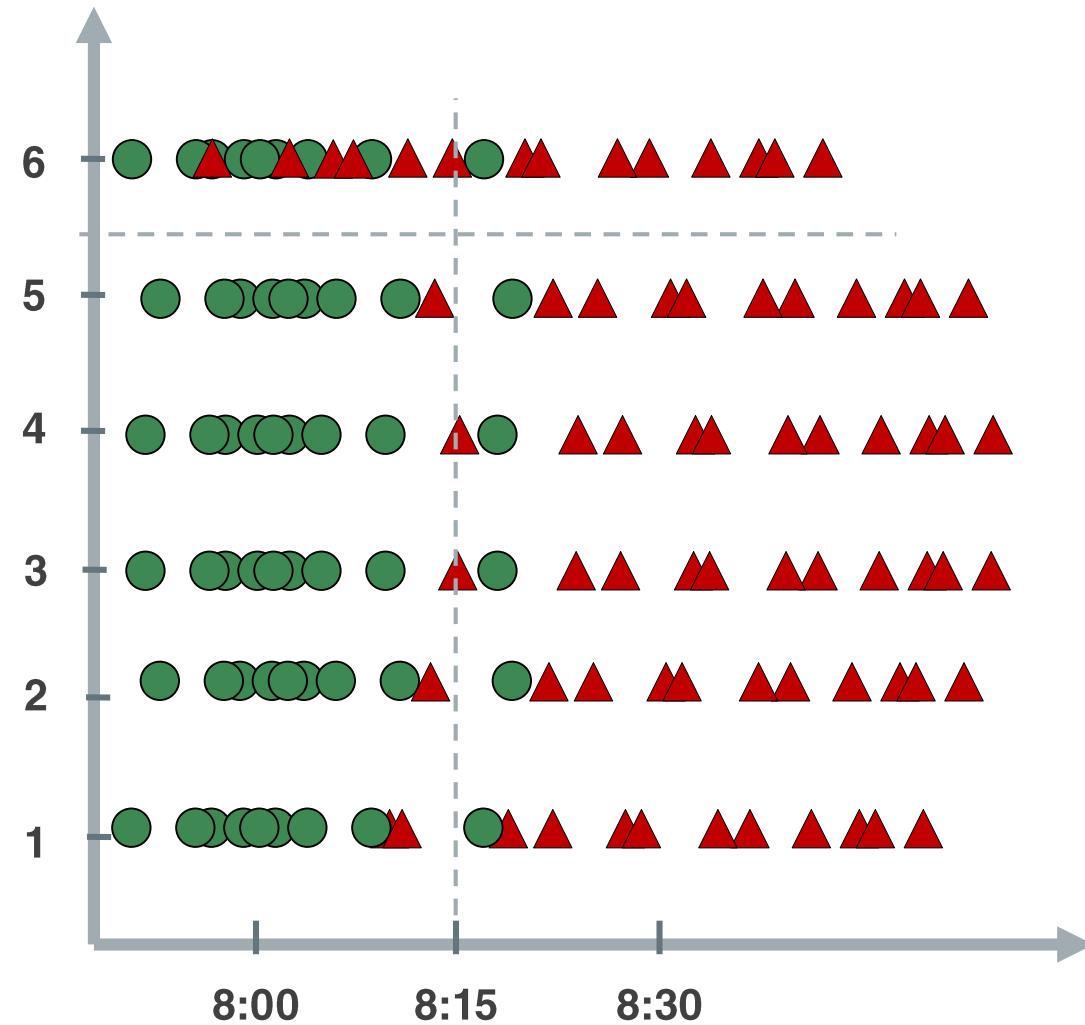
impuntual

EJEMPLO

día de la semana

8:00 8:15 8:30

hora de salida

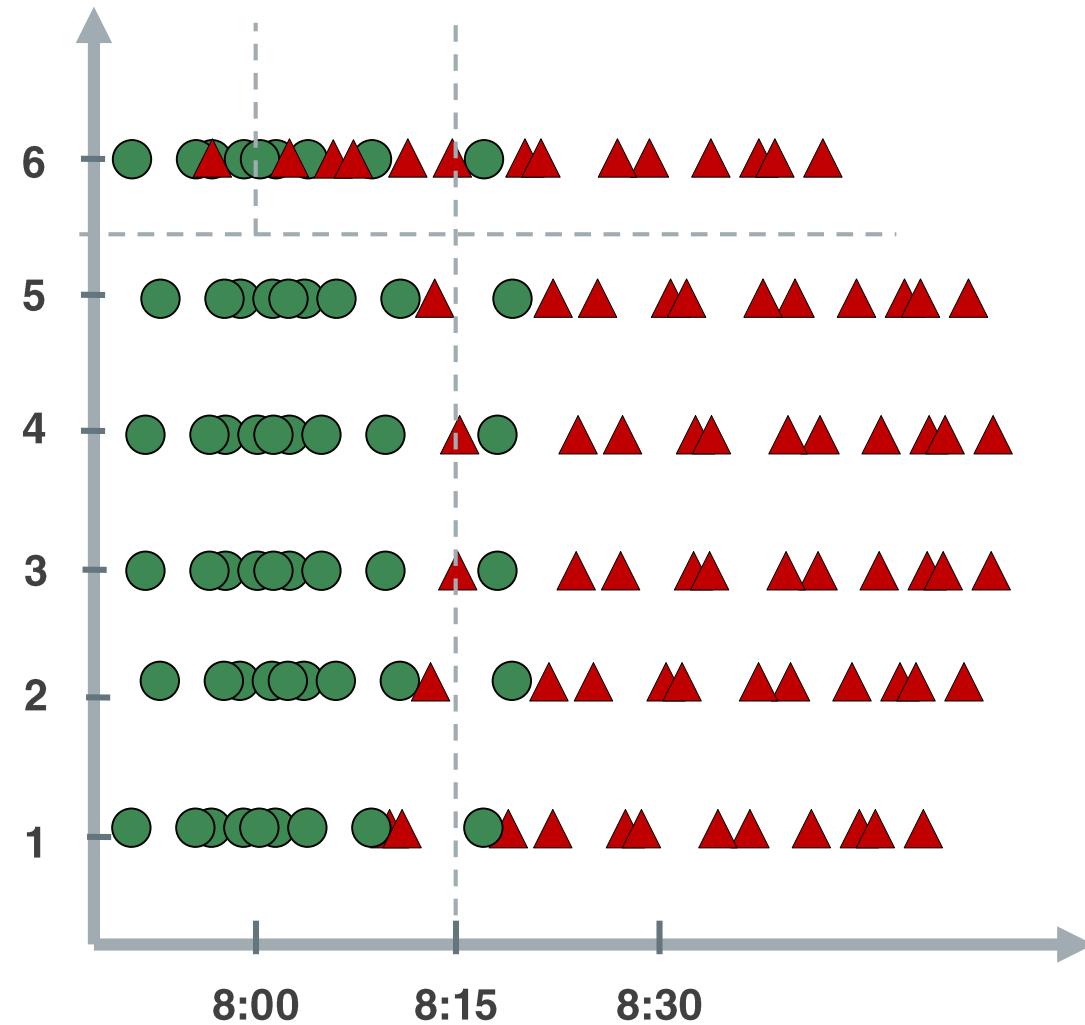


EJEMPLO

día de la semana

8:00 8:15 8:30

hora de salida



Salida < 8:15?

si no

día ∈ [lun-vie]?

si no

Salida < 8:00?

si no

puntual

puntual



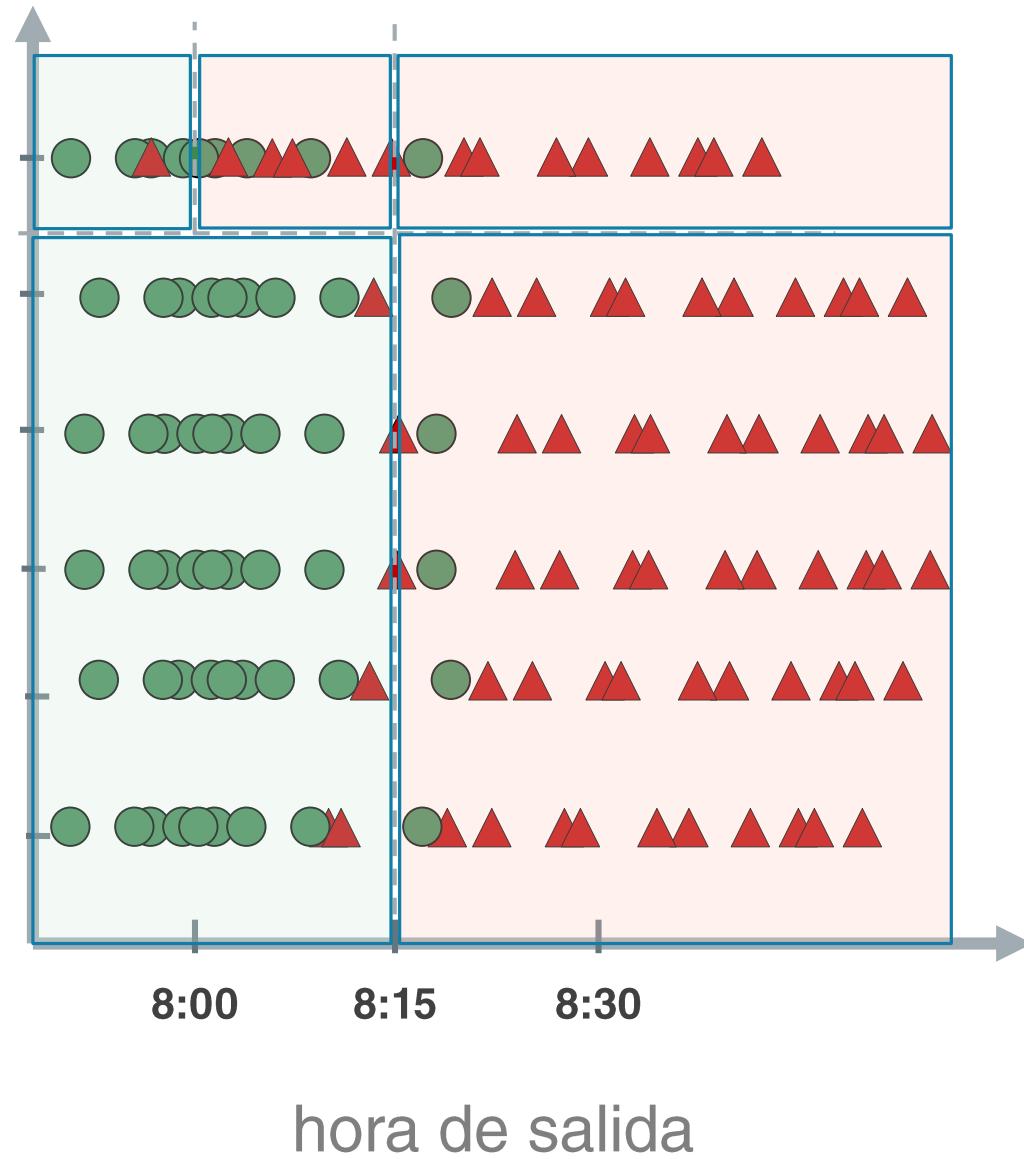
impuntual



impuntual

EJEMPLO

día de la semana



hora de salida

Salida < 8:15?



día ∈ [lun-vie]?



puntual

puntual



impuntual

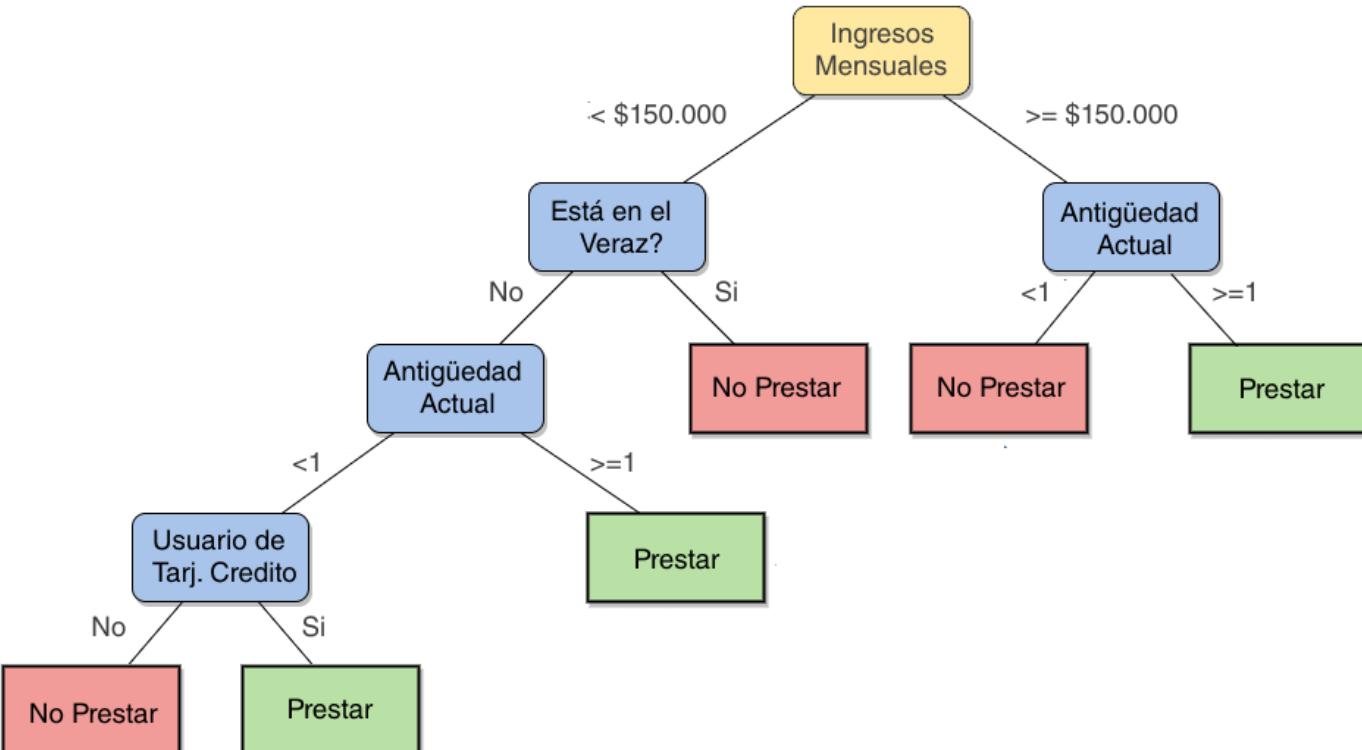
▲ impuntual

Salida < 8:00?



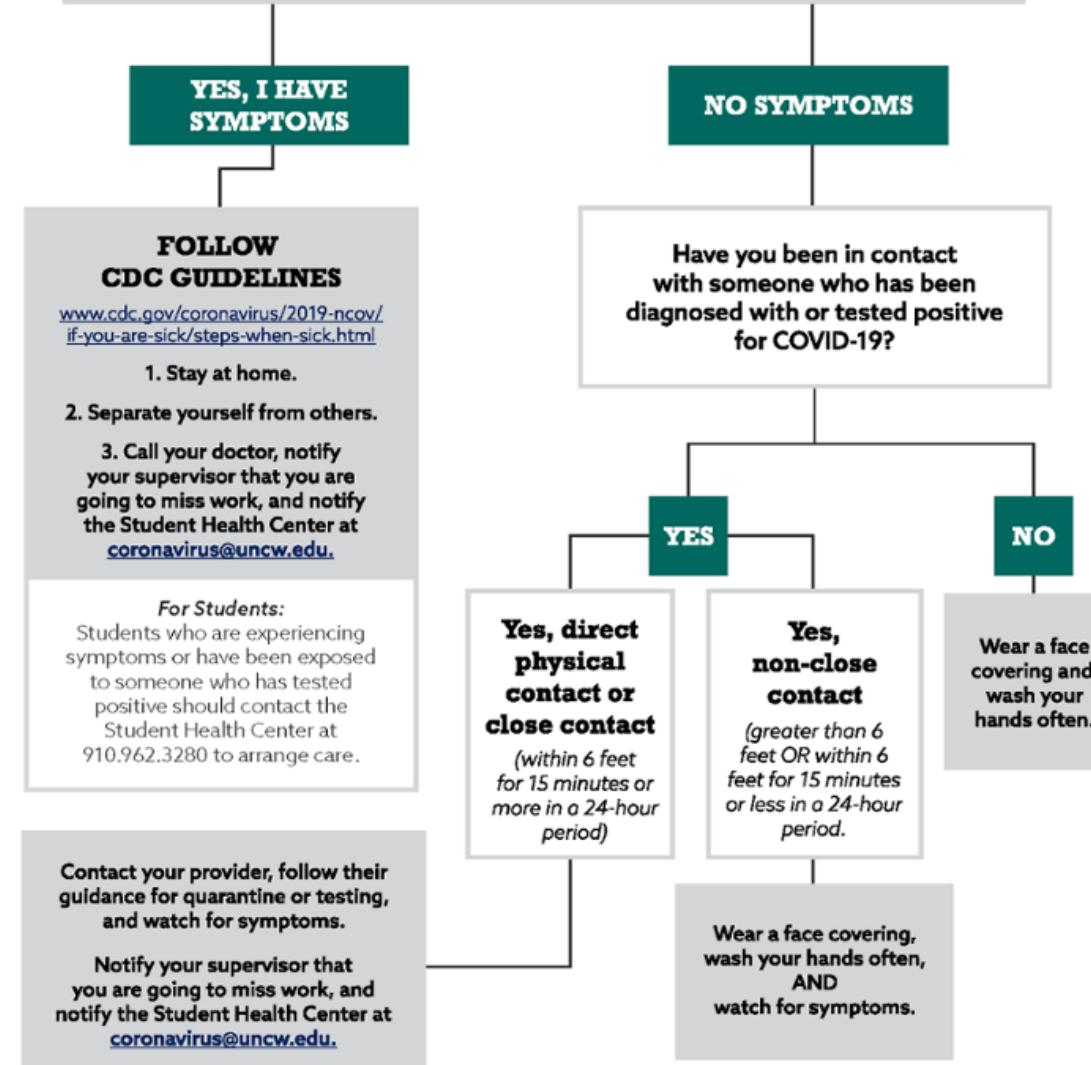
▲ impuntual

REPRESENTACIÓN



Are you experiencing COVID-19 symptoms? They include:

- » Fever and chills
- » Fatigue
- » Sore throat
- » Cough
- » Muscle or body aches
- » Congestion or runny nose
- » Shortness of breath
- » Headache
- » Nausea or vomiting
- » Difficulty breathing
- » New loss of taste or smell
- » Diarrhea

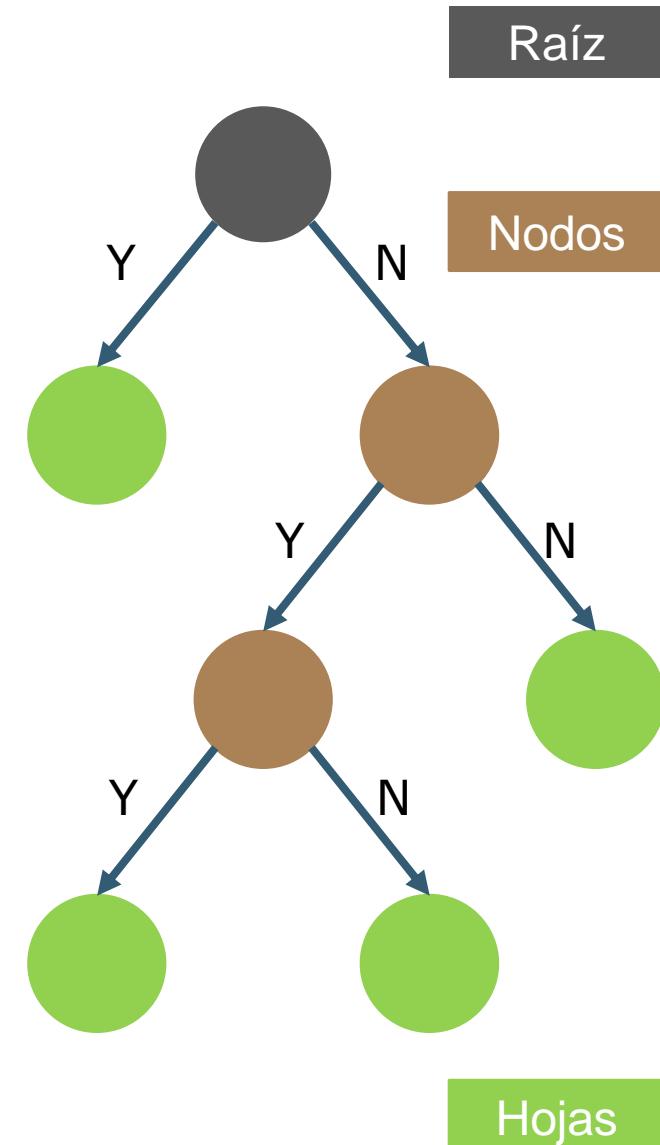


REPRESENTACIÓN

- El modelo aprendido se representa mediante árboles (grafos acíclicos dirigidos), generalmente binarios

- Tipos de nodos:

- **Raíz**: el primer nodo del árbol. Sólo emite arcos
- **Internos (nodos)**: donde se toman las decisiones de separación. Reciben y emiten arcos
- **Hojas**: es donde se realiza la asignación de clases. Reciben arcos, pero no los emiten

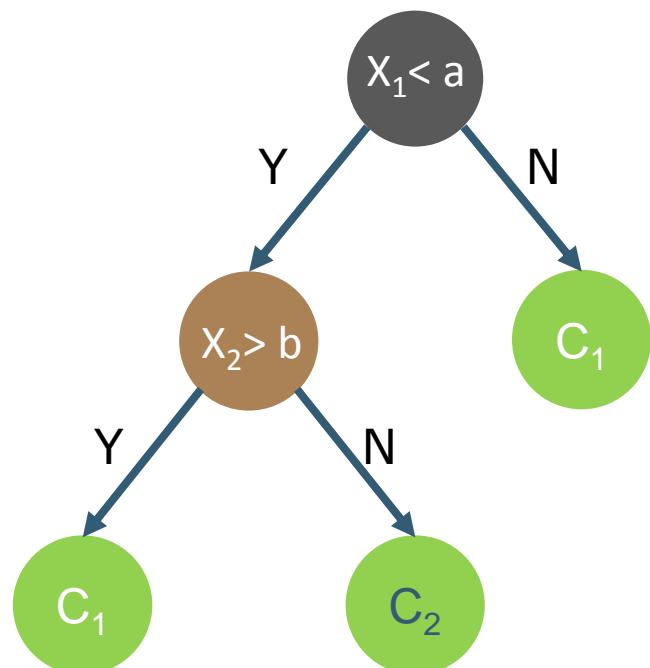


REPRESENTACIÓN

- Los árboles de decisión representan una **disyunción de conjunciones de restricciones** sobre los valores de los atributos de las instancias:
 - Cada camino desde la raíz hasta un nodo hoja, corresponde a una conjunción de pruebas de atributos ($\dots \wedge \dots \wedge \dots$)
 - El árbol corresponde a una disyunción de estas conjunciones:
$$(\dots \wedge \dots \wedge \dots) \vee (\dots \wedge \dots \wedge \dots) \vee (\dots \wedge \dots \wedge \dots) \vee \dots$$

REPRESENTACIÓN

Se puede traducir cada camino del árbol a reglas IF-THEN:



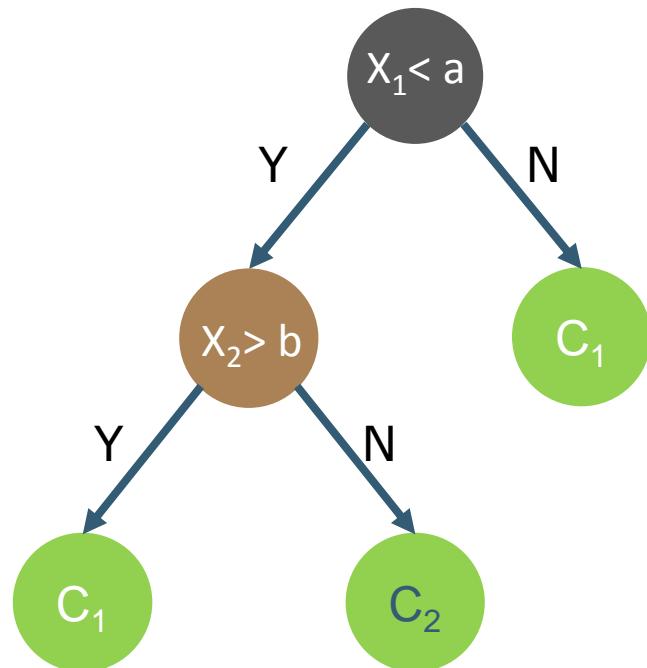
if ($X_1 < a$) & ($X_2 > b$) then (Clase = C_1)

if ($X_1 < a$) & $\neg(X_2 > b)$ then (Clase = C_2)

if $\neg(X_1 < a)$ then (Clase = C_1)

REPRESENTACIÓN

La clasificación de una nueva instancia usando un árbol entrenado se hace aplicando esa secuencia de reglas IF-THEN



$$d = (X_1 = (a-1), X_2 = (b+2))$$

if $(X_1 < a) \ \& \ (X_2 > b)$ then (Clase = C_1)

if $(X_1 < a) \ \& \ \neg(X_2 > b)$ then (Clase = C_2)

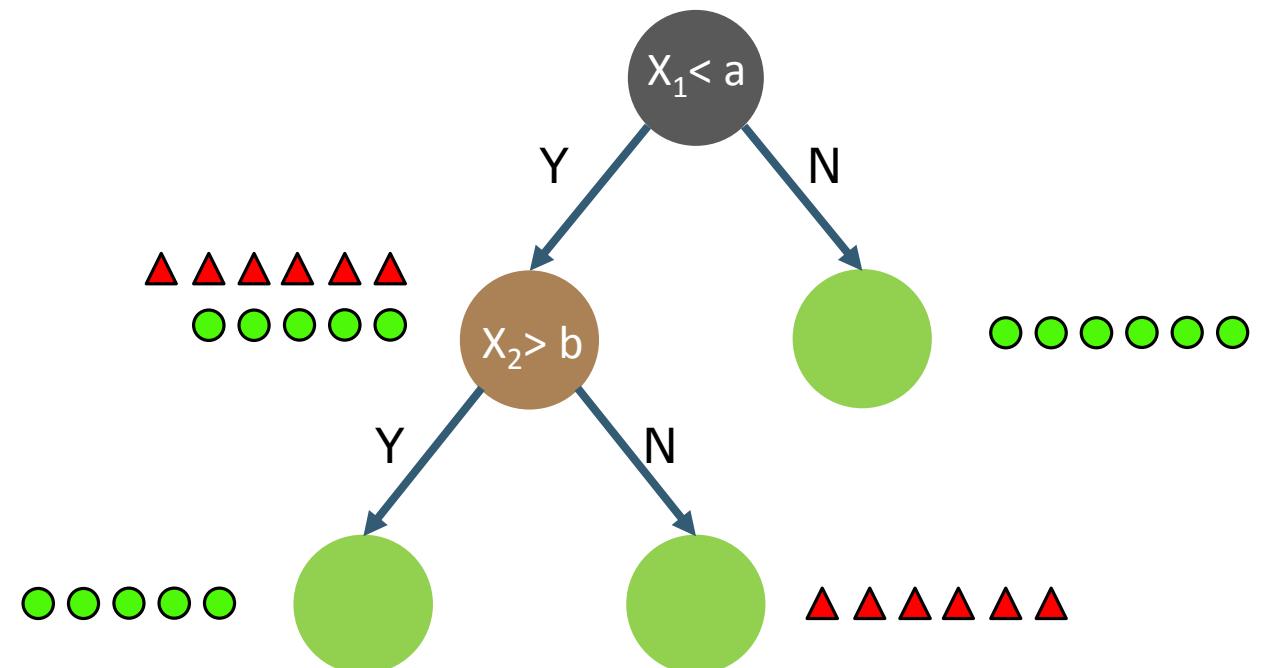
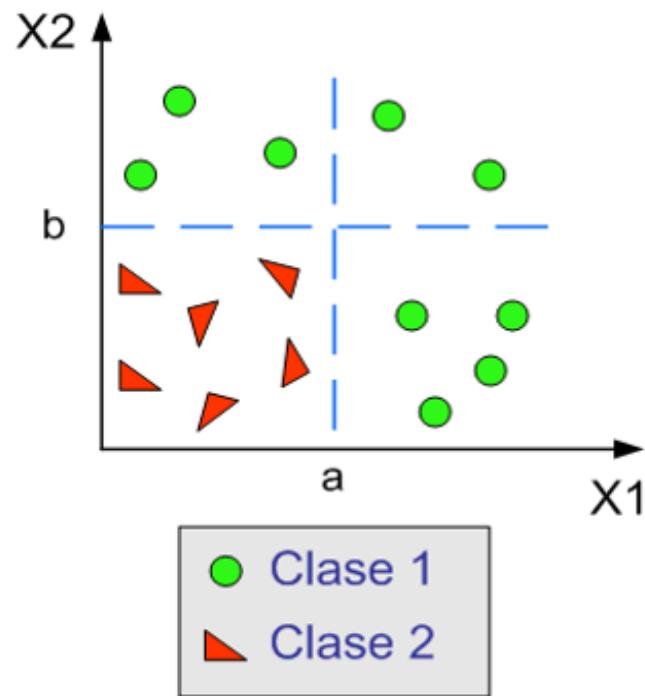
if $\neg(X_1 < a)$ then (Clase = C_1)

$$d \in C_1$$

CÓMO IR DE DATOS A ÁRBOLES? *(INDUCCIÓN)*

ALGORITMO ID3

- Secuencia de nodos de comparación
- Se examina un atributo diferente por vez
- Particionamiento recursivo: repetitivamente dividen los datos en subconjuntos cada vez más pequeños y más puros



ALGORITMO ID3

Inducción

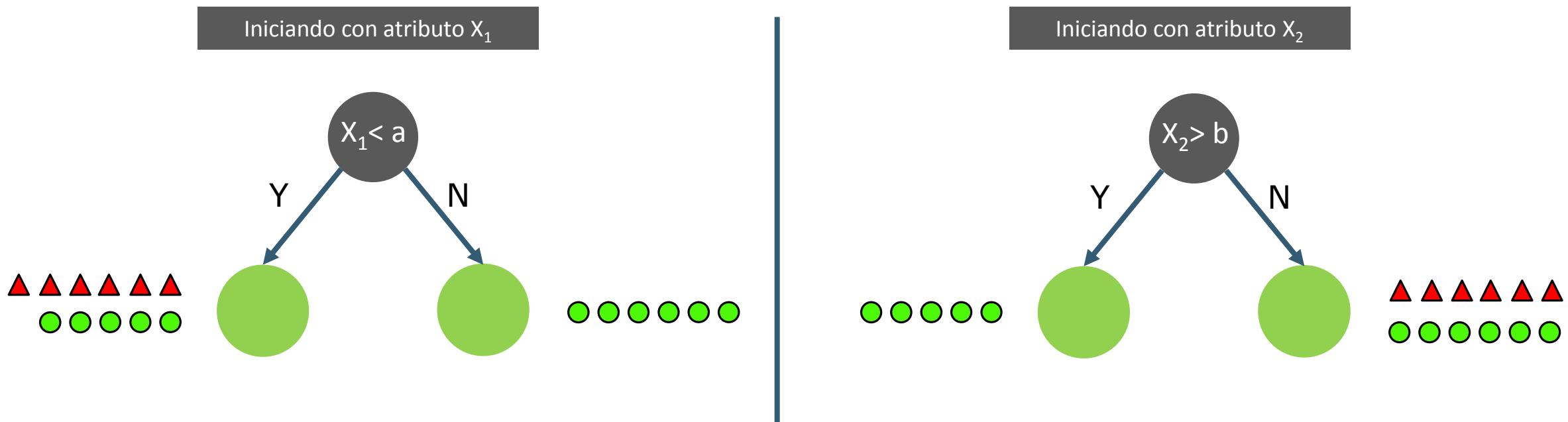
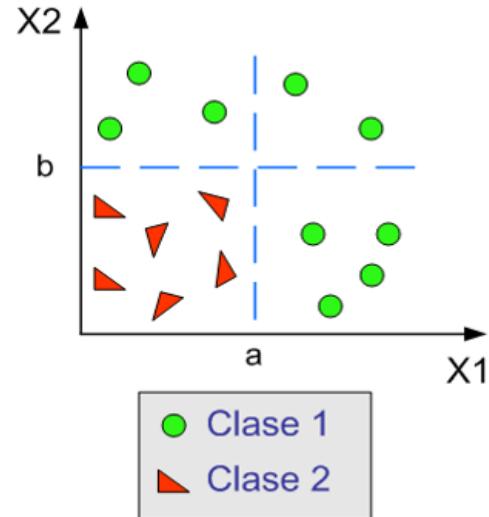
1. Crear un nodo raíz conteniendo todos los ejemplos
2. Evaluar diferentes particiones (atributos) y separar los ejemplos de entrenamiento en subconjuntos
3. Hacer un ranking de las particiones posibles y elegir la mejor
4. Para cada nodo de la partición obtenida repetir desde el paso 2, hasta que se satisfaga alguna condición de parada:
 - Todas las muestras para un nodo dado pertenecen a la misma clase
 - No restan atributos para seguir con las particiones
 - No quedan ejemplos

SELECCIÓN DE ATRIBUTOS

- Clave en el funcionamiento de los árboles: qué secuencia de atributos es la que conviene usar para el particionamiento
- Sesgo inductivo (Heurística): navaja de Occam. Preferir la solución más simple que resuelve el problema
- Aproximación "golosa" (*greedy*): en cada paso se elige el atributo que genera la “mejor partición” sin importarles lo complejo de seguir subdividiendo esas particiones más adelante
- Se busca trabajar con un “horizonte acotado” para evitar la explosión combinatoria. Esto genera susceptibilidad a óptimos locales

SELECCIÓN DE ATRIBUTOS

- “Mejor partición”: la que genera subconjuntos más puros



Intuitivamente conviene comenzar por X_1 porque separa 6 instancias puras y X_2 5

SELECCIÓN DE ATRIBUTOS

- Alternativas para estimar numéricamente la “pureza” de las particiones:
 - Ganancia de Información/entropía (ID3)
 - Relación de Ganancia de Información (C4.5)
 - Índice de Gini (CART)
 - χ^2
 - ...
- Todos basados en la relación del grado de impureza en el nodo padre (antes de la separación) respecto a la de los nodos hijos (resultante)

SELECCIÓN DE ATRIBUTOS

Ganancia de Información (ID3)

Se basa en conceptos de teoría de la información. Sea S el conjunto de entrenamiento con s_i casos de la clase C_i para $i = \{1, 2, \dots, m\}$ se define:

- Entropía o Información Total del Conjunto:

$$I(s_1, s_2, \dots, s_m) = - \sum_{i=1}^m \left(\frac{s_i}{S} \right) \log_2 \left(\frac{s_i}{S} \right)$$

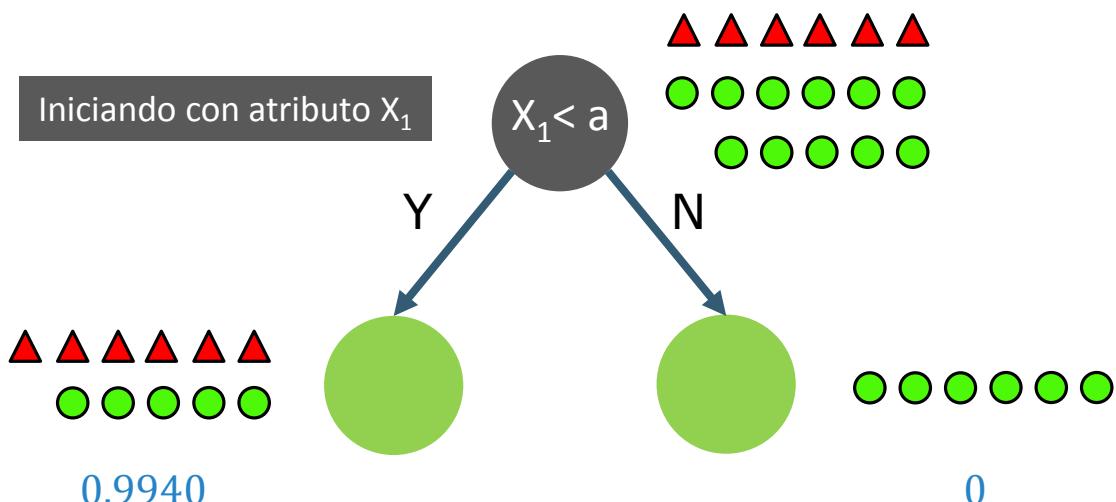
SELECCIÓN DE ATRIBUTOS

Ganancia de Información

- Calculemos la entropía para cada conjunto:

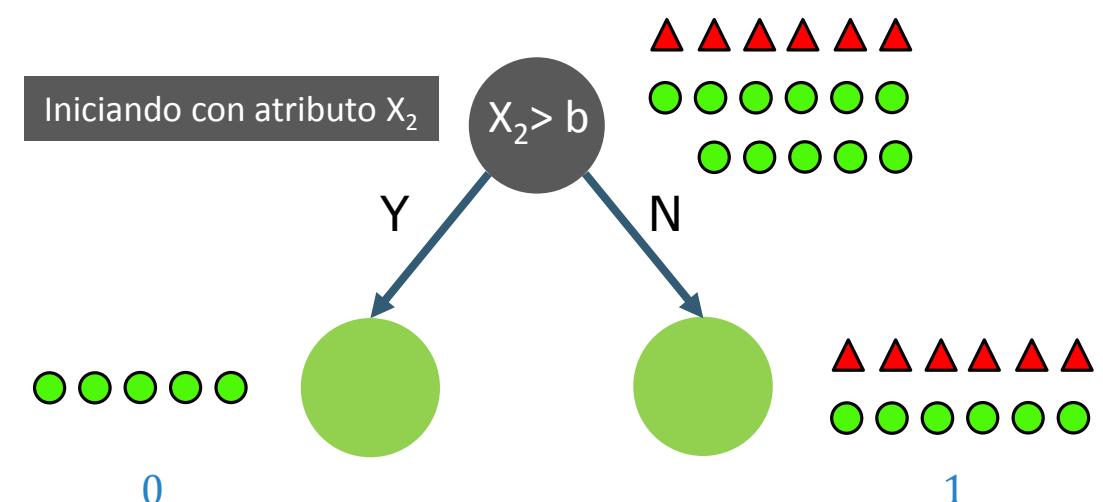
$$I(s_1, s_2) = -\left(\frac{s_1}{S}\right)\log_2\left(\frac{s_1}{S}\right) - \left(\frac{s_2}{S}\right)\log_2\left(\frac{s_2}{S}\right)$$

$$I_{total}(s_1, s_2) = -\left(\frac{11}{17}\right)\log_2\left(\frac{11}{17}\right) - \left(\frac{6}{17}\right)\log_2\left(\frac{6}{17}\right) = 0.9367$$



$$I_{x1}^1 = -\left(\frac{5}{11}\right)\log_2\left(\frac{5}{11}\right) - \left(\frac{6}{11}\right)\log_2\left(\frac{6}{11}\right) = 0.9940$$

$$I_{x1}^2 = -\left(\frac{6}{6}\right)\log_2\left(\frac{6}{6}\right) - \left(\frac{0}{6}\right)\log_2\left(\frac{0}{6}\right) = 0$$



$$I_{x2}^1 = -\left(\frac{5}{5}\right)\log_2\left(\frac{5}{5}\right) - \left(\frac{0}{5}\right)\log_2\left(\frac{0}{5}\right) = 0$$

$$I_{x2}^2 = -\left(\frac{6}{12}\right)\log_2\left(\frac{6}{12}\right) - \left(\frac{6}{12}\right)\log_2\left(\frac{6}{12}\right) = 1$$

SELECCIÓN DE ATRIBUTOS

Ganancia de Información (ID3)

- Entropía Residual del atributo A con valores $\{a_1, a_2, \dots, a_v\}$:

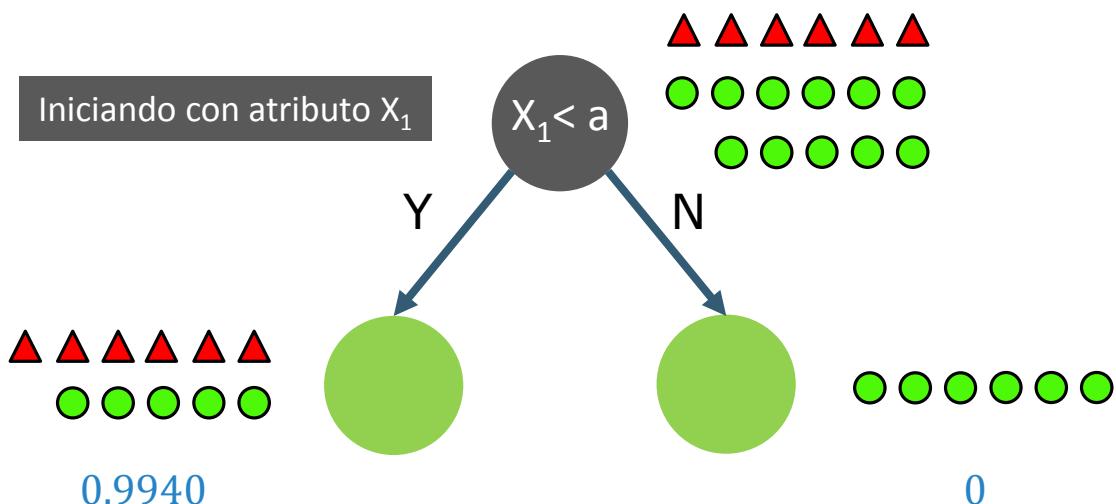
$$I_{RES}(A) = \sum_{j=1}^v p(a_i)I(s_{1j}, s_{2j}, \dots, s_{mj})$$

Es la incertidumbre residual después de haber obtenido información sobre el atributo A

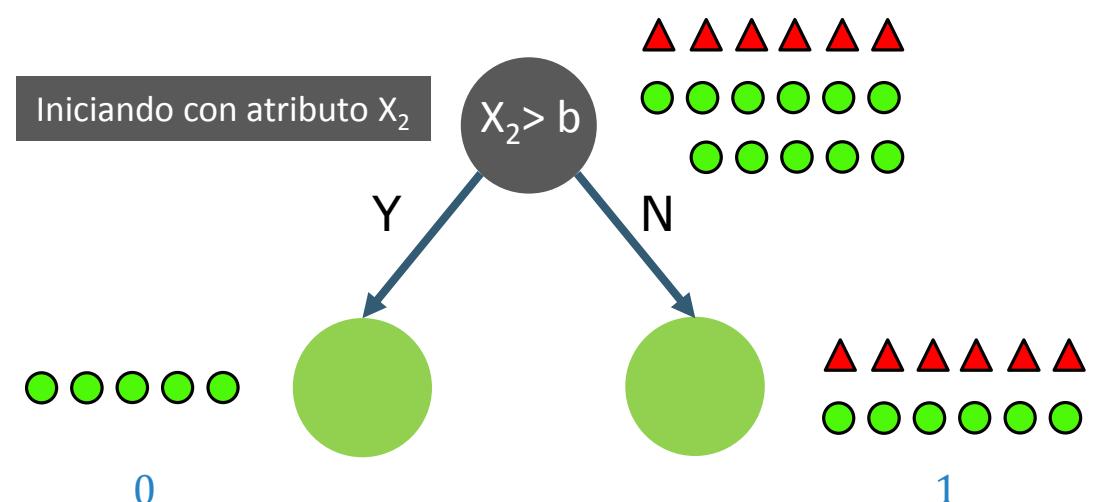
SELECCIÓN DE ATRIBUTOS

Ganancia de Información

- Calculemos la entropía residual para cada atributo:



$$I_{Res}(x_1) = \left(\frac{11}{17}\right)0.994 + \left(\frac{6}{17}\right)0 = 0.6432$$



$$I_{Res}(x_2) = \left(\frac{5}{17}\right)0 + \left(\frac{12}{17}\right)1 = 0.7059$$

SELECCIÓN DE ATRIBUTOS

Ganancia de Información (ID3)

- Ganancia de información al partir el conjunto de ejemplos usando el atributo A:

$$Gain(A) = I(s_1, s_2, \dots, s_m) - I_{RES}(A)$$

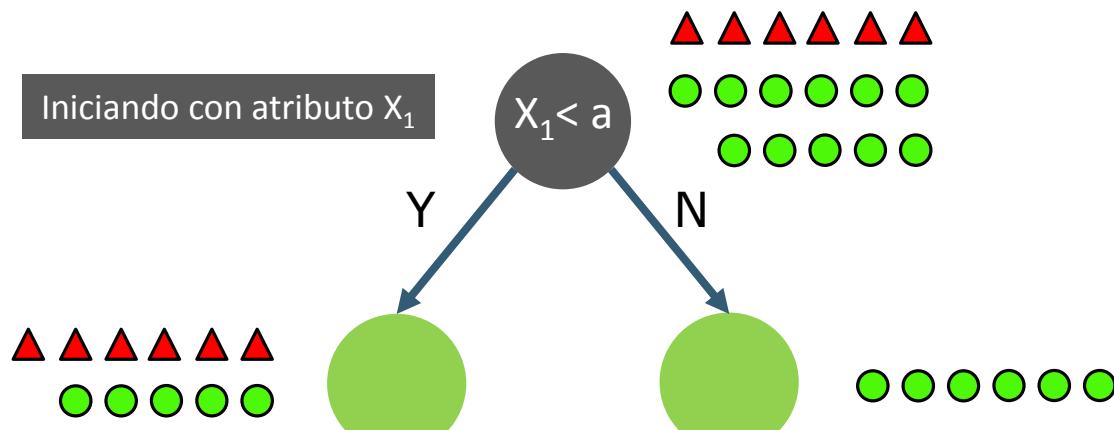
- Se elige el atributo que garantiza la mayor ganancia de información (heurística: preferir árboles poco profundos)
- Se repite el proceso con los subconjuntos restantes

SELECCIÓN DE ATRIBUTOS

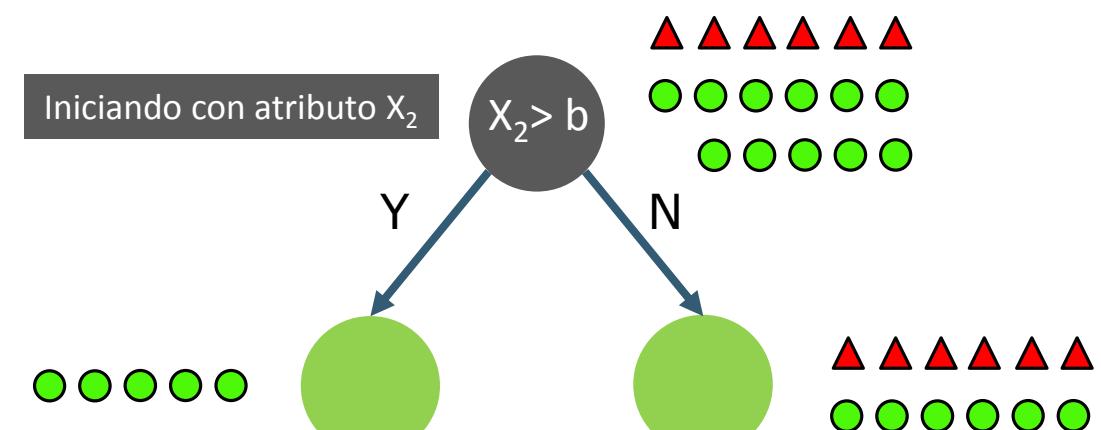
Ganancia de Información

- Calculemos la ganancia de información para cada atributo:

$$I_{total}(s_1, s_2) = -\left(\frac{11}{17}\right)\log_2\left(\frac{11}{17}\right) - \left(\frac{6}{17}\right)\log_2\left(\frac{6}{17}\right) = 0.9367$$



$$Gain(x_1) = 0.9367 - 0.6432 = 0.2935$$

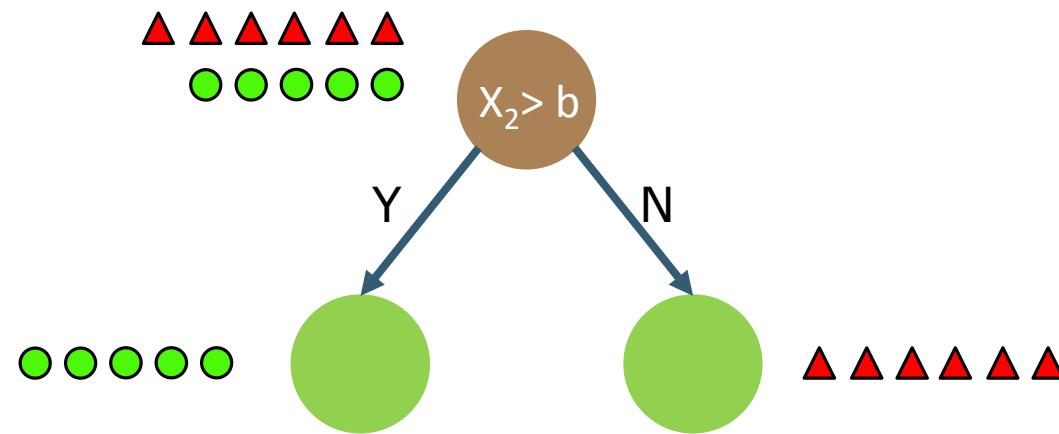


$$Gain(x_2) = 0.9367 - 0.7059 = 0.2308$$

Conviene comenzar por X_1 porque genera una ganancia de información mayor

SELECCIÓN DE ATRIBUTOS

- El proceso de inducción del árbol se repite con los nodos impuros, como si fuera un nuevo problema:



- Si quedan atributos para seguir con la partición, se los elige con el criterio de ganancia de información hasta llegar a una de las condiciones de finalización

EJEMPLO CLASIFICADOR BASADO EN ÁRBOLES DE DECISIÓN

CLASIFICADOR BASADO EN ID3

Ejemplo: clasificación cáncer de próstata

Edad	Antecedentes Familiares	Volumen Próstata (mm ³)	Valor PSA	Cáncer de Próstata
<45	N	< 60	< 4	N
<45	N	< 60	< 4	N
[45,70]	N	> 60	[4,10]	P
>70	N	> 60	> 10	P
>70	S	< 60	[4,10]	N
>70	S	< 60	>10	P
[45,70]	S	> 60	< 4	N
<45	N	< 60	[4,10]	N
<45	S	< 60	< 4	N
>70	S	< 60	< 4	N
<45	S	> 60	[4,10]	P
[45,70]	N	> 60	>10	P
[45,70]	S	< 60	[4,10]	P
>70	N	> 60	> 10	P

SELECCIÓN DE ATRIBUTOS

Ganancia de Información

- Esta medida impone un sesgo de preferencia hacia los atributos que tienen mayor número de valores diferentes
- Ejemplo típico: si el conjunto de datos tiene el DNI como atributo, podrá separar muy bien cada individuo, pero no conviene usarlo porque no va a generalizar a datos diferentes
- Se propusieron alternativas para contrarrestar ese problema:
 - Relación de ganancia de información
 - Índice Gini

SELECCIÓN DE ATRIBUTOS

Relación de Ganancia de Información

- Ross Quinlan, propuso esta versión actualizada de su algoritmo ID3, que llamó C4.5 (equivalente a J48 implementación Java de código abierto)
- Busca resolver el problema de sesgo normalizando la ganancia de A por la información de partición (*SplitInfo*), la entropía de los valores del atributo A:

$$GainRatio(S, A) = \frac{Gain(S, A)}{SplitInfo(S, A)}$$

$$SplitInfo(S, A) = - \sum_{v \in Valores(A)} P_v \log_2(P_v)$$

SELECCIÓN DE ATRIBUTOS

Relación de Ganancia de Información

- Para el ejemplo de cáncer de próstata:

$$SplitInfo(Edad) = -\frac{5}{14} \log_2 \left(\frac{5}{14} \right) - \frac{4}{14} \log_2 \left(\frac{4}{14} \right) - \frac{5}{14} \log_2 \left(\frac{5}{14} \right) = 1.5774$$

$$SplitInfo(Antecedentes) = -\frac{7}{14} \log_2 \left(\frac{7}{14} \right) - \frac{7}{14} \log_2 \left(\frac{7}{14} \right) = 1$$

$$SplitInfo(VProst) = -\frac{8}{14} \log_2 \left(\frac{8}{14} \right) - \frac{6}{14} \log_2 \left(\frac{6}{14} \right) = 0.9852$$

$$SplitInfo(PCA) = -\frac{5}{14} \log_2 \left(\frac{5}{14} \right) - \frac{5}{14} \log_2 \left(\frac{5}{14} \right) - \frac{4}{14} \log_2 \left(\frac{4}{14} \right) = 1.5774$$

SELECCIÓN DE ATRIBUTOS

Índice de Gini

- Empleado en CART (Breiman et al, 1984)
- Sea S el conjunto de entrenamiento con k clases diferentes y p_j la probabilidad de la clase j en S , el índice de Gini de S está dado por:

$$Gini(S) = 1 - \sum_{j=1}^k p_j^2$$

- Para cada atributo, este procedimiento considera una división binaria
- Si S se divide en base al atributo A en dos subconjuntos S_1 y S_2 :

$$Gini(S, A) = \frac{|S_1|}{|S|} Gini(S_1) + \frac{|S_2|}{|S|} Gini(S_2)$$

SELECCIÓN DE ATRIBUTOS

Índice de Gini

- Se asume el índice Gini como una medida de impureza
- Se necesita saber en cuánto reduce la impureza cada atributo A:

$$\Delta Gini (S, A) = Gini (S) - Gini (S, A)$$

- El atributo A que proporciona el menor $Gini (S, A)$ o la mayor reducción de impureza, se elige para dividir el nodo
- Para el ejemplo anterior:

$$Gini (CancerProstata) = 1 - \sum_{j=1}^k p_j^2 = 1 - \left(\frac{7}{14}\right)^2 - \left(\frac{7}{14}\right)^2 = 0.5$$

SELECCIÓN DE ATRIBUTOS

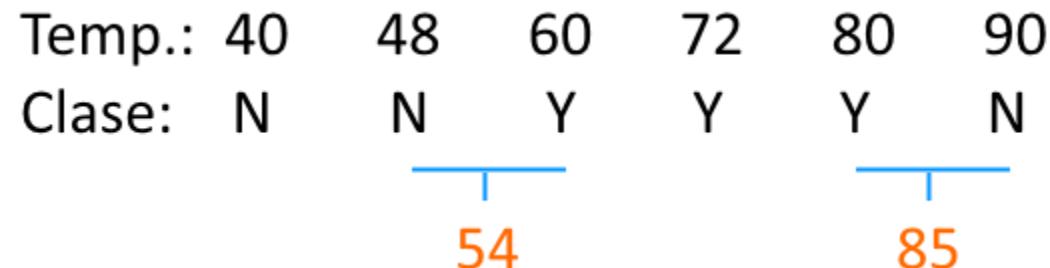
Índice de Gini para atributos no binarios categóricos

- Se binarizan las categorías mediante reagrupaciones
- Ejemplo Edad = {<45, [45,70], >70}
- Se consideran las divisiones de estas tres categorías en dos grupos:
 - {<45, [45,70]} y {>70}
 - {<45} y {[45,70]}, >70}
 - {<45, >70} y {[45,70]}
- Se calcula el índice de Gini para cada partición alternativa
- Se elige la partición con menor índice Gini (mayor reducción de impureza)

SELECCIÓN DE ATRIBUTOS

Índice de Gini para atributos numéricos

- Se debe elegir un umbral para hacer binario el atributo valuado continuo A
- Se ordenan los valores que adopta A en orden creciente
- Se identifican ejemplos adyacentes que difieren en sus etiquetas de clase



- Se asumen cada uno como posibles umbrales de binarización
- Se calcula el índice de Gini para cada partición alternativa
- Se elige la partición con menor índice Gini (mayor reducción de impureza)

SELECCIÓN DE ATRIBUTOS

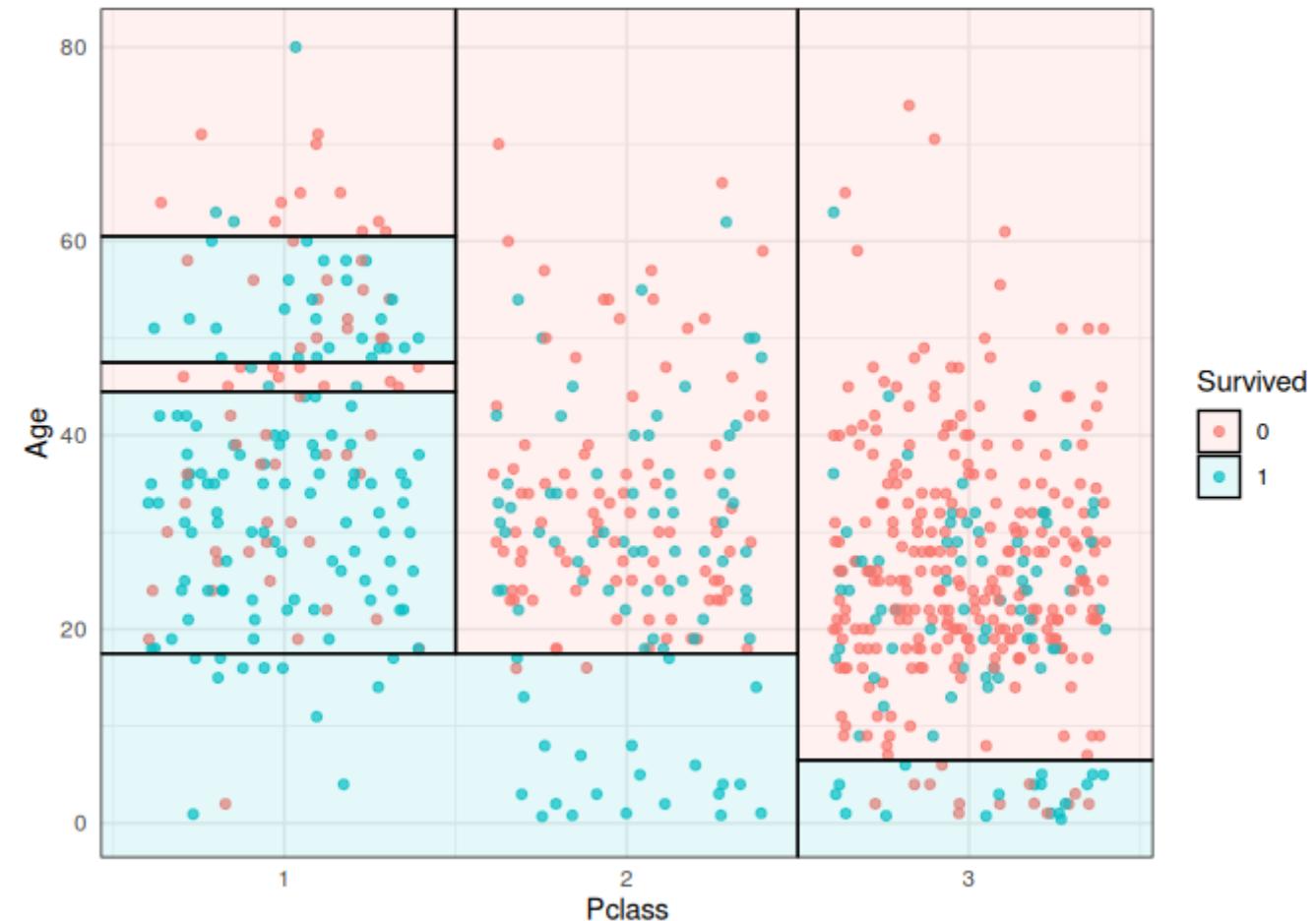
Comparación de métodos

- Ganancia de información
 - sesgada hacia atributos multivaluados
- Relación de ganancia
 - preferencia por atributos con valores desbalanceados (una partición es mucho más pequeña que las demás)
- Índice de Gini
 - sesgo hacia atributos multivaluados
 - dificultades cuando el número de clases es grande
 - tiende a favorecer los atributos que dan lugar a particiones de igual tamaño y pureza en ambas particiones

LÍMITES DE DECISIÓN

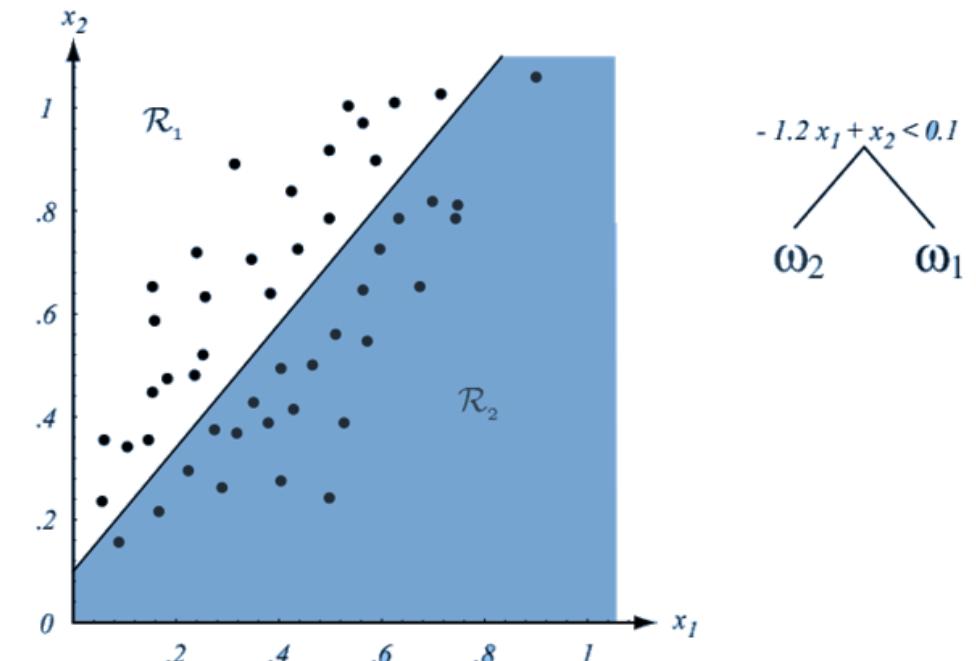
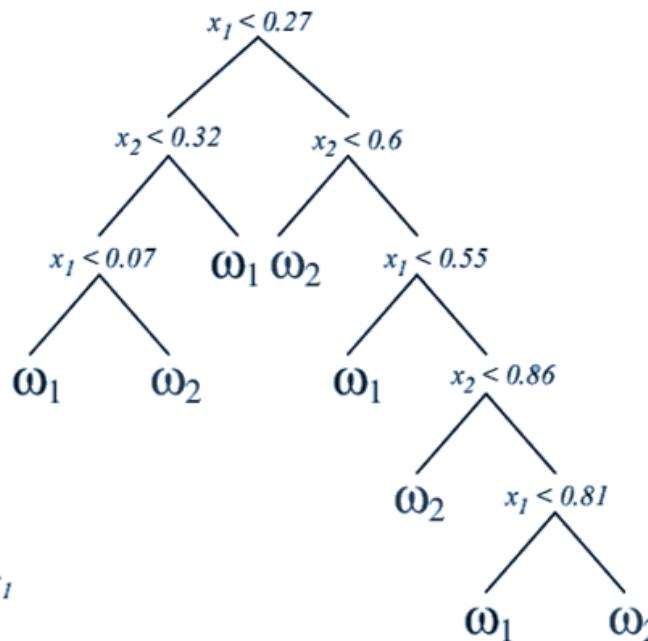
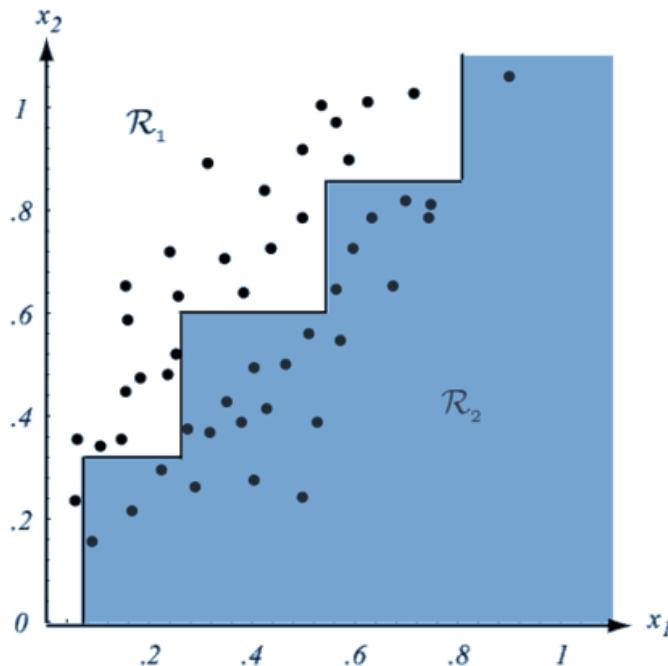
Los AD dividen el espacio de características en **hiper-rectángulos paralelos a los ejes** (regiones de decisión)

- Una región puede ser etiquetada en función de la clase (mayoritaria) y/o de la distribución de clases de la región
- Límite de decisión: línea fronteriza entre dos regiones vecinas de diferentes clases



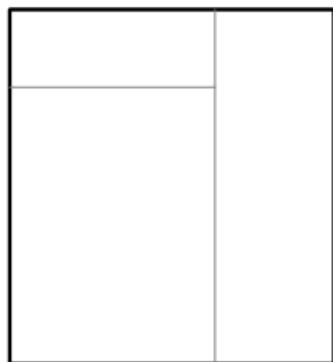
LIMITACIONES DE LOS ÁRBOLES DE DECISIÓN

- No resultan eficientes en muchos tipos de problemas
- Una alternativa, modificar los atributos (**AD oblicuos**, pueden considerar más de un atributo en las particiones)

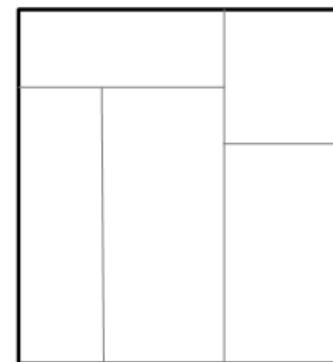


SOBREAJUSTE EN ÁRBOLES DE DECISIÓN

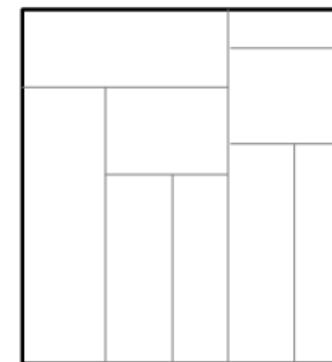
- El árbol inducido puede sobre-ajustar los datos de entrenamiento:
 - Un muy buen rendimiento en las muestras de entrenamiento (ya vistas)
 - Poca precisión en las muestras no vistas
- Demasiadas ramas, algunas pueden reflejar anomalías debidas a que las reglas aprendidas se deben al comportamiento de ruido o outliers



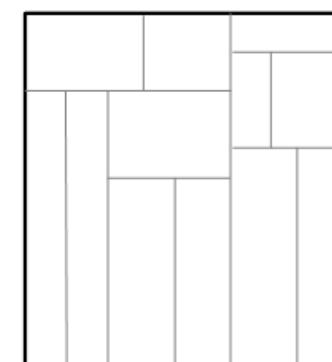
3 Nodos



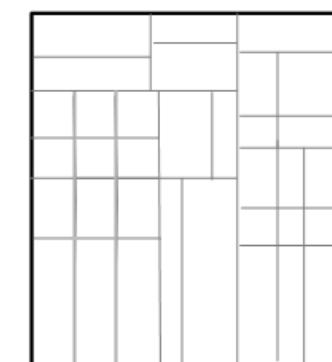
5 Nodos



9 Nodos



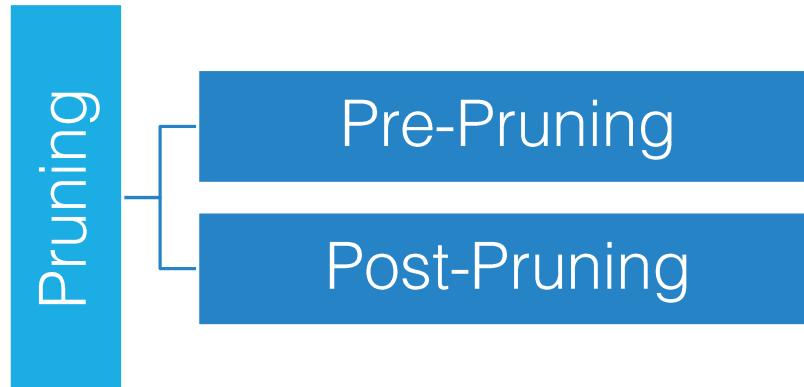
12 Nodos



22 Nodos

PODA (PRUNING)

- Alternativa para reducir sobreajuste → podar el árbol de decisión



Detener la inducción del árbol cuando la información deja de ser fiable
A partir de un árbol de decisión completo, descartar las partes no fiables

- En la práctica se prefiere el post-pruning, ya que el pre-pruning puede "detenerse antes de tiempo"

PRE-PRUNING

Estrategias

- Basado en la prueba de significación estadística
 - Detener el crecimiento del árbol cuando para un nodo no haya asociación estadísticamente significativa entre cualquier atributo y la clase de un nodo particular
 - Prueba más popular: prueba de chi-cuadrado
 - ID3 usa la prueba de chi-cuadrado además de la ganancia de información
 - El procedimiento de ganancia de información sólo permite seleccionar los atributos estadísticamente significativos
- Basados en un umbral de número mínimo de instancias por nodo
 - C4.5 utiliza una estrategia más sencilla, pero la combina con la post-pruning
 - Cada nodo por encima de una hoja debe tener al menos dos sucesores que contengan al menos m ejemplos (por defecto usa $m=2$)

POST-PRUNING

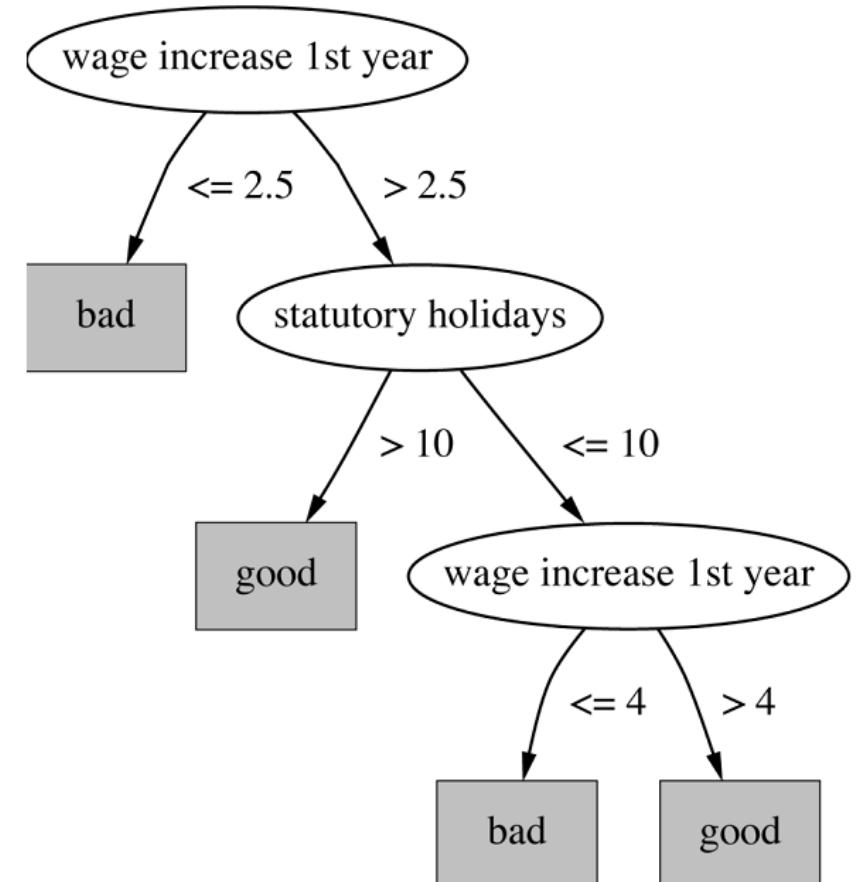
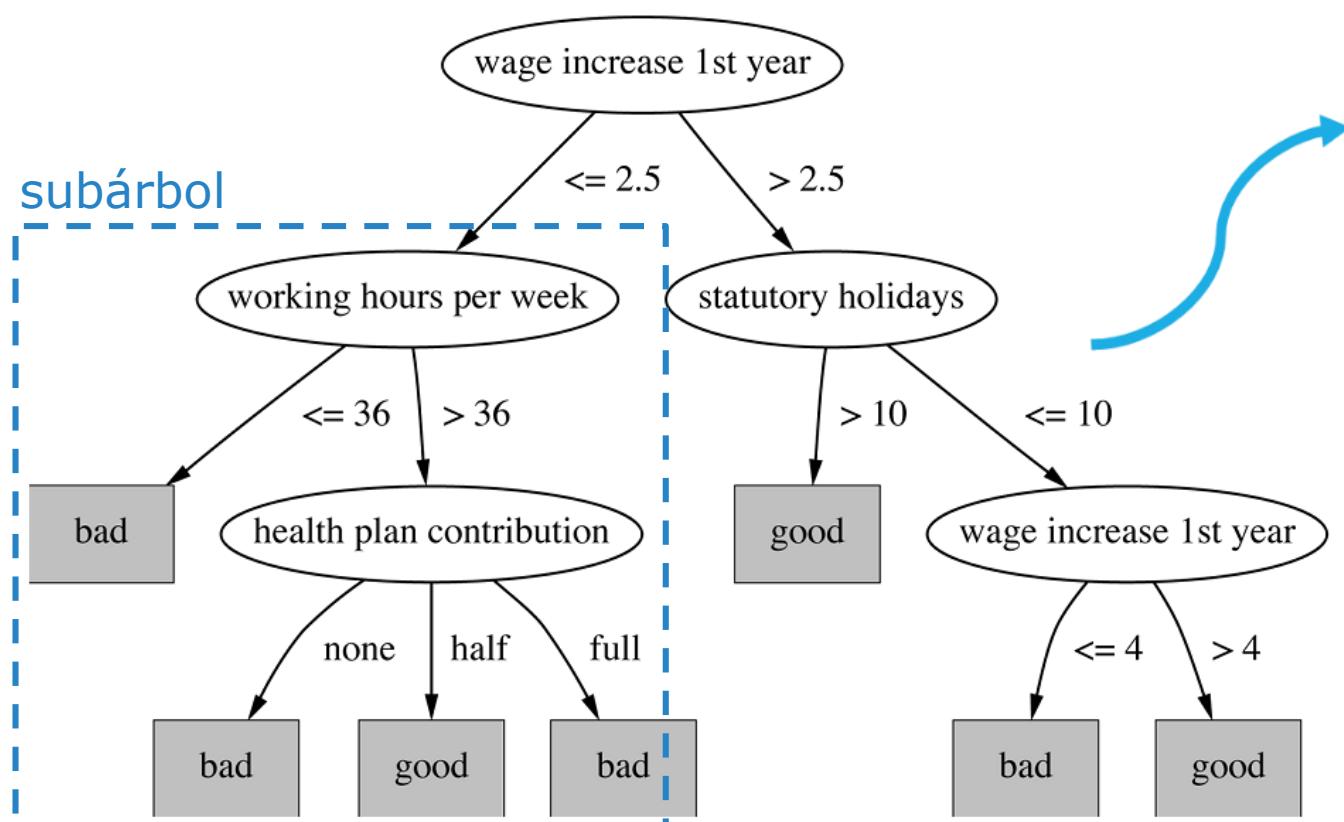
- Inducir el árbol de decisión que clasifique todos los datos de entrenamiento
 - El árbol completo muestra todas las interacciones de los atributos
 - Algunos subárboles podrían deberse al azar
- Podar los nodos del árbol de decisión de forma ascendente (*bottom-up*)



POST-PRUNING

Sustitución de subárboles:

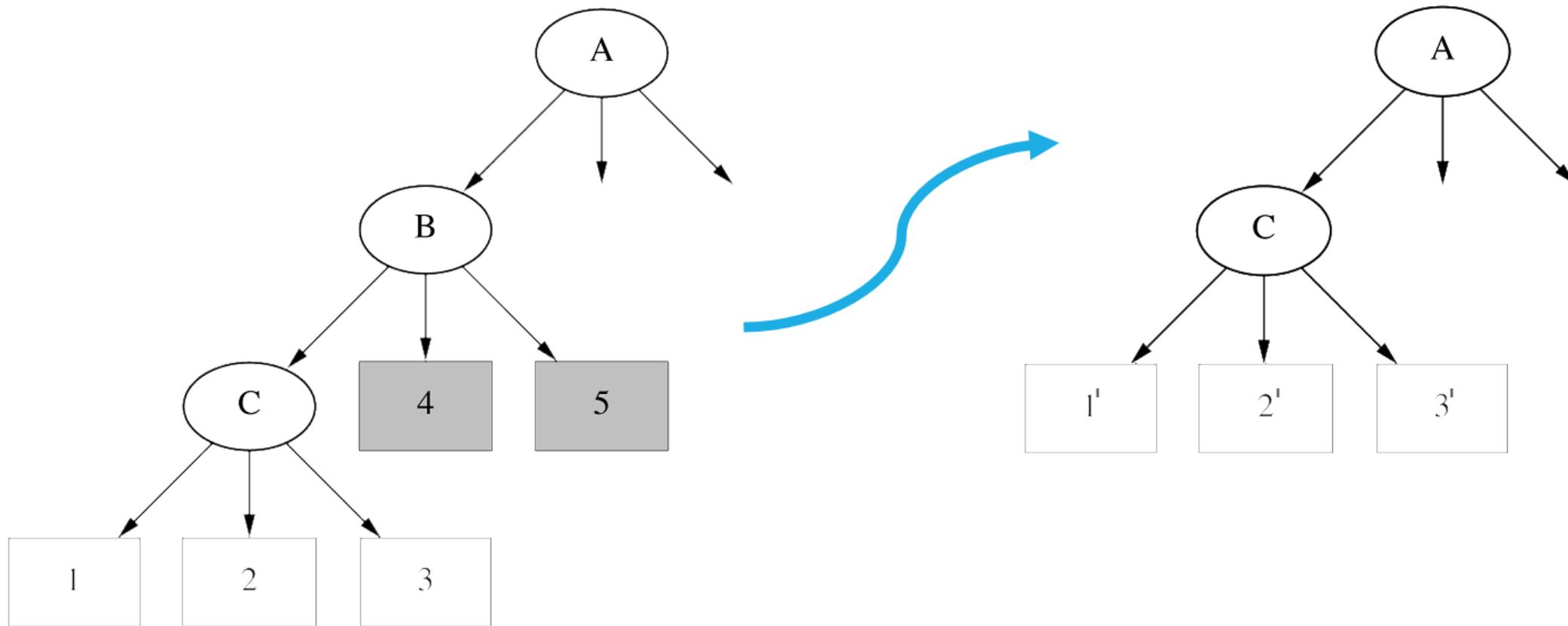
- Seleccionar un subárbol
- Reemplazarlo por una hoja



POST-PRUNING

Elevación de subárboles

- Borrar el nodo B
- Redistribuir instancias de las hojas 4 y 5 en C



ESTRATEGIAS DE PRUNING

- Basada en reducción de errores
 - Podar sólo si no aumenta el error estimado
 - El error en los datos de entrenamiento NO es un estimador útil (podría llevar a que no se podes nada)
 - El error se evalúa en un conjunto de datos independiente (conjunto de validación)
- Basada en complejidad
 - Se añade una penalización por complejidad en la medida de desempeño
 - Ej. longitud mínima de la descripción (MDL)

COMPARACIÓN DE ALGORITMOS

Características	ID3	C4.5 (J48)	CART	CHAID
Atributos admitidos	Categóricos	Categóricos + Continuos	Categóricos + Continuos	Categóricos
Selección de atributos	Ganancia de Información (Entropía)	Relación de Ganancia de Información (Split Info)	Índice GINI	Chi ²
Pruning	Ninguno	Pre-pruning	Post-pruning	Pre-pruning
Valores faltantes	-	+	+	+
Outliers	-	-	+	-

ÁRBOLES DE DECISIÓN

Fortalezas

- Fácil de entender y visualizar
- Robusto al ruido y admiten datos faltantes
- Admiten variables categóricas y continuas (no hay distancias)
- Eficientes en tiempo de clasificación: $O(\log (N))$ (N : patrones)

Debilidades

- Algoritmo goloso (mínimos locales)
- Particionamientos univariados limitan el tipo de árboles generados
- Susceptibles a la maldición de dimensionalidad
- Complejidad de construcción $O(dN[\log N]^2)$ (d : Nº de atributos)

LIMITACIONES

- A pesar de ser muy populares, cada vez se emplean más en la forma de ensambles 
- Son la base para algoritmos mucho más potentes como:
 - Random Forest
 - Gradient Boosting Machine
 - XGBoost (*eXtreme Gradient Boost*)
- Éstos admiten implementaciones paralelizables y uso de GPUs para acelerar el aprendizaje

BIBLIOGRAFÍA

- Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc.
- Han, J.; Kamber, M.; & Pei, J. (2011). *Data Mining Concepts and Techniques*, 3Ed. Mogan Kaufman
- Cios, K.J.; Pedrycz, W.; and Swiniarski, R. (2007) *Data Mining Methods for Knowledge Discovery*. Springer
- Tan, P.; Steinbach, M.; Karpatne, A.; Kumar, V. (2018) *Introduction to Data Mining*, 2 Ed. Pearson
- Mitchell, T. (1997) *Machine Learning*. McGraw-Hill