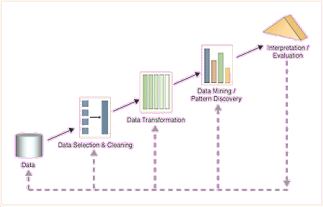
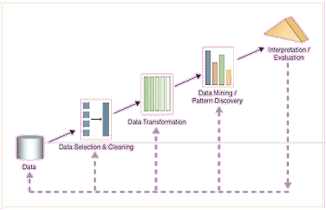
Universidad del Valle

Clasificación



Clasificación vs. Predicción

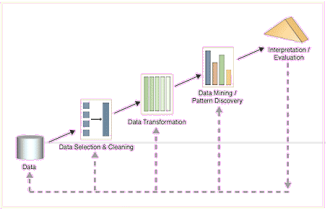
- Clasificación
  - Predicción de **etiquetas categóricas**
  - Clasificar los datos (construir un modelo) basándose en un **conjunto de datos de entrenamiento** y los valores de un atributo de clasificación, y luego utilizar el modelo para clasificar nuevos datos



Clasificación vs. Predicción

- Predicción
  - Modelos (funciones) para variables con valores continuos, i.e., predicción de valores desconocidos

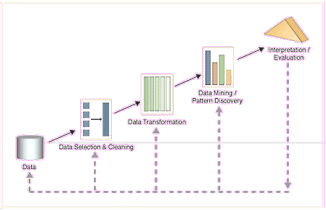
- Aplicaciones típicas
  - Concesión de créditos
  - Campañas de marketing
  - Diagnósticos médicos



Clasificación

- Muy similar a la experiencia de aprendizaje humana
  - Utilización de la observación para formar un modelo

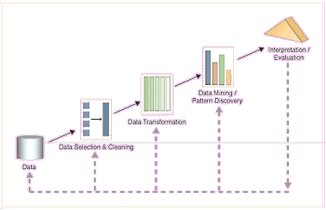
- Analizar un conjunto de datos para determinar las características de los mismos (creación de un modelo)



Clasificación

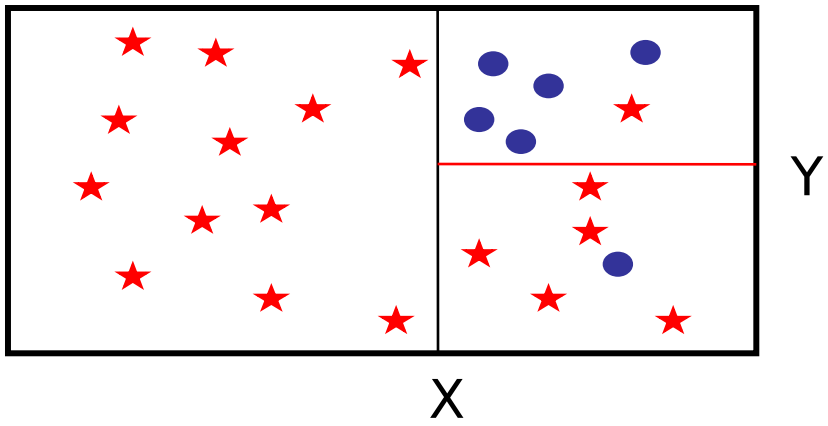
- Aprendizaje supervisado
  - El modelo se forma a partir de datos clasificados correctamente de antemano

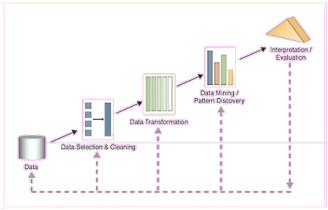
- Los modelos se desarrollan en dos fases
  - Entrenamiento
  - Prueba



Clasificación
Objetivo

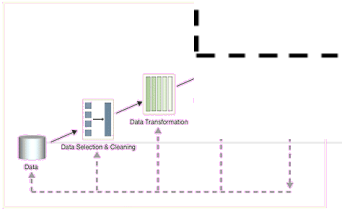
Obtener modelos que discrimine las instancias de entrada en diferentes clases de equivalencia por medio de los valores de diferentes atributos.





Clasificación
Requisitos

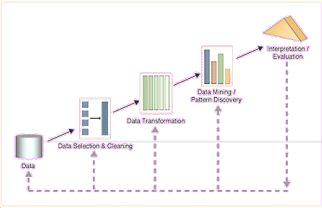
- Suministrar el **atributo decisión** o clase (label)
  - El conjunto de valores de este atributo debe ser finito y no excesivamente grande
- Suministrar los **atributos condición**
- Podría requerir datos que no sean numéricos pero existen algoritmos que tratan con datos numéricos
- Número máximo de precondiciones



Clasificación
Entrada de los algoritmos

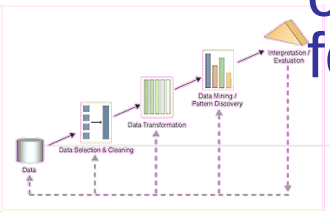
- **Atributos condición**: Atributos usados para describir por medio del proceso de inducción las clases.

- **Atributos decisión o label**: Atributos usados para construir las clase en los métodos supervisados (una clase por cada valor o combinación de valores de dichos atributos).

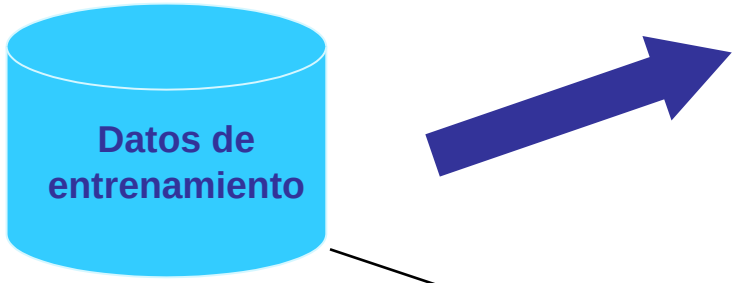


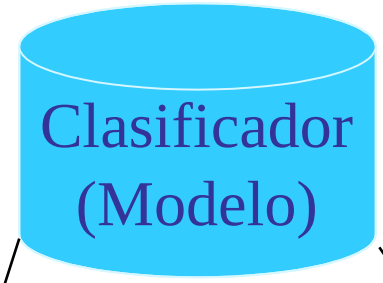
Clasificación

- **Construcción del modelo**
  - Describir un conjunto de datos con base en una característica
    - Cada tupla pertenece a una clase predefinida determinada por el atributo de decisión
    - Se utiliza el conjunto de datos de entrenamiento
    - El modelo se representa a través de reglas de clasificación, árboles de decisión o mediante formulas matemáticas



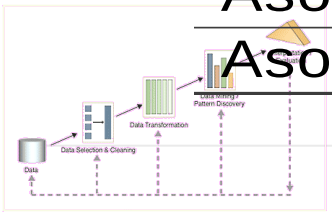
Clasificación
Construcción del modelo





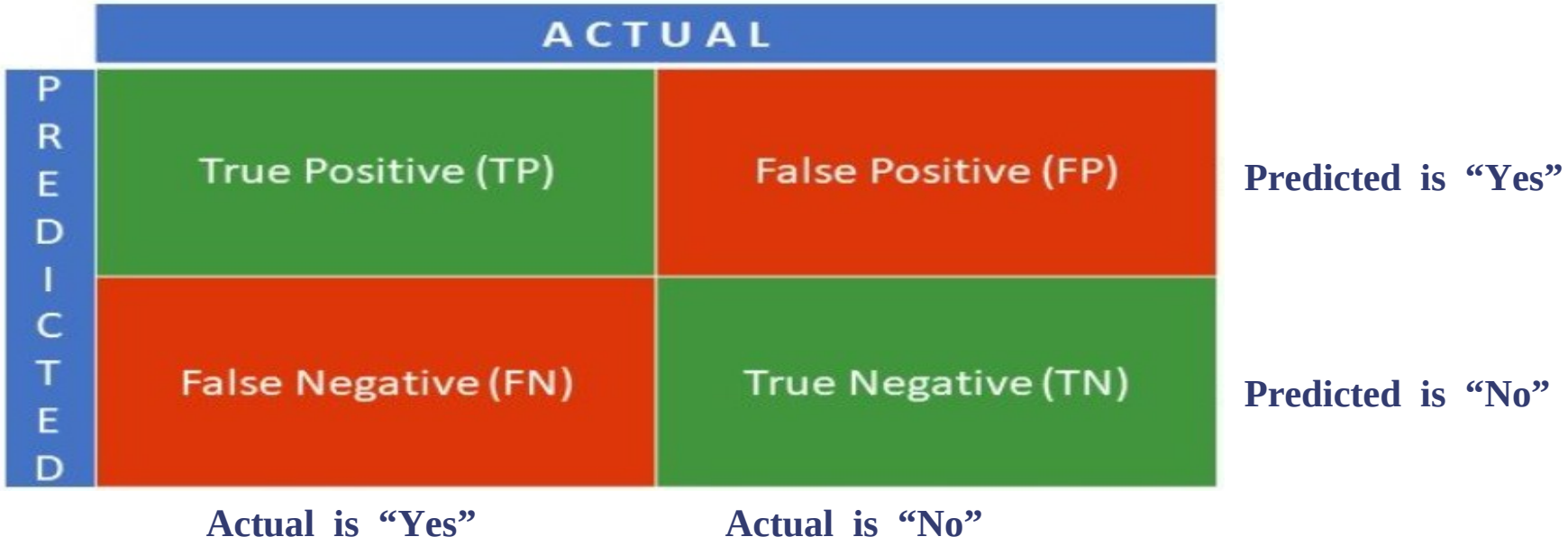
Tipo	Años	Fijo
Asociado	3	no
Asociado	7	si
Titular	2	si
Asociado	7	si
Asociado	6	no
Asociado	3	no

IF tipo = 'Titular'
OR años > 6
THEN fijo = 'si'



Matriz de Confusión

Se usa para representar la forma como se clasifican las instancias.



Precision = TP/ (TP + FP)

Accuracy = TP+ TN / (TP+FP+TN + FN)

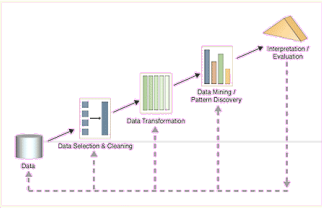
Specificity = TN/ (TN + FP)

Recall= TP/ (TP + FN) *(Sencibility)*

Clasificación
Técnicas

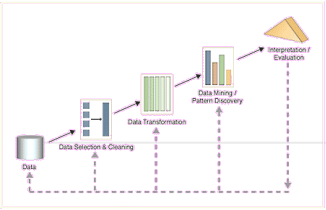
- **Árboles de decisión**
  - C4.5 y sus variantes, ID3
  - Muy eficientes en tiempo de proceso
  - Resultados intuitivos

- **Redes neuronales**
  - El resultado es una arquitectura de nodos con pesos
  - Muy robustas



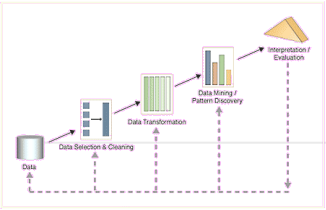
Clasificación
Técnicas

- Técnicas simbólicas: árboles de decisión
  - Muy eficientes en tiempo de proceso
  - Resultados intuitivos
  - Particiones lineales
  - Algunos presentan problemas con variables continuas



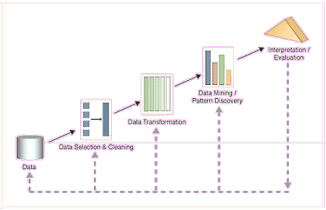
Clasificación
Técnicas

- Redes neuronales
  - Sólo entrada numérica
  - Más robusto
  - Difícil de entender la salida



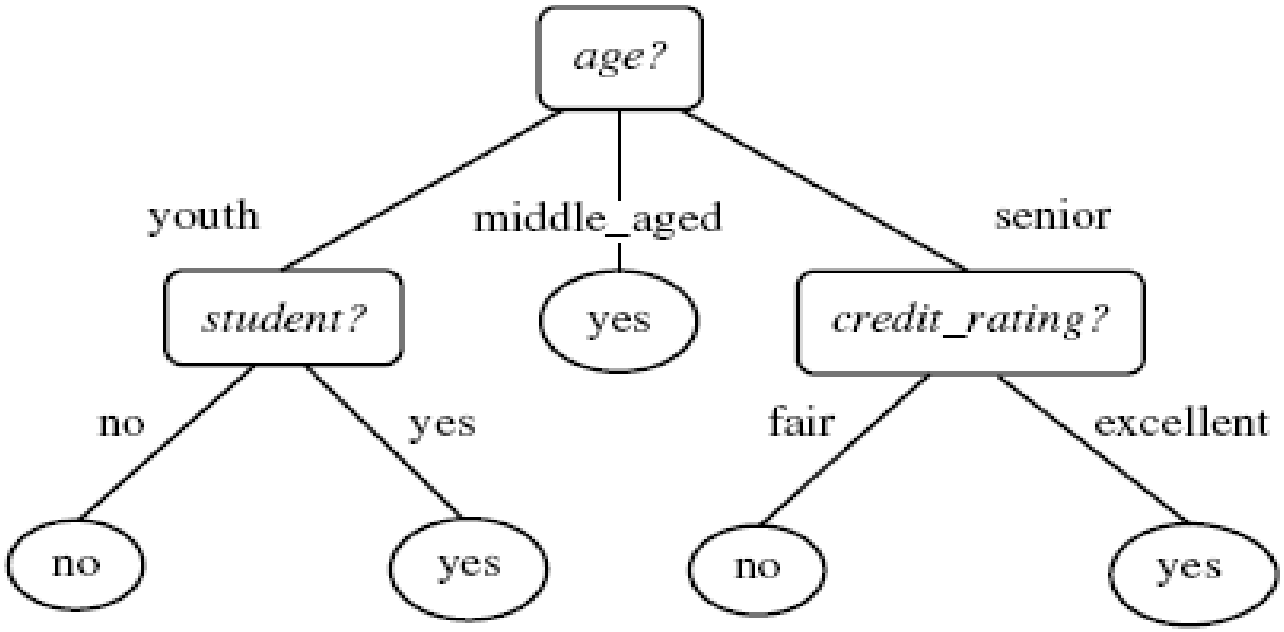
Clasificación
Árboles de decisión

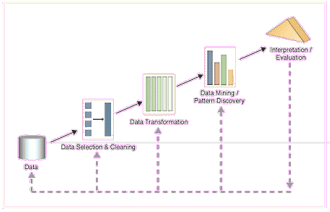
- Árboles de decisión
  - La representación es en forma de árbol
  - Los **nodos** representan la verificación de una condición sobre un atributo
  - Las **ramas** representan el valor de la condición comprobada en el nodo del cual derivan
  - Los **nodos hoja** representan las etiquetas de clase



Clasificación
Técnicas

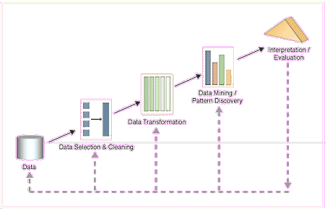
Árboles de decisión



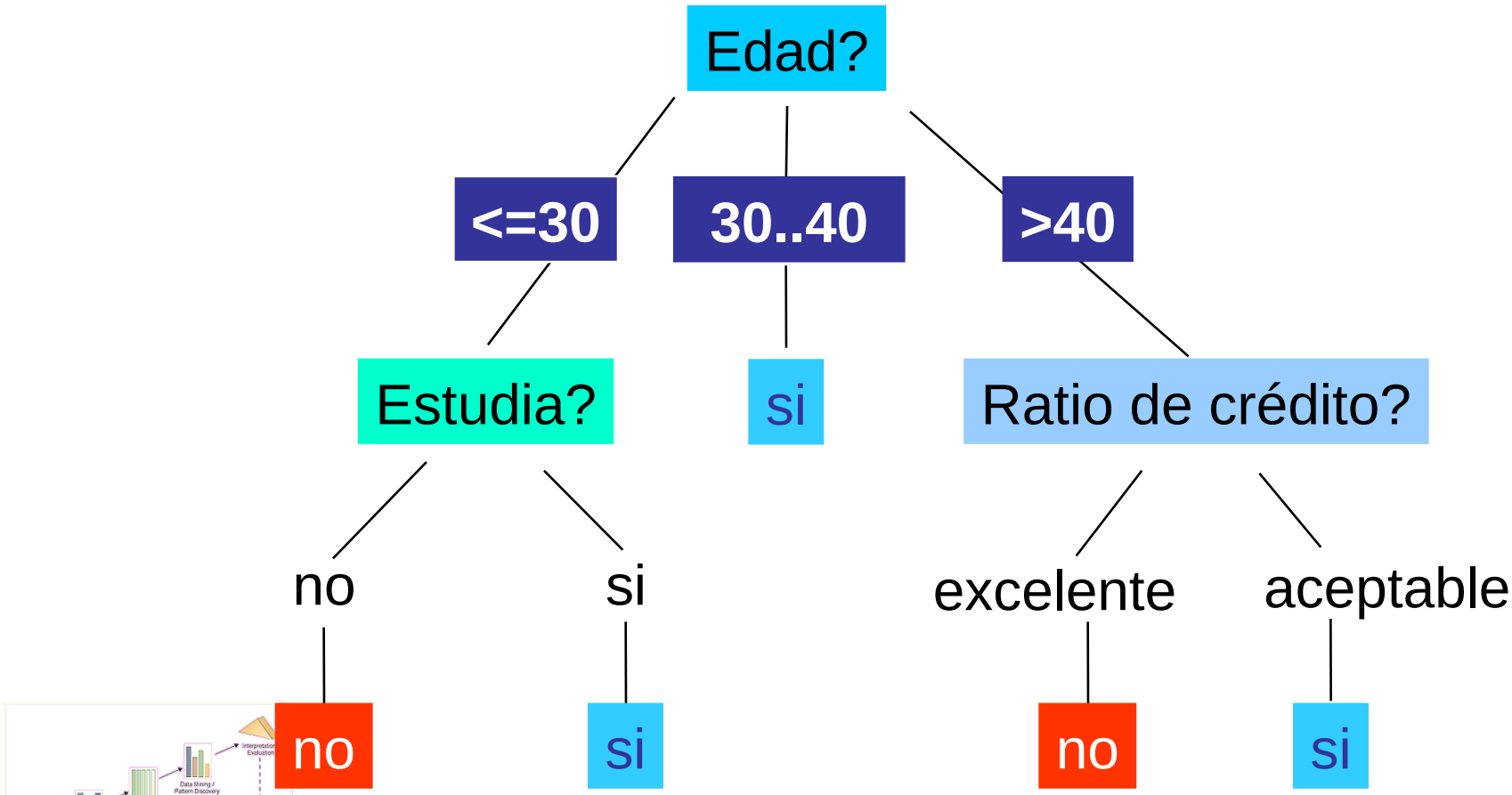


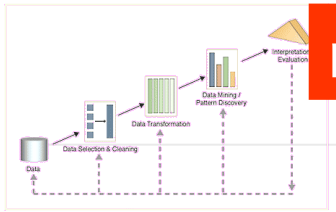
Ejemplo
Datos de entrenamiento

Edad	Estudia	Ratio de crédito	Compra
alta	no	aceptable	no
alta	no	excelente	no
alta	no	aceptable	si
media	no	aceptable	si
baja	si	aceptable	si
baja	si	excelente	no
baja	si	excelente	si
media	no	aceptable	no
baja	si	aceptable	si
media	si	aceptable	si
media	si	excelente	si
media	no	excelente	si
alta	si	aceptable	si
media	no	excelente	no



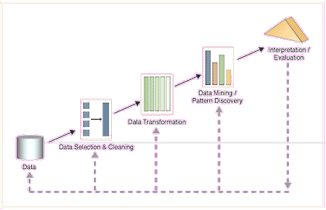
Ejemplo
Árbol de decisión





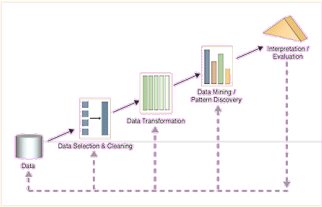
Árbol de decisión: algoritmo

- **Algoritmo básico (voraz)**
  - El árbol se construye de forma **top-down** recursiva utilizando divide y vencerás
  - Al principio, todas las tuplas se encuentran en la raíz
  - Los atributos deben ser categóricos, si son valores continuos hay que **discretizarlos** previamente
  - Las tuplas se van dividiendo recursivamente en base al atributo seleccionado
  - Los atributos de condición se seleccionan en base a heurísticas o mediante medidas estadísticas, por ejemplo, ganancia de información



Árbol de decisión: algoritmo

- **Condiciones de terminación**
  - Todas las muestras en un nodo pertenecen a la misma clase
  - No hay más atributos para futuras particiones. Se puede utilizar votación para clasificar el nodo hoja
  - No quedan más ejemplos

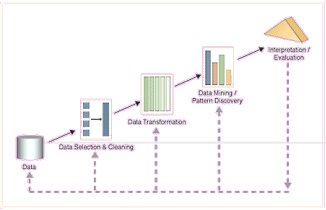


Ganancia de información (ID3/C4.5/C5.0)

- Seleccionar el atributo con mayor ganancia de información
- Si hay dos clases, P y N
  - Sea el conjunto de ejemplo S que contiene p elementos de la clase P y n elementos de las clase N
  - La cantidad de información, que se necesita para decidir si una muestra cualquiera de S pertenece a P o a N se define como:

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Entropía



Entropía de la información (ID3/C4.5/C5.0)

Entropía de un atributo

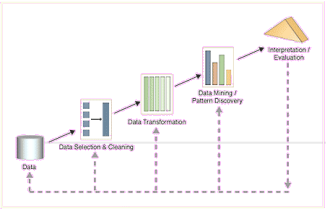
$$Entropy_{min} = -1 \cdot log_2(1) = 0$$

$$Entropy_{max} = -0.5 \cdot log_2(0.5) - 0.5 \cdot log_2(0.5) = 1$$

La división óptima se elige por la **característica** con **menor entropía**.

Máximo: Cuando la probabilidad de las dos clases es la misma

Mínimo (Nodo puro): cuando la entropía tiene su valor mínimo, que es 0:



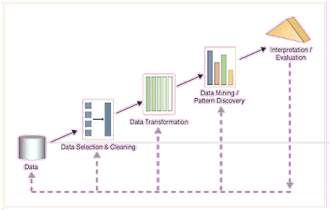
Ganancia de información en árboles de decisión

- Si se utiliza un atributo A, un conjunto S se dividirá en conjuntos {S1, S2 , …, Sv}
  - Si S_i contiene p_i ejemplos de P y n_i ejemplos de N, la entropía, o la información necesaria para clasificar objetos en todos los subárboles S_i es

$$E(A)=\sum_{i=1}^{v}\frac{p_i+n_i}{p+n}I(p_i,n_i)$$

- La ganancia de información de la rama A es

$$Gain(A)=I(p,n)-E(A)$$



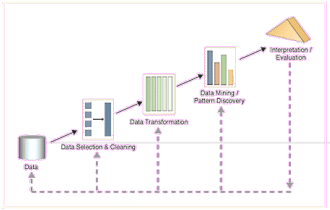
Ganancia de información en árboles de decisión

- Si se utiliza un atributo A, un conjunto S se dividirá en conjuntos {S1, S2 , …, Sv}
  - Si S_i contiene p_i ejemplos de P y n_i ejemplos de N, la entropía, o la información necesaria para clasificar objetos en todos los subárboles S_i es

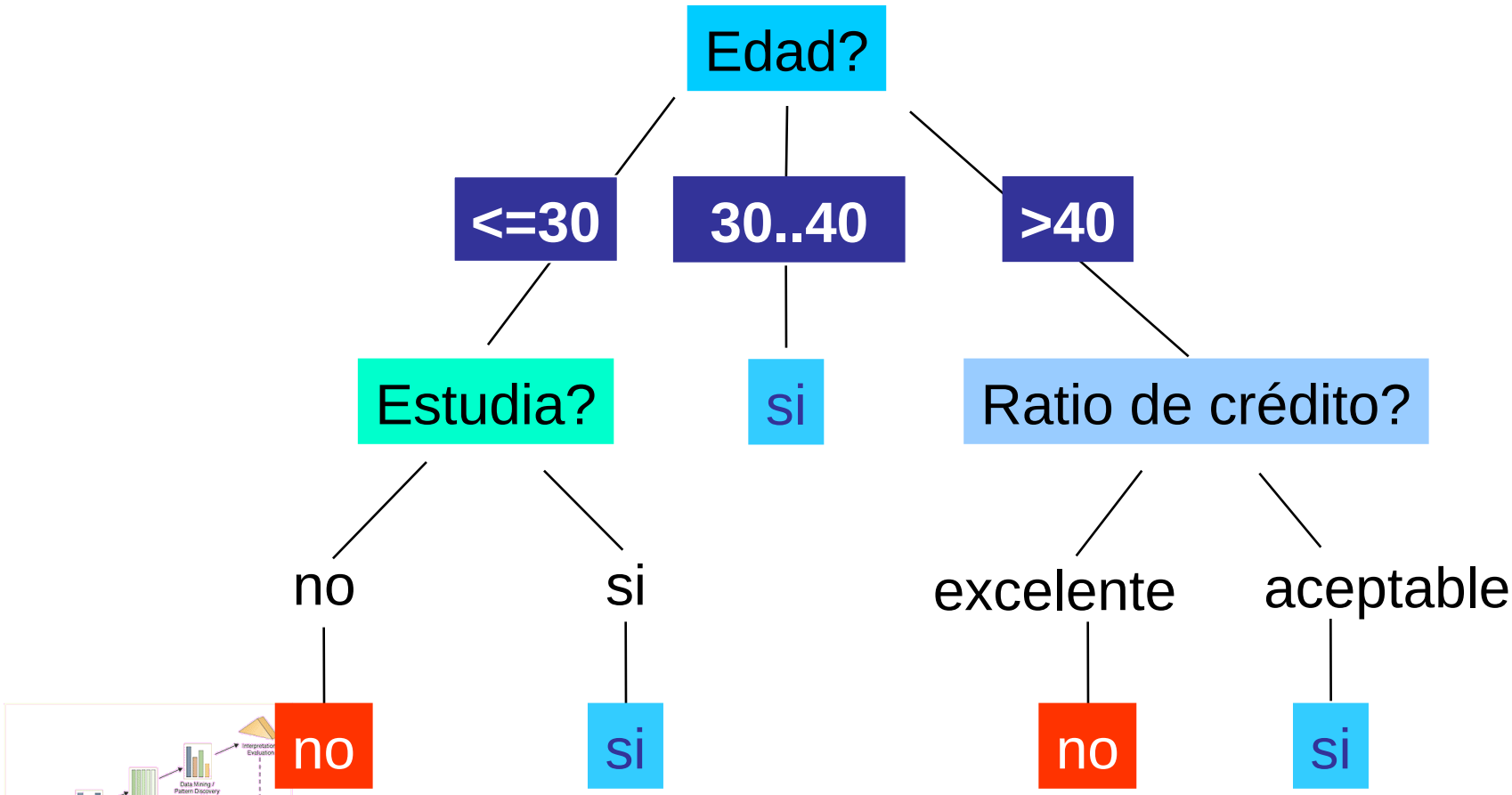
$$E(A)=\sum_{i=1}^{v}\frac{p_i+n_i}{p+n}I(p_i,n_i)$$

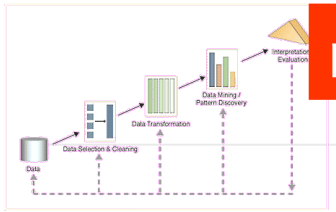
- La ganancia de información de la rama A es

$$Gain(A)=I(p,n)-E(A)$$



Ejemplo
Árbol de decisión

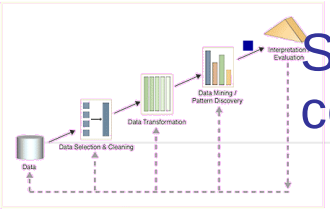




Extracción de reglas de árboles de decisión

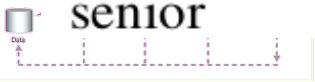
- **Si** condición **Entonces** decisión
- Se crea una regla por cada camino de la raiz a las hojas
- Cada par atributo-valor a lo largo del camino representa una conjunción

- El nodo hoja representa la clase
  - SI edad = “<=30” Y estudiante = “no” ENTONCES compra_pc = “no”
  - SI edad = “<=30” Y estudiante = “si” ENTONCES compra_pc = “SI”



Ejemplo

age	income	student	credit_rating	Class: buys_computer
youth	high	no	fair	no
youth	high	no	excellent	no
middle_aged	high	no	fair	yes
senior	medium	no	fair	yes
senior	low	yes	fair	yes
senior	low	yes	excellent	no
middle_aged	low	yes	excellent	yes
youth	medium	no	fair	no
youth	low	yes	fair	yes
senior	medium	yes	fair	yes
youth	medium	yes	excellent	yes
middle_aged	medium	no	excellent	yes
middle_aged	high	yes	fair	yes
senior	medium	no	excellent	no



Ejemplo

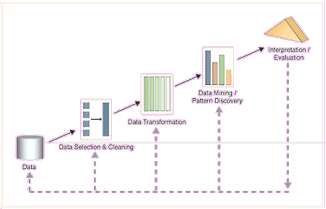
$$Info(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940 \text{ bits.}$$

Entropía del sistema

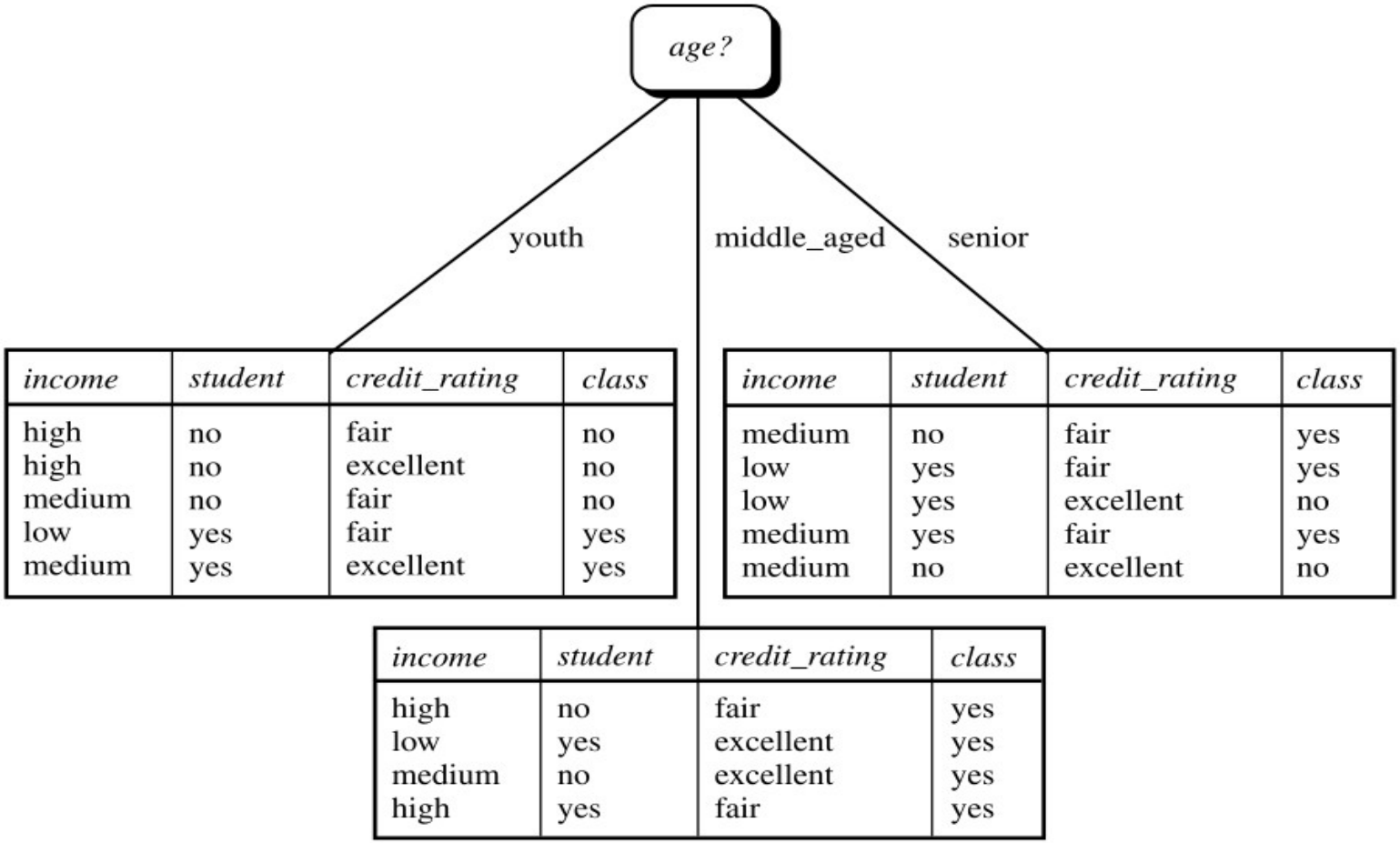
$$\begin{aligned} Info_{age}(D) &= \frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) \\ &+ \frac{4}{14} \times \left(-\frac{4}{4}\log_2\frac{4}{4}\right) \\ &+ \frac{5}{14} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) \\ &= 0.694 \text{ bits.} \end{aligned}$$

Entropía Edad

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246 \text{ bits.}$$



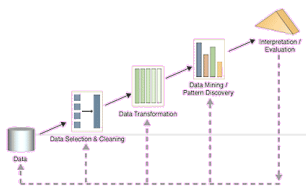
Ejemplo



income	*student*	*credit_rating*	*class*
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	*student*	*credit_rating*	*class*
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

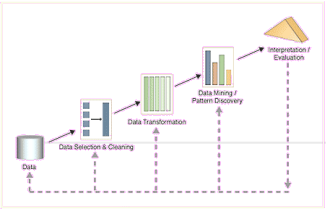
income	*student*	*credit_rating*	*class*
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes



Aplicar recursivamente el paso anterior

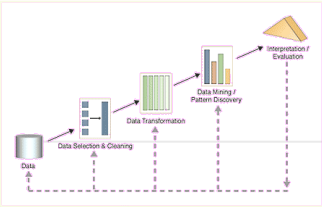
Evitar el “overfitting”

- El árbol generado es posible que sea muy exacto para el conjunto de entrenamiento
  - Demasiadas ramas puede significar que algunas son debidas al ruido o a los **outliers**
  - Poca exactitud en los ejemplos no vistos
- Dos enfoques para evitarlo
  - Prepruning
  - Postpruning



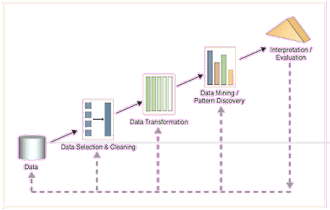
Mejoras en los árboles

- Permitir atributos con valores continuos
  - Se definen dinámicamente los valores discretos que dividen los valores del atributo en un conjunto discreto de intervalos
- Tratamiento de valores nulos
  - Se asigna el valor mas frecuente
  - Se asigna una probabilidad a cada uno de los posibles valores
- Creación de nuevos atributos que reduzcan la repetición y la replicación



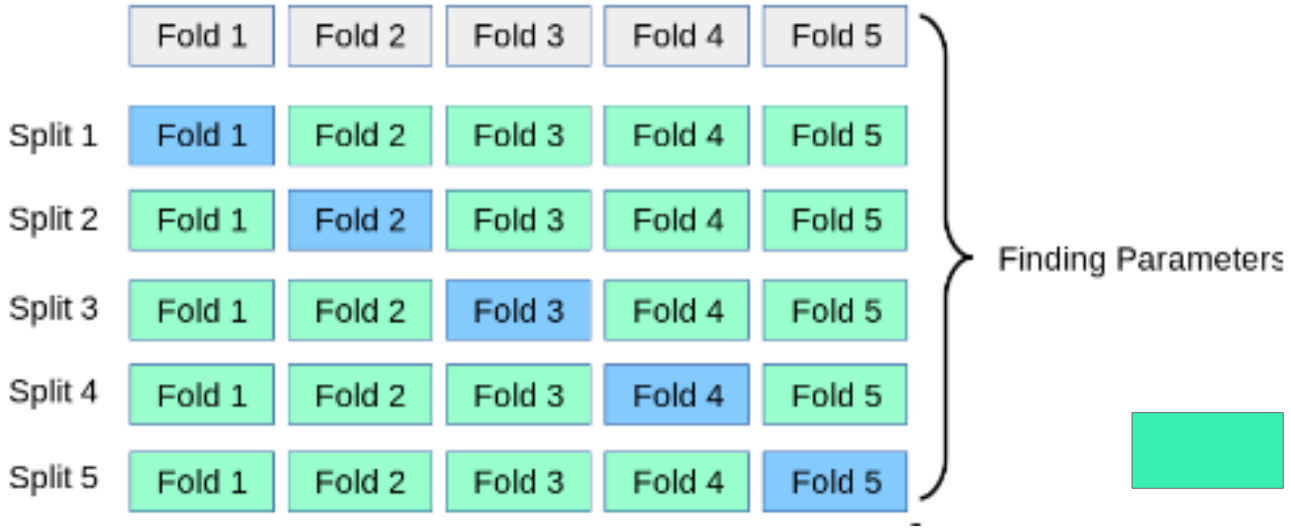
Enfoques para entrenamiento del árbol

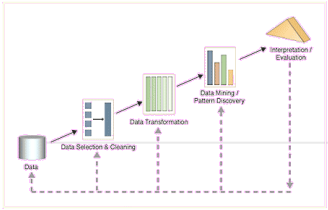
- Separa los datos en los conjuntos de **entrenamiento** (2/3) y **prueba** (⅓)
- Utilizar la **validación cruzada** e.g., la validación 10-fold
- Utilizar todos los datos para entrenamiento
  - Pero aplicar un test estadístico (e.g., chi-square) para estimar si expandir o podar un nodo



Cross Validation

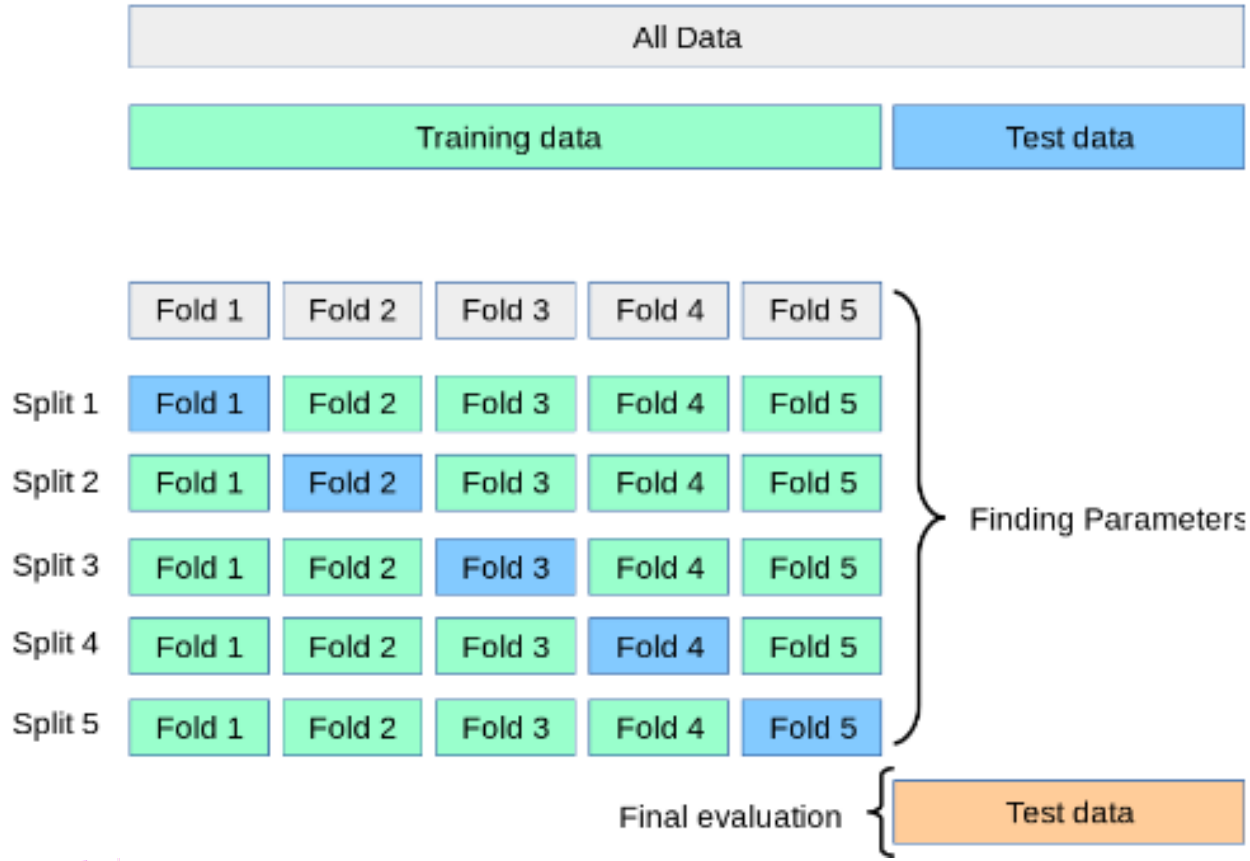
La validación cruzada consiste en **división de muestras** que utilizan diferentes porciones de los datos para **entrenar** y **probar** un modelo en diferentes iteraciones

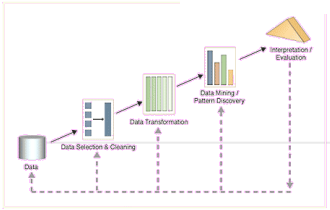




K = 5

División del Dataset (Enfoques)



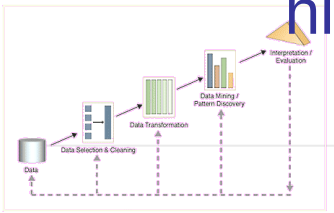


Clasificación bayesiana

- Aprendizaje probabilístico: Calcula hipótesis probabilísticas explícitas y destaca entre los enfoques mas comunes para ciertos tipos de problemas

- Incremental:Cada ejemplo puede incrementar/decrementar la probabilidad de que una hipótesis sea correcta.

- La predicción probabilística predice múltiple hipótesis ponderadas



Teorema de Bayes

- Dado un conjunto de datos, la probabilidad a posteriori de una hipótesis h es:

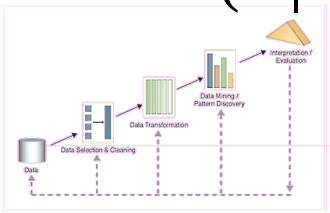
$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

P(h) = probabilidad apriori de la hipótesis h

P(D) = probabilidad apriori de los datos de entrenamiento D

P($h|D$) = probabilidad de h dado D

P($D|h$) = probabilidad de D dado h

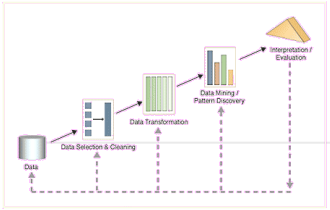


Clasificador Naïve Bayes Classifier (I)

- Suposición simplificada: los atributos son condicionalmente independientes

- Reduce enormemente el costo computacional pues solo tiene en cuenta la distribución de la clase.

$$P(C_j|V) = P(C_j)\prod_{i=1}^{n} P(v_i|C_j)$$



Clasificador Naive Bayes (II) Ejemplo:

age	income	student	credit_rating	Class: buys_computer
youth	high	no	fair	no
youth	high	no	excellent	no
middle_aged	high	no	fair	yes
senior	medium	no	fair	yes
senior	low	yes	fair	yes
senior	low	yes	excellent	no
middle_aged	low	yes	excellent	yes
youth	medium	no	fair	no
youth	low	yes	fair	yes
senior	medium	yes	fair	yes
youth	medium	yes	excellent	yes
middle_aged	medium	no	excellent	yes
middle_aged	high	yes	fair	yes
senior	medium	no	excellent	no

Clasificador Naive Bayes (II) Ejemplo:

Predecir la etiqueta de clase de la nueva tupla

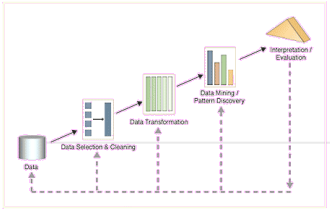
***X*=** (*age= youth, income=medium, student=yes, credit_rating=fair*)

Buys _computer => ?

Maximizar $P(X \mid C_i)\, P(C_i)$, para i= 1, 2, ... {yes, no}

1. Calcular la probabilidad a priori para las etiquetas de clase

$$P(buys\_computer = yes) = 9/14 = 0.643$$
$$P(buys\_computer = no) = 5/14 = 0.357.$$

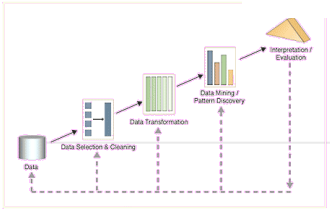


Clasificador Naive Bayes (II) Ejemplo:

***X*= (*age= youth*, *income=medium*, *student=yes*, *credit_rating=fair*)**

2. Calcular la probabilidad condicional para cada $P(X \mid C_i)$

$$
\begin{aligned}
P(age = youth \mid buys\_computer = yes) &= 2/9 = 0.222 \\
P(age = youth \mid buys\_computer = no) &= 3/5 = 0.600 \\
P(income = medium \mid buys\_computer = yes) &= 4/9 = 0.444 \\
P(income = medium \mid buys\_computer = no) &= 2/5 = 0.400 \\
P(student = yes \mid buys\_computer = yes) &= 6/9 = 0.667 \\
P(student = yes \mid buys\_computer = no) &= 1/5 = 0.200 \\
P(credit\_rating = fair \mid buys\_computer = yes) &= 6/9 = 0.667 \\
P(credit\_rating = fair \mid buys\_computer = no) &= 2/5 = 0.400.
\end{aligned}
$$



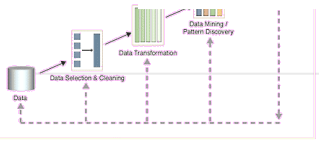
Clasificador Naive Bayes (II) Ejemplo:

3. Calcular la probabilidad condicional para $P(X \mid C)$

$$P(\mathbf{X}|buys\_computer = yes) = P(age = youth \mid buys\_computer = yes)$$
$$\times P(income = medium \mid buys\_computer = yes)$$
$$\times P(student = yes \mid buys\_computer = yes)$$
$$\times P(credit\_rating = fair \mid buys\_computer = yes)$$
$$= 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044.$$

De manera similar:

$$P(\mathbf{X}|buys\_computer = no) = 0.600 \times 0.400 \times 0.200 \times 0.400 = 0.019.$$



Clasificador Naive Bayes (II) Ejemplo:

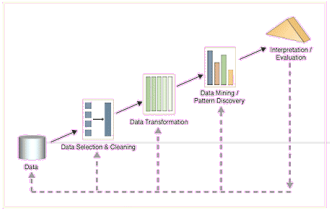
4. Maximizar $P(X \mid C_i)\, P(C_i)$

$$P(\mathbf{X}|buys\_computer = yes)\, P(buys\_computer = yes) = 0.044 \times 0.643 = 0.028$$
$$P(\mathbf{X}|buys\_computer = no)\, P(buys\_computer = no) = 0.019 \times 0.357 = 0.007.$$

Para la tupla X, el clasificar bayesiano predice:

Buys_computer => ***Yes***

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$



Clasificador Naive Bayes (II) Ejemplo:

Normalización (Probabilidades)

$$P(buys\_computer = yes) = 0.028 / (0.028 + 0.007) = 0{,}80$$

$$P(buys\_computer = no) = 0.007 / (0.028 + 0.007) = 0{,}20$$

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$