Clasificación – Conceptos Básicos y Técnicas

IDM - 2021

Clasificación: Definición

- Dada una colección de registros (conjunto de entrenamiento)
  - Cada registro esta caracterizado por una tupla (x,y), donde x es el conjunto de atributos e y es la etiqueta de clase
    - x: atributo, predictor, variable independiente, entrada
    - y: clase, respuesta, variable dependiente, salida

- **Tarea**:
  - **Aprender** un modelo que mapea cada conjunto de atributos x en una de las etiquetas predefinidas de clase y

Ejemplos de Tareas de Clasificación

Tarea	Conjunto de Atributos, ***x***	Etiqueta de Clase, *y*
Categorizar correos electrónicos	Features extraídas del correo electrónico (encabezado y contenido)	spam o no-spam
Identificar celulas tumorales	Features extraídas de escaneos MRI	Células malignas o benignas
Catalogar Galaxias	Features extraidas de imagenes obtenidas por telescopios	Elípticas, espirales, o galaxias irregulares.

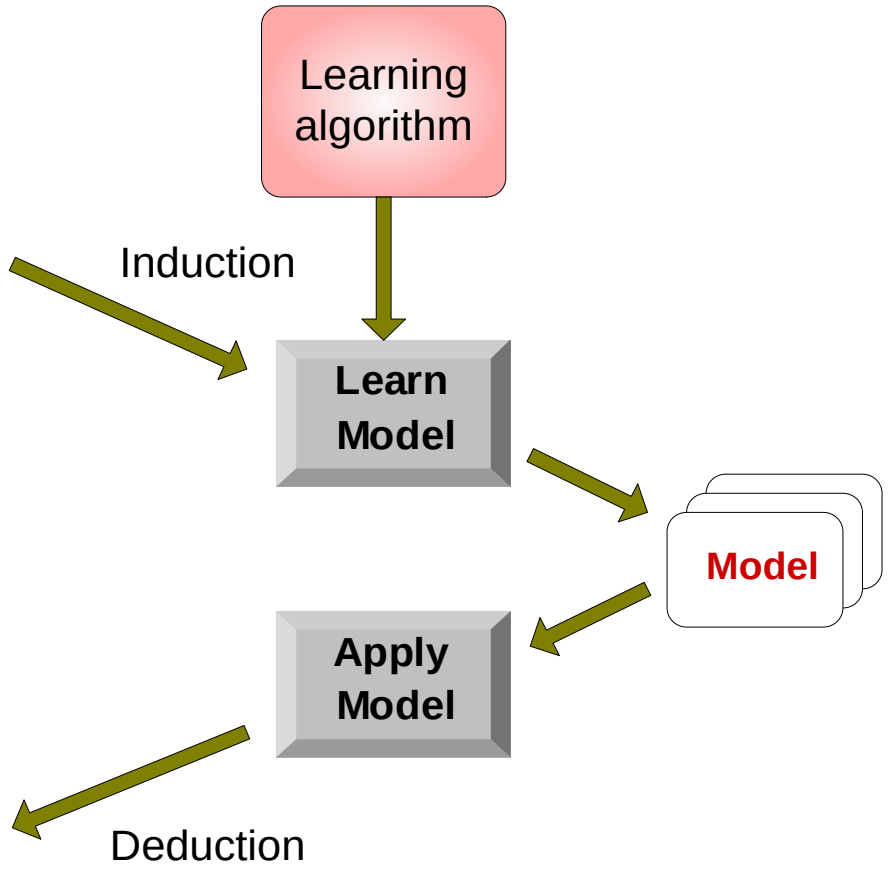
Aproximación General a la construcción de un Modelo de Clasificación

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Tecnicas de Clasificacion

- *Clasificadores Base*
  - Métodos basados en Arboles de Decisión
  - Métodos Basados en Reglas
  - Vecino Mas cercano
  - Redes Neuronales
  - Deep Learning
  - Naïve Bayes y Redes de Creencias Bayesianas
  - Support Vector Machines (Maquinas de Soporte Vectorial)

- *Clasificadores en Conjunto (Ensemble Classifiers)*
  - Boosting, Bagging, Random Forests

Matriz de Confusión para un problema de 2 clases		**Clase Predicha**	
		clase = 1	clase = 0
Clase Conocida	clase = 1	f_{11}	f_{10}
	clase = 0	f_{01}	f_{00}

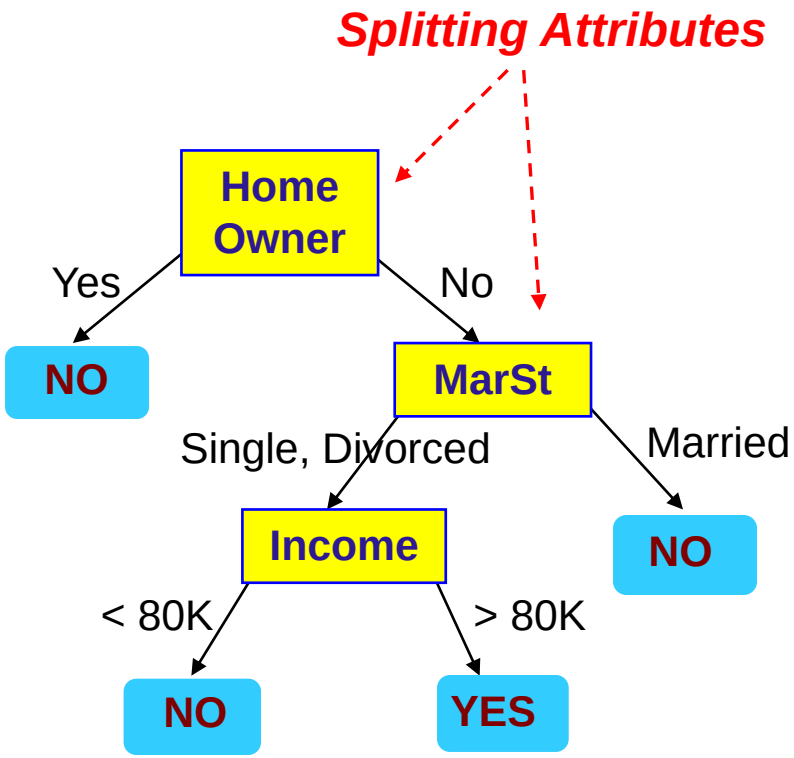
$$Accuracy = \frac{\#\ de\ Predicciones\ Correctas}{\#\ de\ Predicciones} = \frac{f_{11} + f_{10}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

$$Tasa\ de\ error = \frac{\#\ de\ Predicciones\ Incorrectas}{\#\ de\ Predicciones} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

Ejemplo de un Árbol de Decisión

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
	categorical	categorical	continuous	class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

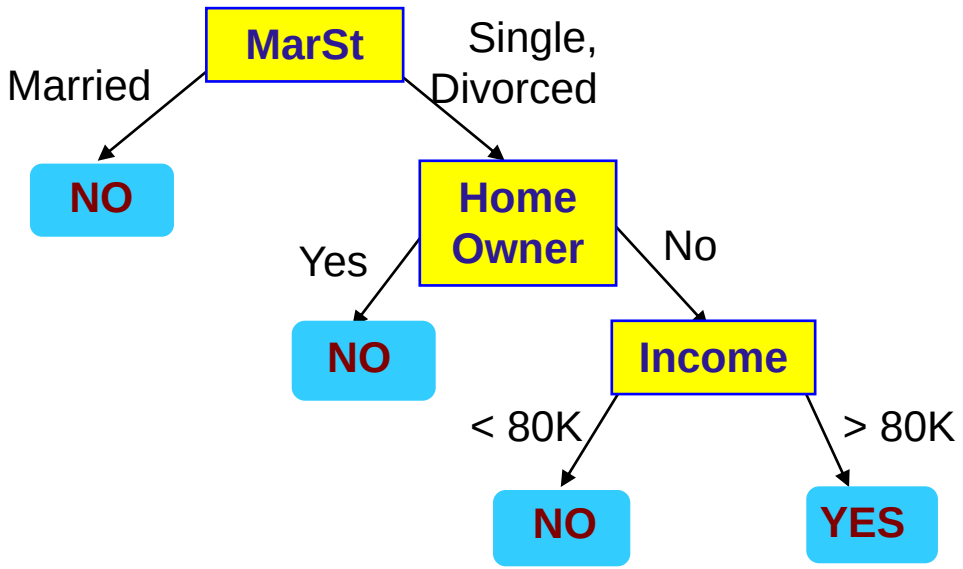
Training Data



Model: Decision Tree

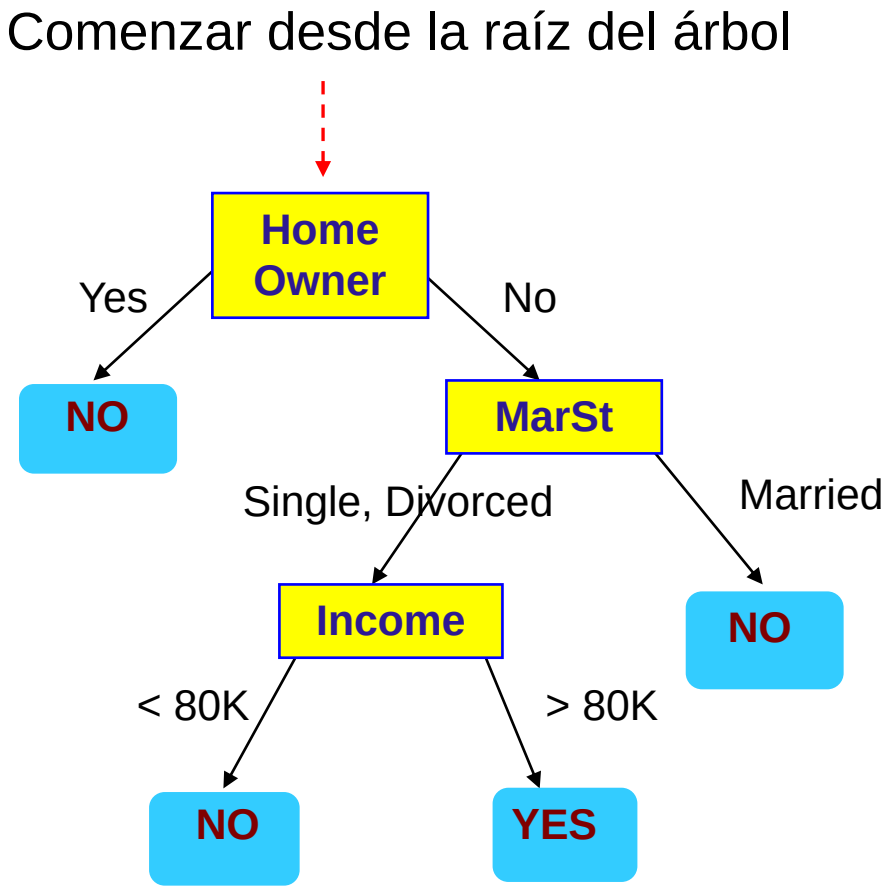
Otro Ejemplo de un Árbol de Decisión

	categorical	categorical	continuous	class
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Puede existir mas de un árbol que ajuste al mismo conjunto de datos!!!

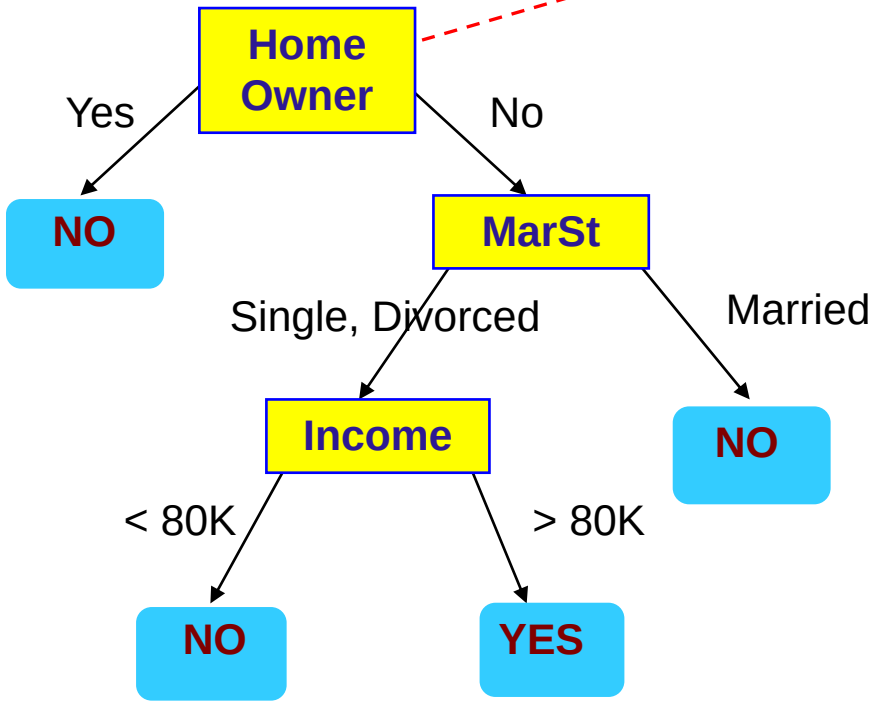
Aplicar el Modelo a los Datos de Testeo



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

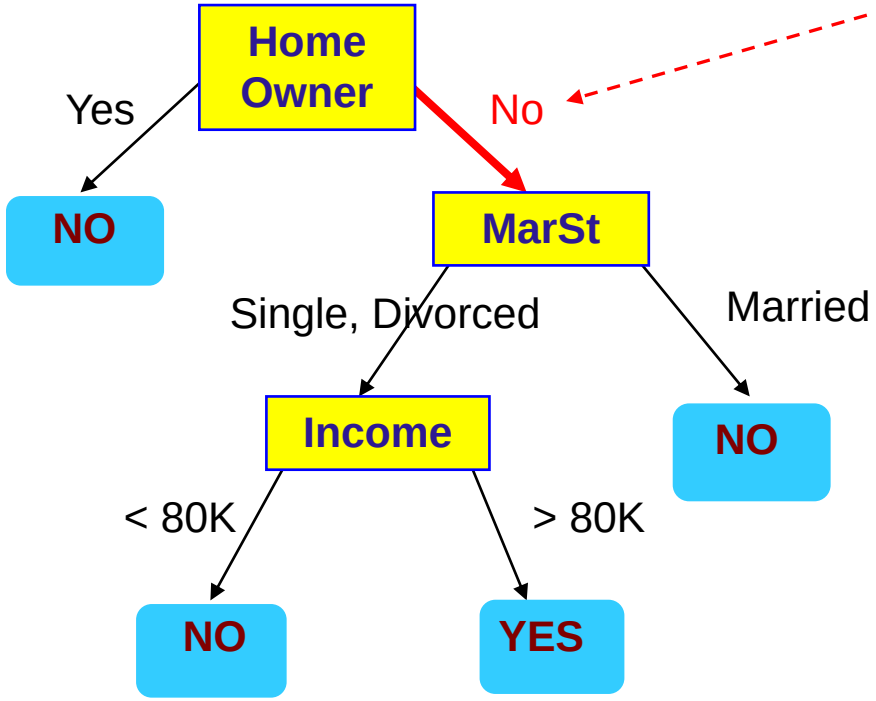
Aplicar el Modelo a los Datos de Testeo



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

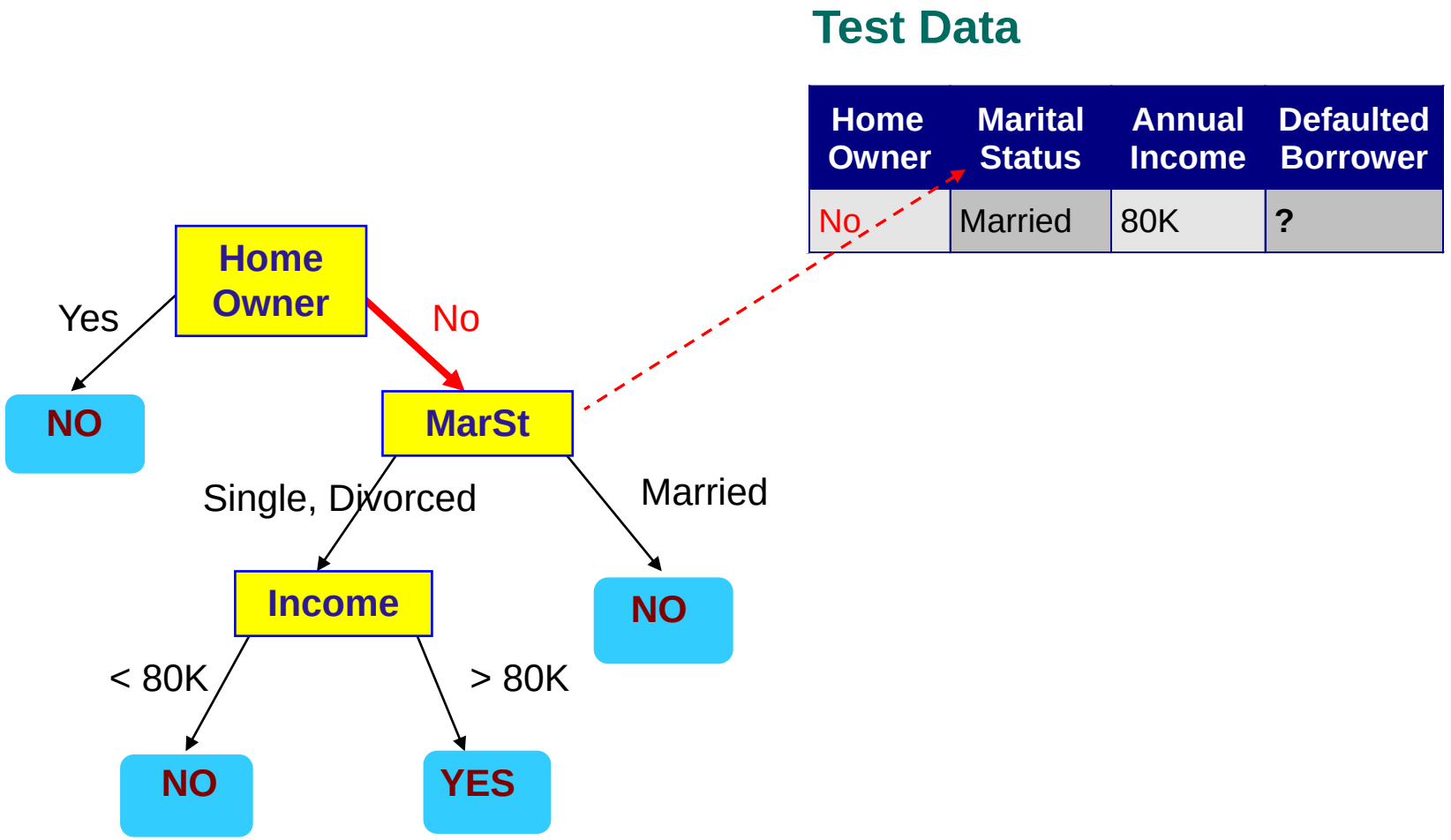
Aplicar el Modelo a los Datos de Testeo



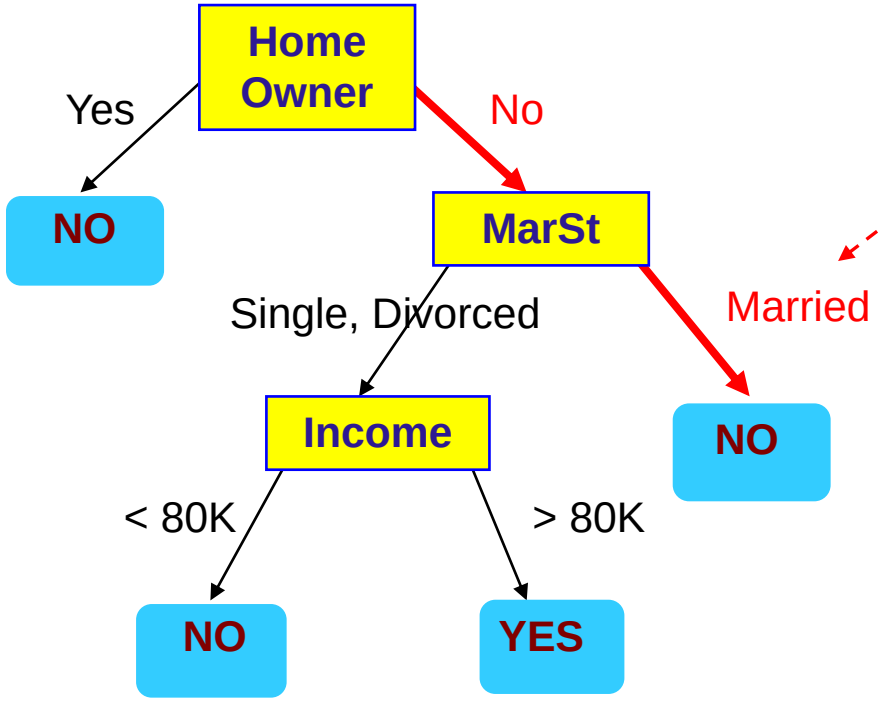
Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Aplicar el Modelo a los Datos de Testeo



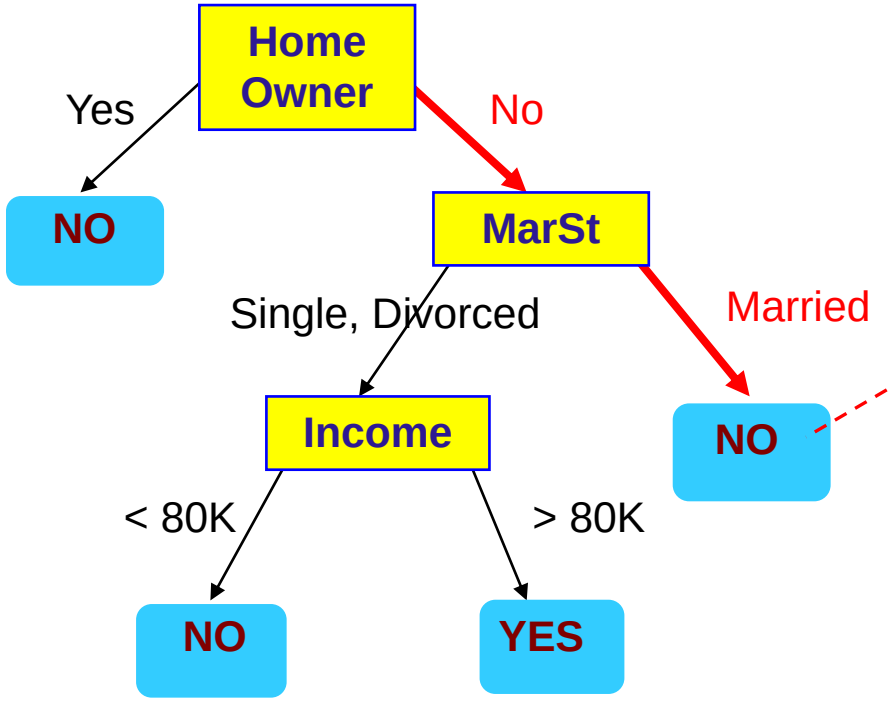
Aplicar el Modelo a los Datos de Testeo



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Aplicar el Modelo a los Datos de Testeo



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Asignar a “NO” por defecto

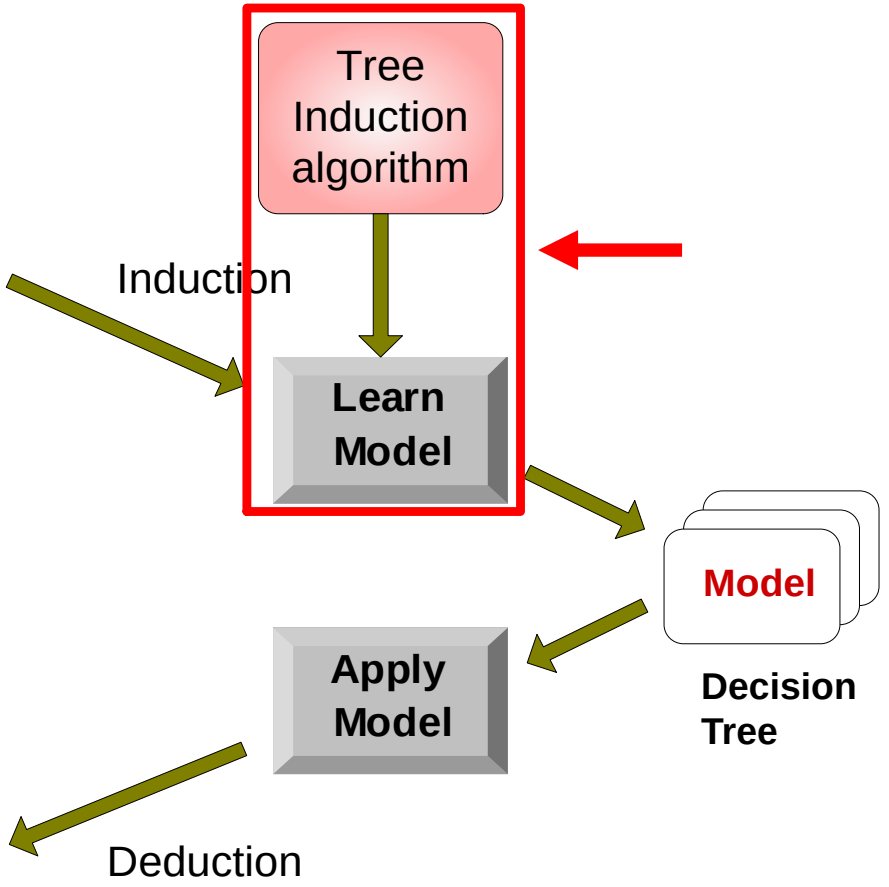
Tarea de Clasificación de un Árbol de Decisión

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Inducción de Arboles de Decisión

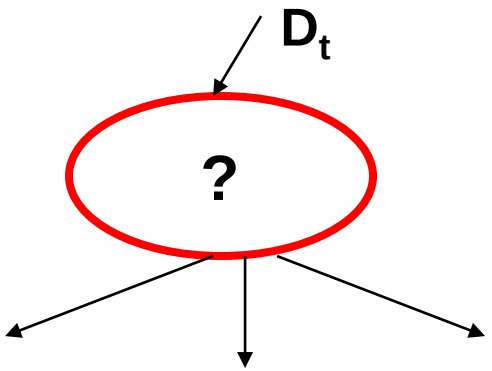
- Varios Algoritmos:
  - Algoritmo de Hunt (uno de los primeros)
  - CART
  - ID3, C4.5, C5.0
  - SLIQ, SPRINT

Estructura General del Algoritmo de Hunt

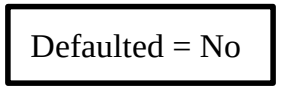
- Sea D_t el conjunto de registros de entrenamiento que llegan a un nodo t

- Procedimiento General:
  - Si D_t contiene registros que pertenecen a la misma clase y_t, entonces t es un nodo hoja que se etiqueta como y_t
  - Si D_t contiene registros que pertenecen a mas de una clase, utilice un test de atributo para partir los datos en subconjuntos mas pequeños. Aplique recursivamente el procedimiento a cada subconjunto.

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Algoritmo de Hunt

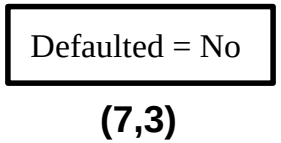


(7,3)

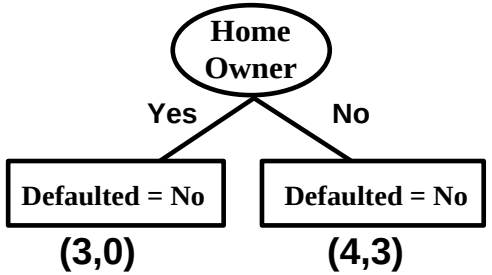
(a)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Algoritmo de Hunt



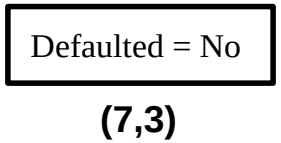
(a)



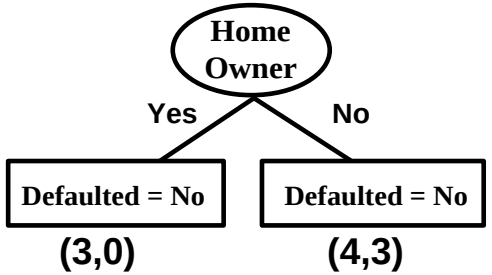
(b)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Algoritmo de Hunt

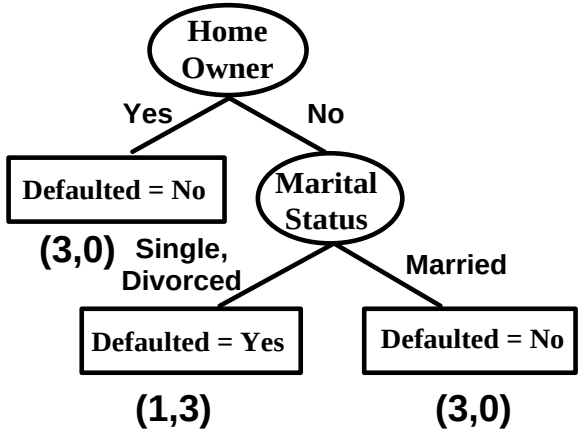


(a)



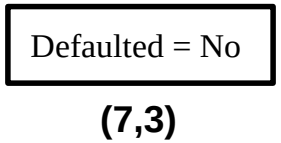
(b)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

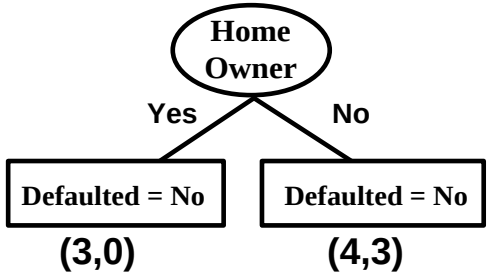


(c)

Algoritmo de Hunt

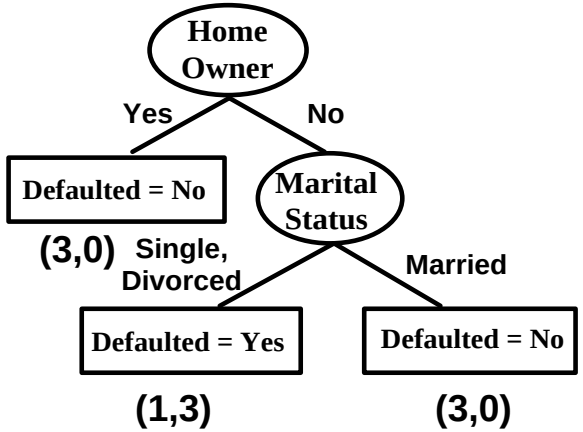


(a)

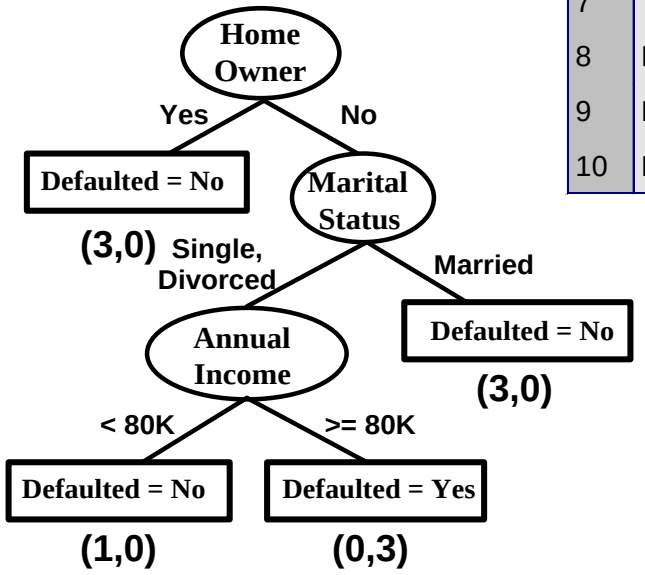


(b)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



(c)



(d)

Decisiones de Diseño de un Algoritmo de Inducción de Arboles de Decisión

- Como se deben partir los registros de entrenamiento (*split*)?
  - Método para especificar la condición de test
    - depende del tipo de atributo
  - Medida para evaluar la bondad de una condición de testeo

- Como debe para el procedimiento de *split*?
  - Dejar de dividir si todos los registros pertenecen a la misma clase o tienen valores idénticos de los atributos.
  - Terminación temprana

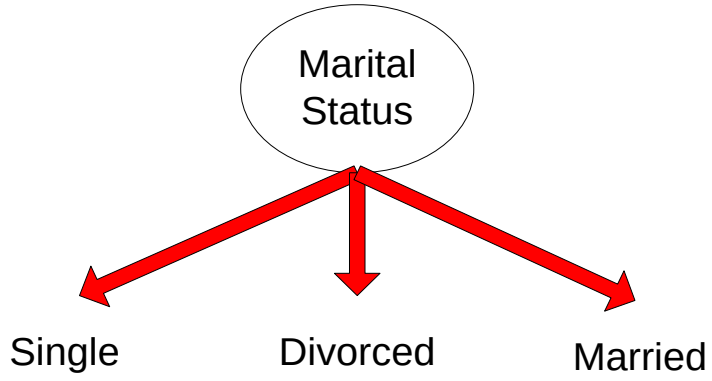
Métodos para expresar las condiciones de Testeo

- Dependen del tipo de atributo
  - Binario
  - Nominal
  - Ordinal
  - Continuo

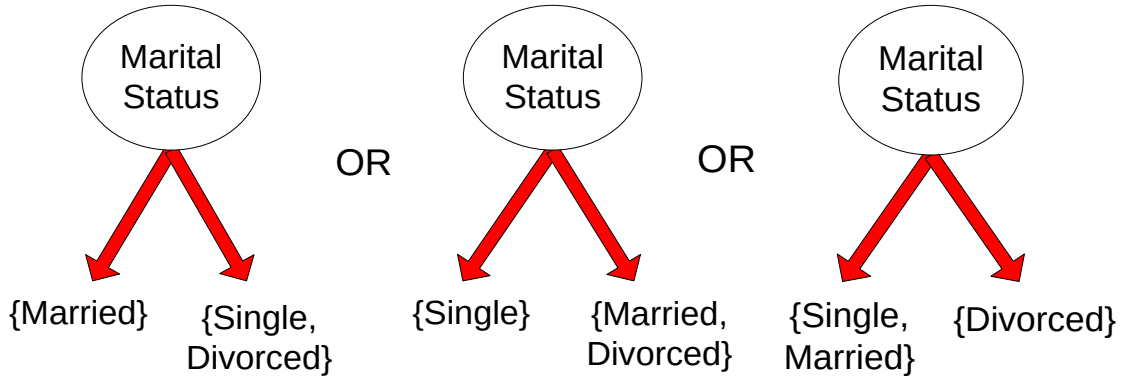
- Dependen de la cantidad de formas de realizar el split
  - Split de 2-vias
  - Split Multi-via

Condicion de Testeo para Atributos Nominales

- Split Multi-vía:
  - Utilizar tantas particiones como valores distintos.

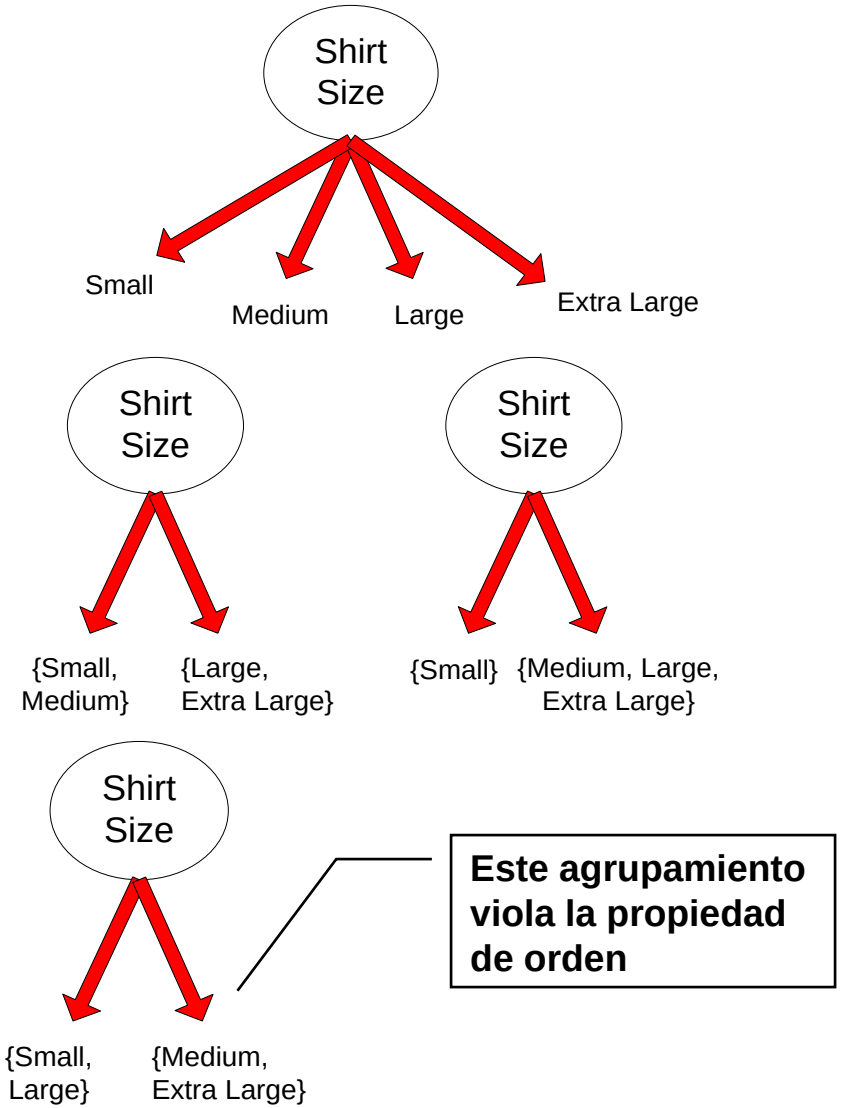


- Split Binario:
  - Divide los valores en dos subconjuntos

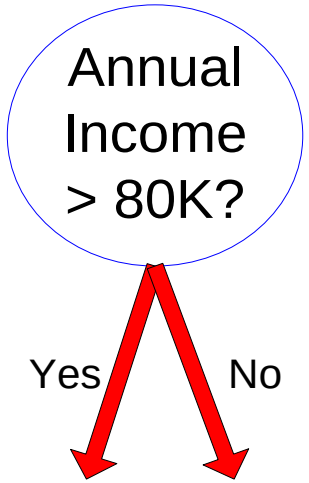


Condición de Testeo para Atributos Ordinales

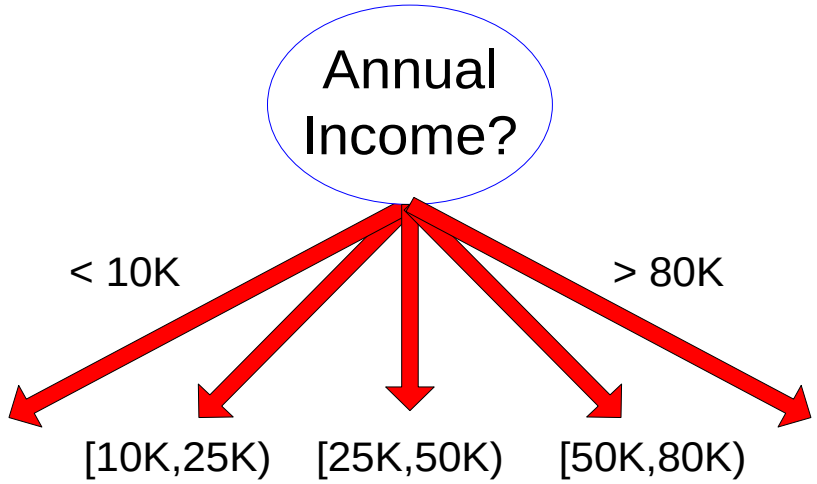
- Split Multi-vía:
  - Utilizar tantas particiones como valores distintos
- Split Binario:
  - Divide los valores en dos subconjuntos
  - Preserva la propiedad de orden entre los valores de los atributos



Condición de Testeo para Atributos Continuos



(i) Binary split



(ii) Multi-way split

Particionamiento Basado en Atributos Continuos

- Diferentes formas de atacar el problema
  - Discretizando a alguna forma de atributo ordinal o categórico

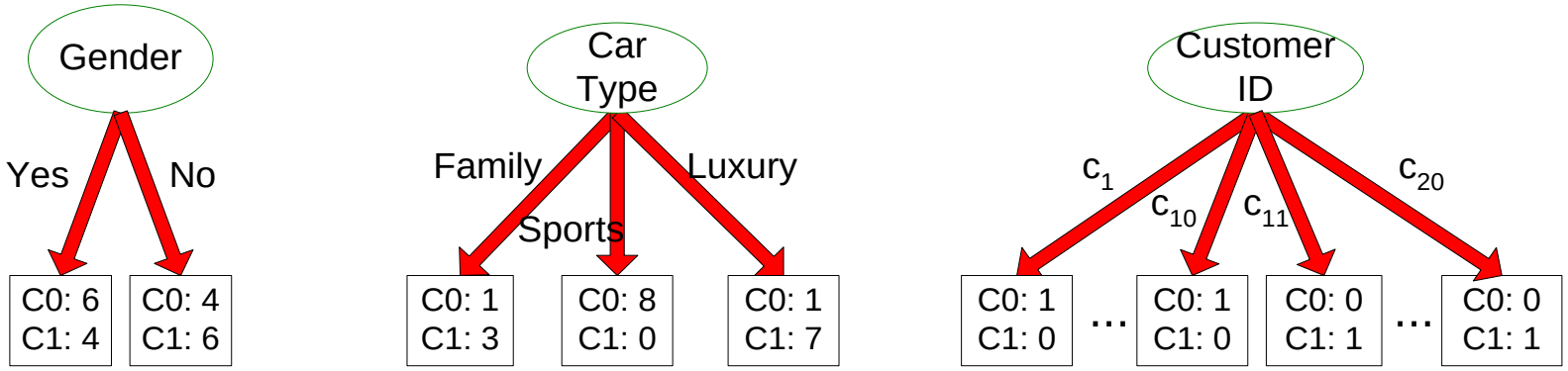
    Se pueden construir rangos mediante mecanismos de intervalos iguales, frecuencia igual (percentiles), clustering, etc.
    - Estático – discretizar una vez al comienzo
    - Dinámico – repetir en cada nodo

  - Decisión Binaria: $(A < v)$ or $(A \geq v)$
    - considerar todos los posibles splits y encontrar el mejor.
    - puede ser computacionalmente intensivo

Como determinar el mejor split?

Antes de Partir: 10 registros de la clase 0, 10 registros de la clase 1

Customer Id	Gender	Car Type	Shirt Size	Class
1	M	Family	Small	C0
2	M	Sports	Medium	C0
3	M	Sports	Medium	C0
4	M	Sports	Large	C0
5	M	Sports	Extra Large	C0
6	M	Sports	Extra Large	C0
7	F	Sports	Small	C0
8	F	Sports	Small	C0
9	F	Sports	Medium	C0
10	F	Luxury	Large	C0
11	M	Family	Large	C1
12	M	Family	Extra Large	C1
13	M	Family	Medium	C1
14	M	Luxury	Extra Large	C1
15	F	Luxury	Small	C1
16	F	Luxury	Small	C1
17	F	Luxury	Medium	C1
18	F	Luxury	Medium	C1
19	F	Luxury	Medium	C1
20	F	Luxury	Large	C1



Cual condición de testeo es la mejor?

Como determinar el mejor *split*

- Aproximación Voraz:
  - Nodos con clases puras son preferidos.

- Se necesita de una medida de la impureza del nodo:

C0: 5 C1: 5	C0: 9 C1: 1
Alto nivel de impureza	**Bajo nivel de impureza**

Medidas de la Impureza de un Nodo

- Índice de Gini

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

- Entropía

$$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

- Error de Clasificación

$$Error(t) = 1 - \max_i P(i \mid t)$$

Como encontrar el mejor Split

1. Calcular la medida de impureza (P) antes del Split.

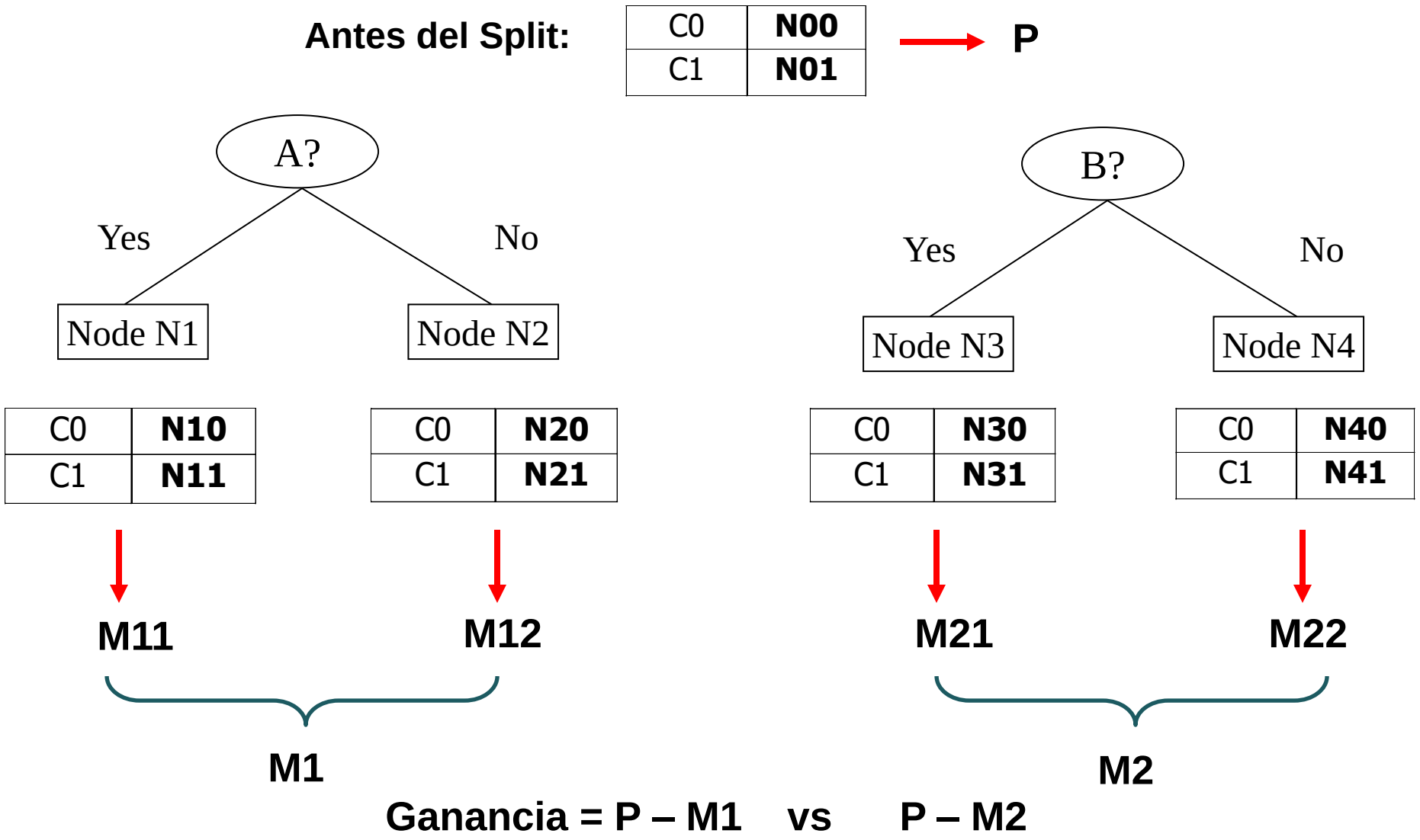
2. Calcular la medida de impureza (M) después del Split.
   - Calcular la medida de impureza para cada nodo hijo.
   - M es la impureza ponderada de los hijos

3. Elegir la condición de testeo del atributo que produce la mayor ganancia.

$$\textbf{Ganancia = Gain = P – M}$$

o equivalentemente, la menor medida de impureza luego del Split (M)

Encontrando el mejor Split



Medida de Impureza: GINI

- El índice de Gini para un dado nodo t es:

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

(NOTA: $p(j \mid t)$ es la frecuencia relativa de la clase j en el nodo t).

  - Máximo (1 - 1/n_c) cuando los registros están igualmente distribuidos entre todas las clases, implicando la información menos interesante.
  - Mínimo (0.0) cuando todos los registros pertenecen a una sola clase, implicando la información mas interesante

Medida de Impureza: GINI

- Índice de Gini para un dado nodo t :

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

(NOTA: $p(j \mid t)$ es la frecuencia relativa de la clase j en el nodo t).

  - Para un problema de 2 clases (p, 1 – p):
    - GINI = $1 - p^2 - (1 - p)^2 = 2p\ (1\text{-}p)$

C1	**0**
C2	**6**
Gini=0.000	

C1	**1**
C2	**5**
Gini=0.278	

C1	**2**
C2	**4**
Gini=0.444	

C1	**3**
C2	**3**
Gini=0.500	

Cálculo del Índice de Gini para un único Nodo

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Gini = 1 – P(C1)² – P(C2)² = 1 – 0 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Gini = 1 – (1/6)² – (5/6)² = 0.278

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Gini = 1 – (2/6)² – (4/6)² = 0.444

Cálculo del Índice de Gini para una colección de Nodos

- Cuando un nodo p se parte en k particiones (hijos)

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

donde, n_i = cantidad de registros en el hijo i,

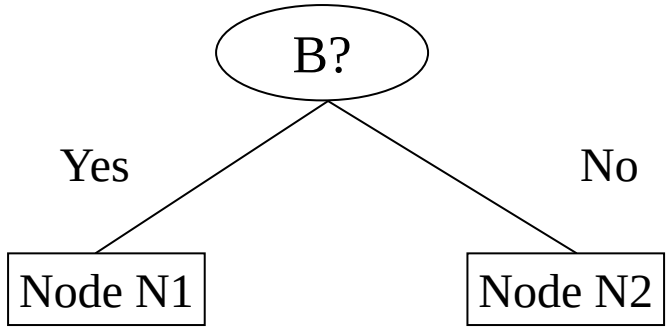
n = cantidad de registros en el nodo padre p.

- Elegir el atributo que minimiza el promedio ponderado del Índice de Gini de los nodos hijos

- El Índice de Gini se utiliza en los algoritmos de inducción de arboles de decisión CART, SLIQ, SPRINT

Atributos Binarios: Cálculo del Índice de GINI

- Partir en dos particiones
- Efecto de particiones ponderadas:
  - Se buscan particiones grandes y puras.



	Parent
C1	**7**
C2	**5**
Gini = 0.486	

Gini(N1)
= 1 – (5/6)² – (1/6)²
= 0.278

Gini(N2)
= 1 – (2/6)² – (4/6)²
= 0.444

	N1	N2
C1	**5**	**2**
C2	**1**	**4**
Gini=0.361		

Gini Ponderado de N1 N2
= 6/12 * 0.278 +
6/12 * 0.444
= 0.361

Ganancia = 0.486 – 0.361 = 0.125

Atributos Categóricos: Cálculo del Índice de GINI

- Para cada valor distinto, calcular los conteos de cada clase en el dataset
- Utilizar la matriz de conteo para tomar la decisión.

Split multi-vía

	CarType		
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini	0.163		

Split Binario
(encontrar la mejor partición de valores)

	CarType	
	{Sports, Luxury}	{Family}
C1	9	1
C2	7	3
Gini	0.468	

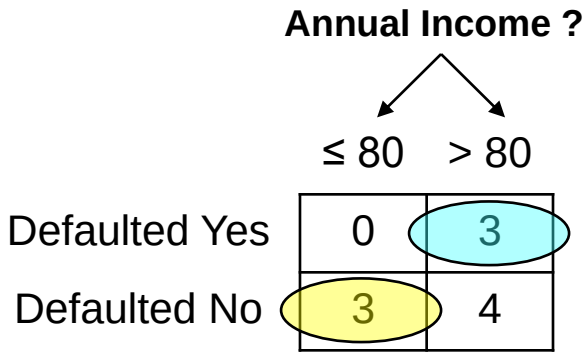
	CarType	
	{Sports}	{Family, Luxury}
C1	8	2
C2	0	10
Gini	0.167	

Cual de estos es mejor?

Atributos Continuos: Cálculo del Índice de GINI

- Utilizar Decisiones Binarias Basándose en un valor.
- Existen varias elecciones para el valor de partición
  - Numero de valores posibles para el split = Numero de valores distintos
- Cada valor de Split tiene una matriz de conteo asociado con él
  - Las clases de conteo en cada una de las particiones, A < v and A ≥ v
- El método mas simple para elegir el mejor valor de v
  - Para cada v, escanear la base de datos para encontrar la matriz de conteo y calcular el índice de Gini
  - Computacionalmente Ineficiente! Repite muchas veces el mismo trabajo.

ID	Home Owner	Marital Status	Annual Income	Defaulted
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



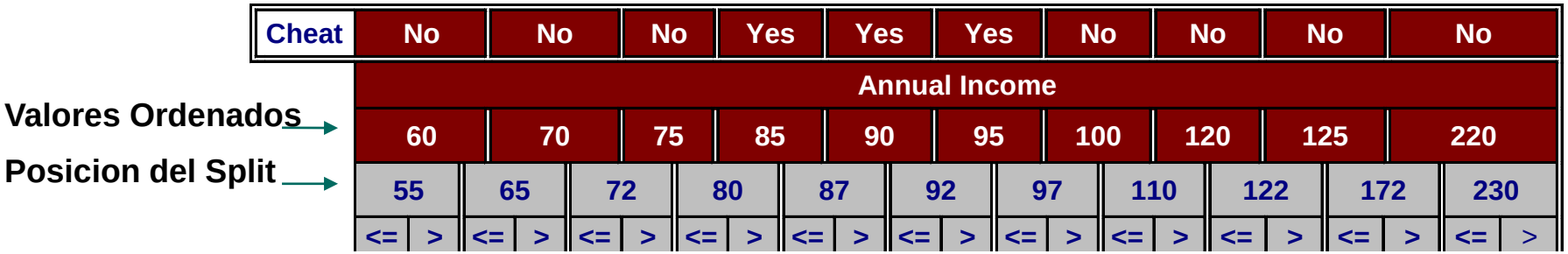
Atributos Continuos: Calculo del índice de GINI...

- Cálculo eficiente: Para cada atributo,
  - Ordenar los valores
  - Linealmente escanear los valores, y en cada momento actualizar la matriz de conteo y calcular el índice de GINI
  - Elegir la ubicación del Split que obtiene el menor valor para el índice de GINI

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
	Annual Income									
Valores Ordenados →	60	70	75	85	90	95	100	120	125	220

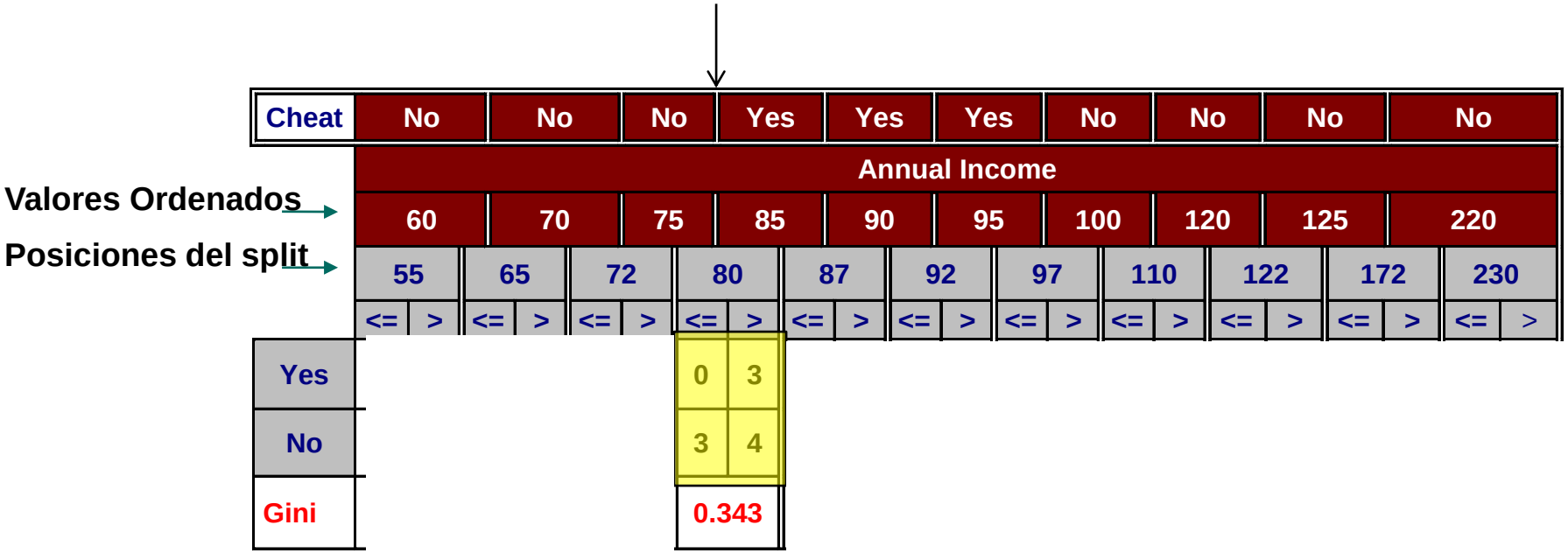
Atributos Continuos: Calculo del indice de GINI...

- Cálculo eficiente: Para cada atributo,
  - Ordenar los valores
  - Linealmente escanear los valores, y en cada momento actualizar la matriz de conteo y calcular el índice de GINI
  - Elegir la ubicación del Split que obtiene el menor valor para el índice de GINI



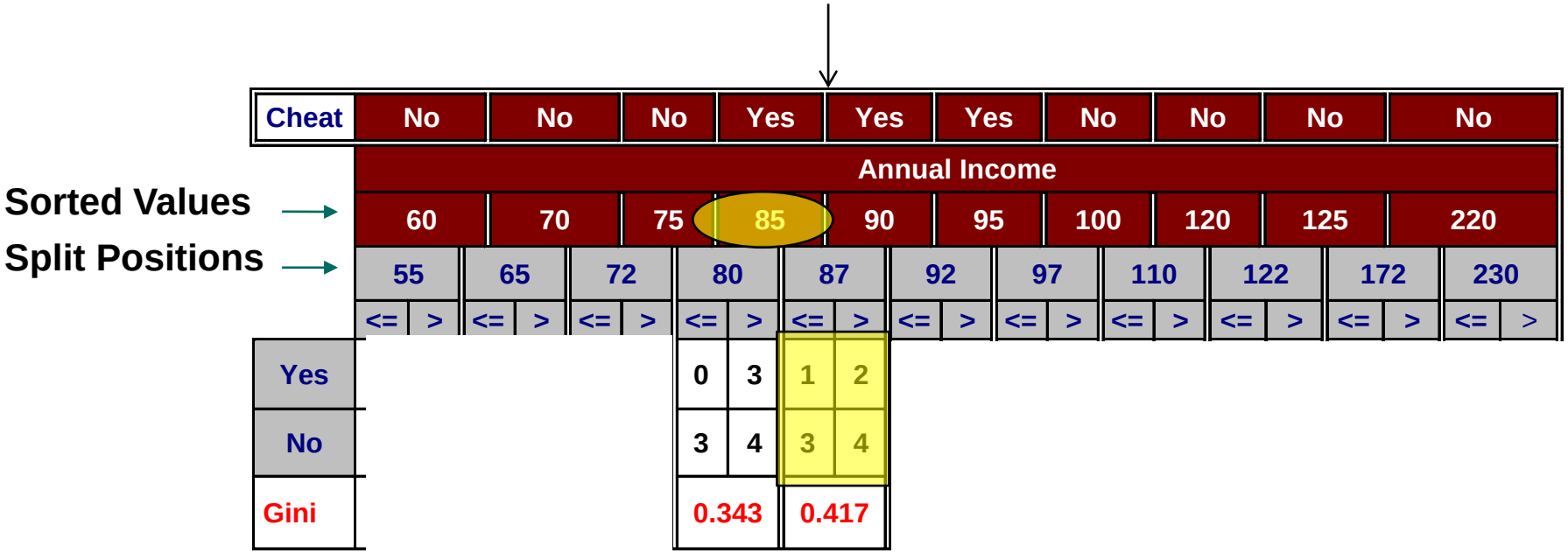
Atributos Continuos: Calculo del indice de GINI...

- Cálculo eficiente: Para cada atributo,
  - Ordenar los valores
  - Linealmente escanear los valores, y en cada momento actualizar la matriz de conteo y calcular el índice de GINI
  - Elegir la ubicación del Split que obtiene el menor valor para el índice de GINI



Atributos Continuos: Calculo del indice de GINI...

- Cálculo eficiente: Para cada atributo,
  - Ordenar los valores
  - Linealmente escanear los valores, y en cada momento actualizar la matriz de conteo y calcular el índice de GINI
  - Elegir la ubicación del Split que obtiene el menor valor para el índice de GINI



Atributos Continuos: Calculo del indice de GINI...

- Cálculo eficiente: Para cada atributo,
  - Ordenar los valores
  - Linealmente escanear los valores, y en cada momento actualizar la matriz de conteo y calcular el índice de GINI
  - Elegir la ubicación del Split que obtiene el menor valor para el índice de GINI

Cheat	No		No		No		Yes		Yes		Yes		No		No		No		No		No	
Annual Income																						
Sorted Values →	60		70		75		85		90		95		100		120		125		220			
Split Positions →	55		65		72		80		87		92		97		110		122		172		230	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
Gini	0.420		0.400		0.375		0.343		0.417		0.400		***0.300***		0.343		0.375		0.400		0.420	

Medida de Impureza: Entropía

- La entropía en un dado nodo t:

$$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

(NOTA: $p(j \mid t)$ es la frecuencia relativa de la clase j en el nodo t).

  - Máximo ($\log n_c$) cuando todos los registros están igualmente distribuidos entre todas las clases, implicando la menor cantidad de información.
  - Mínimo (0.0) cuando todos los registros pertenecen a la misma clase, implicando la máxima información.

  - Los cálculos basados en entropía son similares a los cálculos del índice de GINI

Calculo de la Entropía para un único nodo

$$Entropy(t) = -\sum_j p(j \mid t) \log_2 p(j \mid t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

Calculando la Ganancia de Información Después del Split

- Ganancia de Información:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

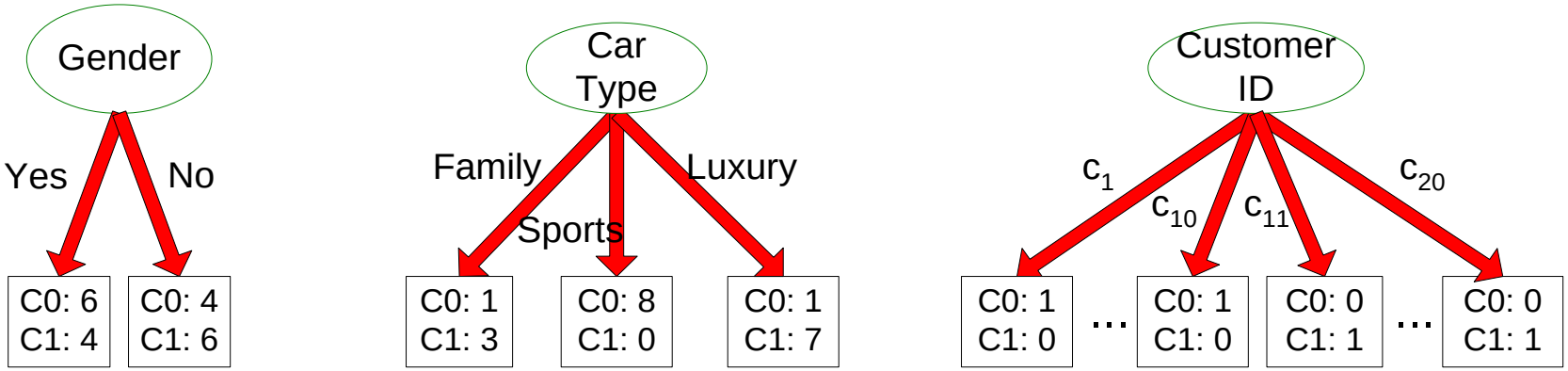
Nodo padre, p es partido en k particiones;

n_i es la cantidad de registros en la partición i

  - Elegir el Split que obtenga la mayor reducción (maximiza la GANANCIA)
  - Usada en los algoritmos de inducción de arboles de decisión ID3 y C4.5

Problema con una gran cantidad de particiones

- Las medidas de impureza de los nodos tienden a preferir *splits* que resulten en una gran cantidad de particiones, cada una de ellas pequeña pero pura.



  - El ID de cliente tiene la mayor ganancia de información porque la entropía de todos sus hijos es cero

Gain Ratio (Razon de Ganancia)

- *Gain Ratio*:

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO} \qquad SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

Nodo Padre, p se parte en k particiones

n_i es el numero de registros en la partición i

  - Ajusta la Ganancia de Información por la entropía de la partición (SplitINFO).
    - Se penaliza una alta entropía de la partición (gran cantidad de pequeñas particiones)!!
  - Lo utiliza el algoritmo C4.5
  - Desinada para sobreponerse a las desventajas de la ganancia de información.

Gain Ratio

- Gain Ratio:

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO} \qquad SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

El nodo padre, p se parte en k particiones

n_i es la cantidad de registros en la particion i

	CarType		
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini	0.163		

SplitINFO = 1.52

	CarType	
	{Sports, Luxury}	{Family}
C1	9	1
C2	7	3
Gini	0.468	

SplitINFO = 0.72

	CarType	
	{Sports}	{Family, Luxury}
C1	8	2
C2	0	10
Gini	0.167	

SplitINFO = 0.97

Medida de Impureza: Error de Clasificacion

- El Error de Clasificación en el nodo t :

$$Error(t) = 1 - \max_i P(i \mid t)$$

  - Máximo (1 - 1/n_c) cuando todos los registros están igualmente distribuidos entre todas las clases, implicando la menor cantidad de información.
  - Mínimo (0) cuando todos los registros pertenecen a una sola clase, implicando la información mas interesante.

Calculando el error en único nodo

$$Error(t) = 1 - \max_i P(i \mid t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Error = 1 – max (0, 1) = 1 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

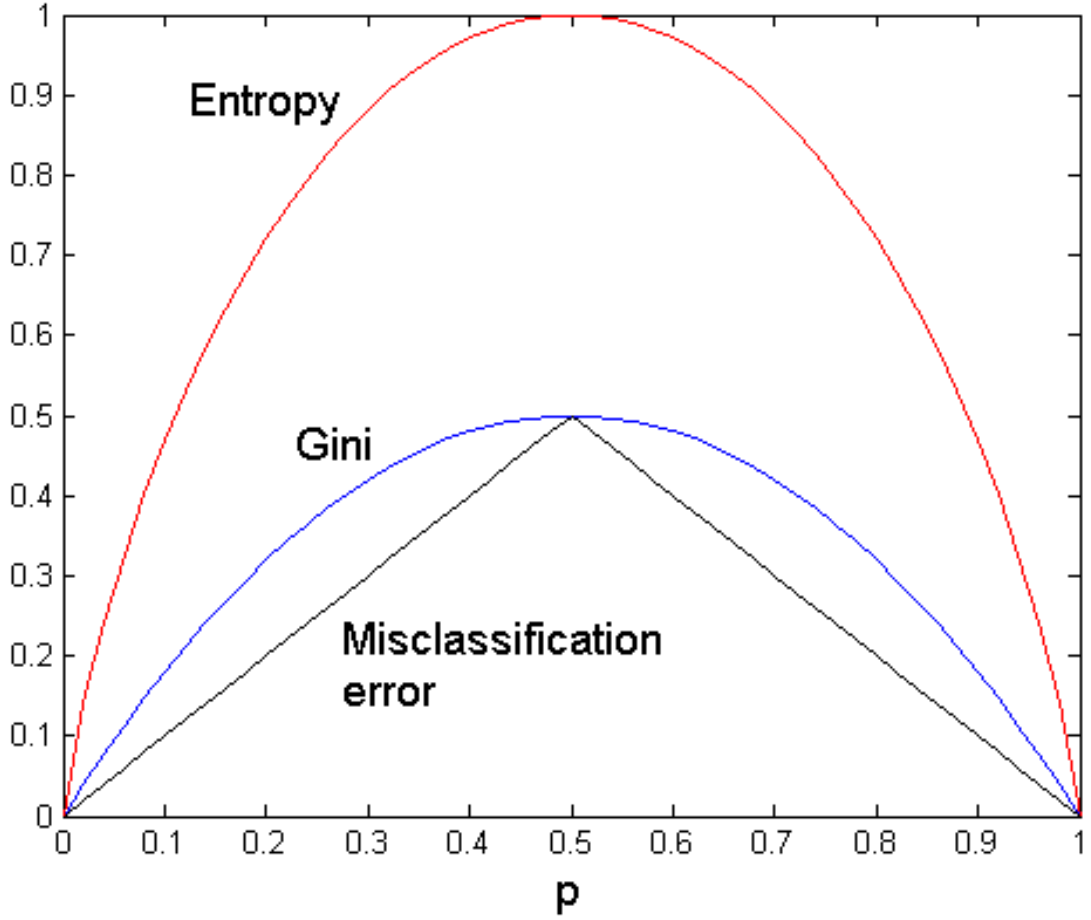
C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

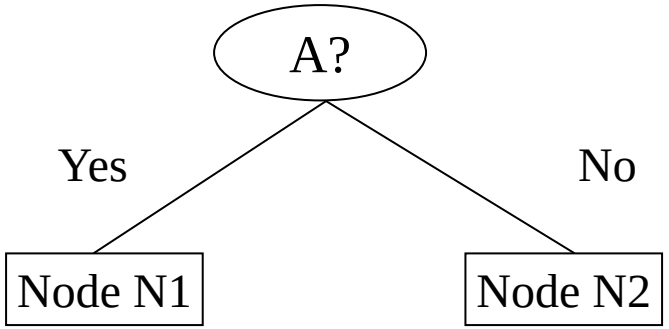
Error = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

Comparación entre las medidas de impureza

Para un problema de 2 clases:



Error de Clasificación vs. Índice de GINI



	Parent
C1	7
C2	3
Gini = 0.42	

Gini(N1)
= 1 – (3/3)² – (0/3)²
= 0

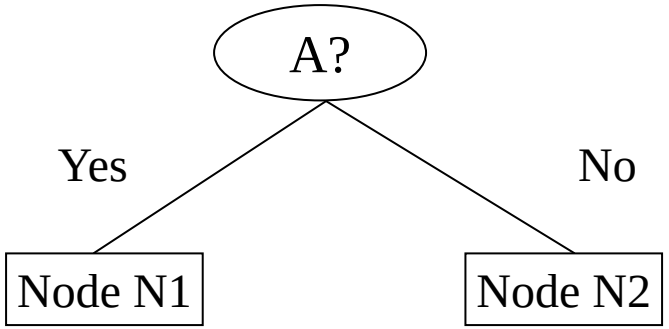
Gini(N2)
= 1 – (4/7)² – (3/7)²
= 0.489

	N1	N2
C1	3	4
C2	0	3
Gini=0.342		

Gini(Hijos)
= 3/10 * 0
+ 7/10 * 0.489
= 0.342

Gini mejora pero el error permanece igual!!

Error de Clasificación vs. Índice de GINI



	Parent
C1	7
C2	3
Gini = 0.42	

	N1	N2
C1	3	4
C2	0	3
Gini=0.342		

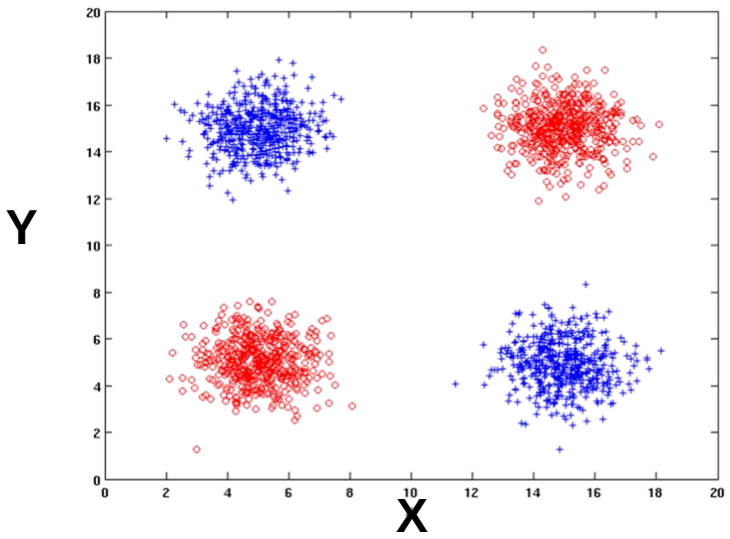
	N1	N2
C1	3	4
C2	1	2
Gini=0.416		

El error de clasificacion en los tres casos = 0.3 !

Clasificación basada en arboles de decisión

- Ventajas:
  - Poco costosos para armar
  - Extremadamente rápidos para clasificar registros no vistos.
  - Fáciles de interpretar para arboles pequeños
  - Robustos con relación al ruido (especialmente cuando se utilizan métodos para evitar el sobre-ajuste)
  - Pueden manejar fácilmente atributos redundantes o irrelevantes (siempre que los mismos no interactúen)
- Desventajas:
  - El espacio de arboles de decisión posibles es exponencialmente grande. Las aproximaciones voraces son muchas veces incapaces de encontrar el mejor árbol.
  - No tiene en cuenta interacciones entre los atributos
  - Cada frontera de decisión involucra únicamente a un solo atributo.

Interacciones…



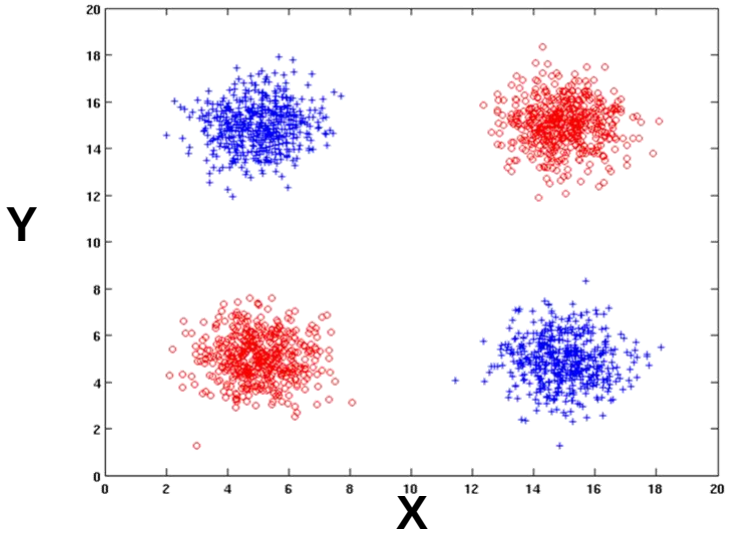
+ : 1000 instances

o : 1000 instances

Entropy (X) : 0.99

Entropy (Y) : 0.99

Interacciones



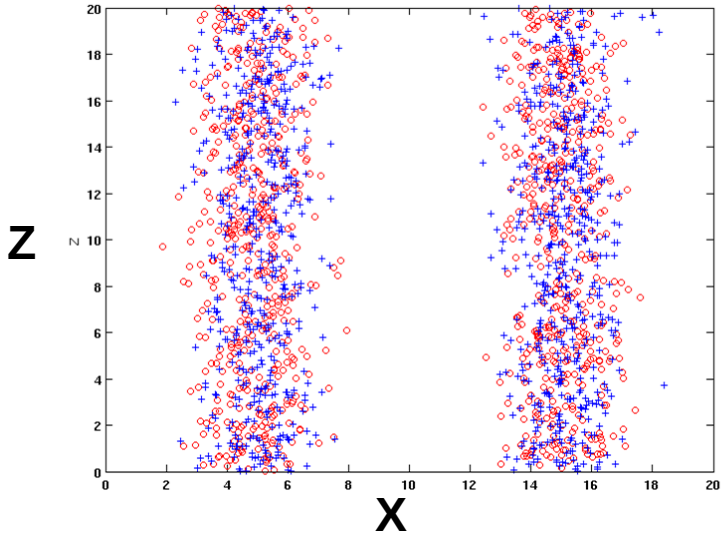
+ : 1000 instances

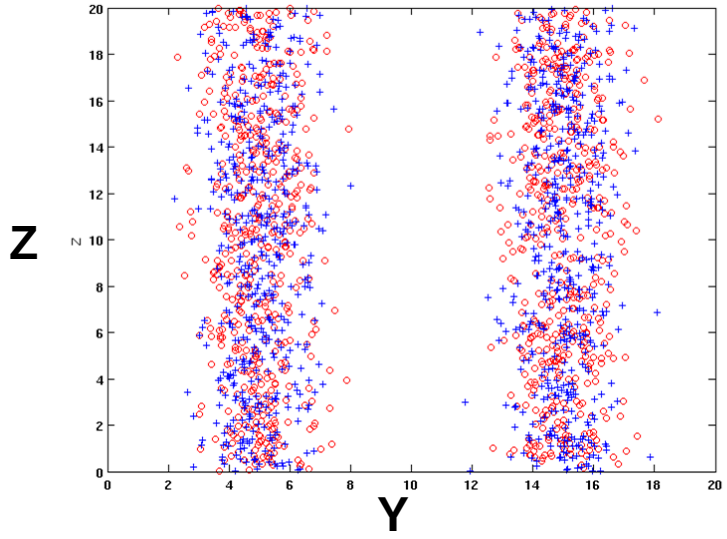
o : 1000 instances

Si se agrega Z como un atributo Ruidoso generado a partir de una distribucion uniforme

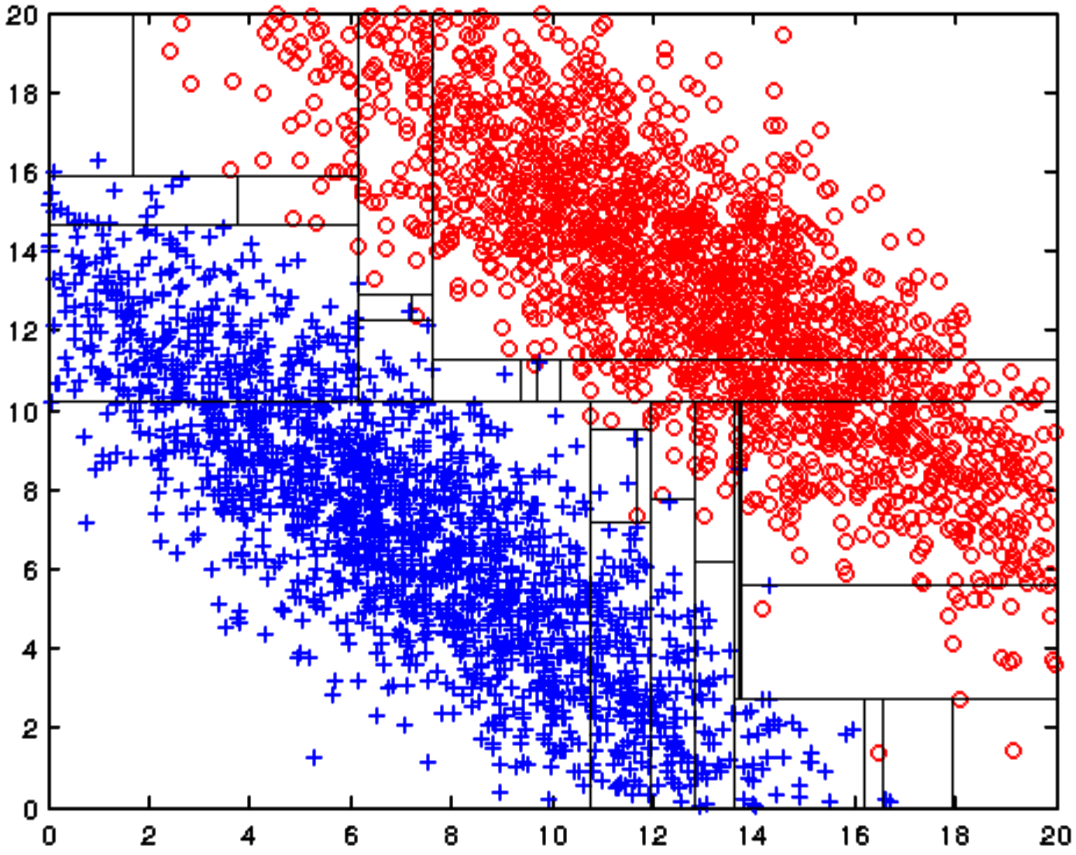
Entropy (X) : 0.99
Entropy (Y) : 0.99
Entropy (Z) : 0.98

El atributo Z seria elegido para el split!!!





Limitaciones de fronteras de decisión basadas en un único atributo



Tanto las clases positiva (+) como negativa (o) fueron generadas a partir de una distribución Gaussiana sesgada con centros en (8,8) y (12,12) respectivamente.