Árboles

Organización de datos

Ing. Juan M. Rodríguez

ID3

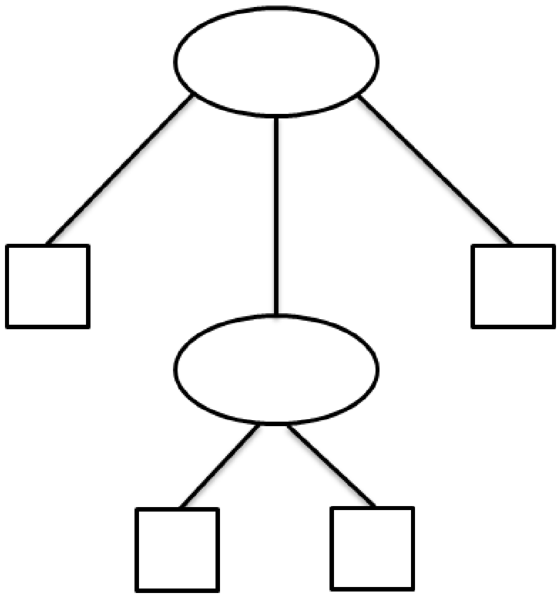
ID3

- **ID3**: Iterative Dichotomiser 3 (tree -> árbol)
- Creado por: **Ross Quinlan**
- Genera un árbol de decisión a partir de un conjunto de ejemplos.

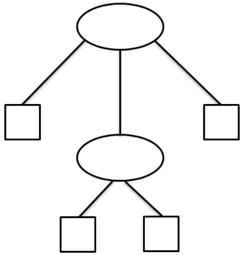
ID3

Representación de un Árbol de decisión

La salida del algoritmo ID3 se representa como un grafo en forma de árbol, cuyos componentes son:



ID3



- Un nodo principal llamado raíz en la parte superior

- Nodos terminales, como su nombre lo indica, son nodos donde termina el flujo y que ya no son raíz de ningún otro nodo. Estos nodos terminales deben contener una respuesta, o sea, la clasificación a que pertenece el objeto que ha conducido hasta él.
- Los demás nodos representan preguntas con respecto al valor de uno de los atributos.

- Las líneas representan las posibles respuestas que los atributos pueden tomar.

ID3 - Entropia (de la información)

La medida del desorden o la medida de la pureza. Básicamente, es la medida de la impureza o aleatoriedad en los datos.

ID3 - Entropia (de la información)

Entropía: Para calcular la entropía de **n** clases se utiliza la fórmula:

$$\text{Entropia}(S) = \sum_{i=1}^{n} -p_i \log_2 p_i$$

ID3 - Entropia:

Entropía: Para calcular la entropía de **n** clases se utiliza la fórmula:

$$\text{Entropia}(S) = \sum_{i=1}^{n} -p_i \log_2 p_i$$

Dónde:

- S: es una lista de valores posibles.
- Pi: es la probabilidad de los valores.
- i: Cada uno de los valores.

ID3 - Entropia:

Entropía: Para calcular la entropía de **n** clases se utiliza la fórmula:

$$\mathrm{Entropia}(S) = \sum_{i=1}^{n} -p_i \log_2 p_i$$

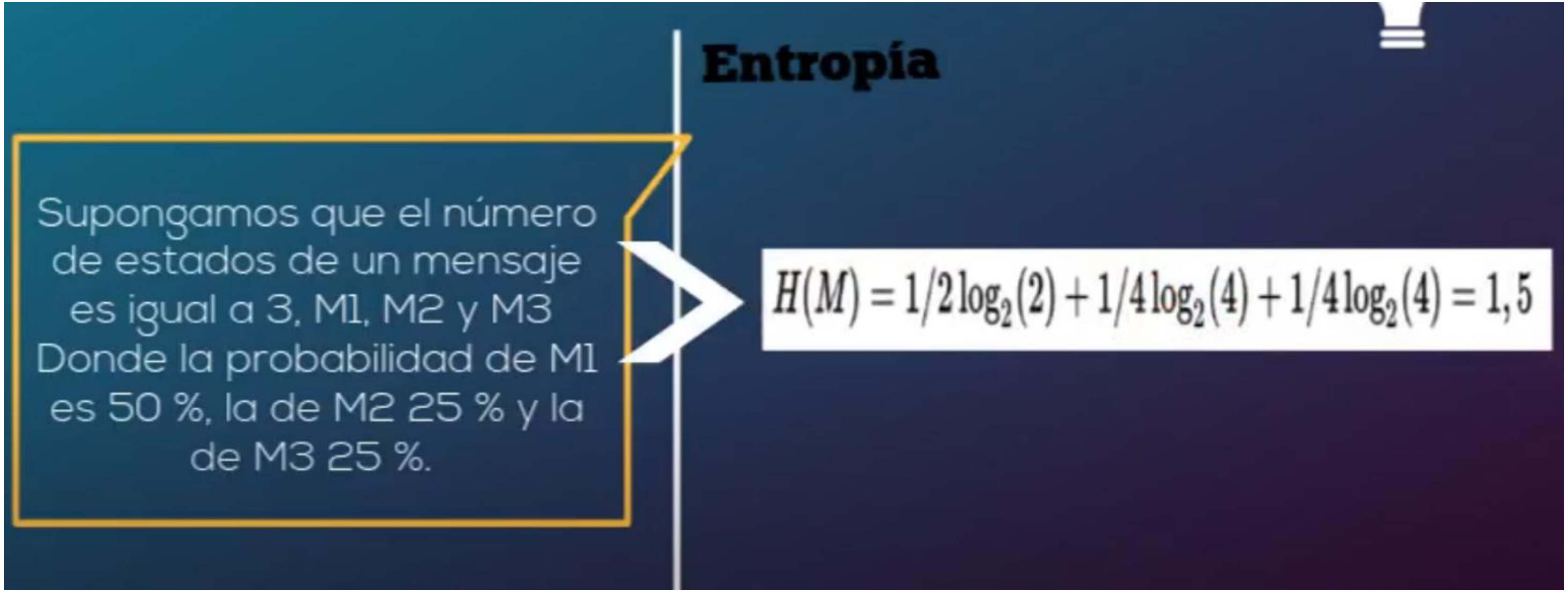
Importante

- Para una muestra homogénea la entropía es igual a 0

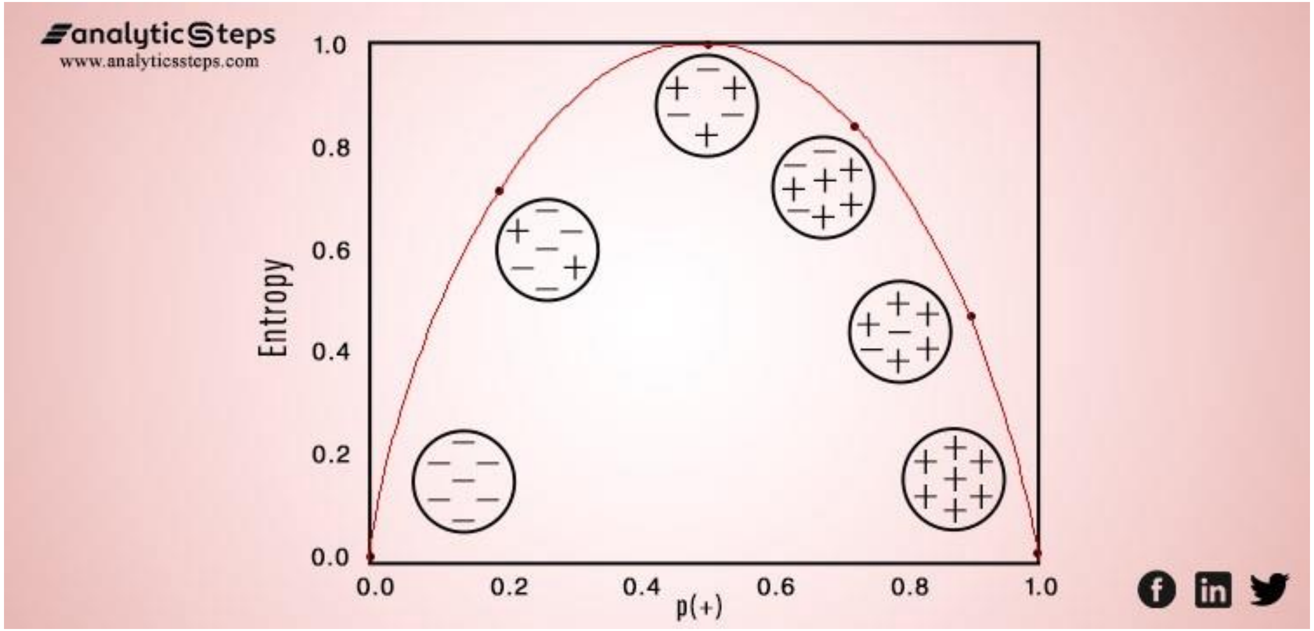
- La máxima entropía viene dada por Log2(n), n son los posibles valores de salida.
  Si n =2 (TRUE o FALSE) entonces, la máxima entropía es 1.
  O sea es la máxima incertidumbre

ID3 - Entropia: Ejemplo

ID3 - Entropia: Ejemplo



ID3 - Entropia: Otro ejemplo



ID3 - Ganancia de información:

La ganancia de información se aplica para cuantificar **qué característica,** de un conjunto de datos dados, **proporciona la máxima información** sobre la clasificación.

Si tuviésemos que elegir una sola característica, para clasificar. ¿Cuál sería?

ID3 - Ganancia de información:

Ganancia de información:

$$\text{Gan Inf}(S, A) = \text{Entropia}(S) - \sum_{v \in V(A)} \frac{|Sv|}{|S|} \text{Entropia}(Sv)$$

Dónde:

- **S**: es una lista de valores posibles para un atributo dado: A
- **A**: Uno de los atributos, en la lista de ejemplo.
- **V(A)** : Conjunto de valores que A puede tomar
- **Sv/S** = Probabilidad de un valor, para el atributo A.
- **Entropía (Sv),** entropía calculada para el valor “v” de A.

ID3 - Algoritmo básico:

1. Calcular la **entropía** para todas las clases.

2. Calcular la entropía para cada valor posible de cada atributo.

3. Seleccionar el mejor atributo basado en la reducción de la entropía. Usando el cálculo de **Ganancia de la Información**

4. **Iterar**, para cada sub-nodo. Excluyendo el nodo raíz, que ya fue usado.

ID3: Ejemplo

Animal	ATRIBUTO			Clase
	Vuela	Piel	Nacimiento	
vaca	no	pelo	placenta	Mamifero
cerdo	no	pelo	placenta	Mamifero
paloma	sí	pluma	huevo	Ave
murcielago	sí	pelo	placenta	Mamifero
gallina	no	pluma	huevo	Ave
iguana	no	escamas	huevo	Reptil
cocodrilo	no	escamas	huevo	Reptil

ID3: Ejemplo

Probabilidad de cada clase (en el conjunto de entrenamiento)

Animal	ATRIBUTO			Clase
	Vuela	Piel	Nacimiento	
vaca	no	pelo	placenta	Mamifero
cerdo	no	pelo	placenta	Mamifero
paloma	sí	pluma	huevo	Ave
murcielago	sí	pelo	placenta	Mamifero
gallina	no	pluma	huevo	Ave
iguana	no	escamas	huevo	Reptil
cocodrilo	no	escamas	huevo	Reptil

ID3: Ejemplo

Probabilidades de la clase		Probabilidad	"-P*Log2(P)"
Mamifero	3/7	0,43	0,52
Ave	2/7	0,29	0,52
Reptil	2/7	0,29	0,52

$$\text{Entropia}(S) = \sum_{i=1}^{n} -p_i \log_2 p_i$$

= 1,56

ID3: Ejemplo - Atributo Vuela = NO

Animal	ATRIBUTO			Clase
	Vuela	Piel	Nacimiento	
vaca	no	pelo	placenta	Mamifero
cerdo	no	pelo	placenta	Mamifero
gallina	no	pluma	huevo	Ave
iguana	no	escamas	huevo	Reptil
cocodrilo	no	escamas	huevo	Reptil

ID3: Ejemplo

5/7 (5 animales no vuelan)

Probabilidades del atributo Vuela NO				0,71
Mamifero	2/5	0,40	0,53	
Ave	1/5	0,20	0,46	
Reptil	2/5	0,40	0,53	
			1,52	**Entropia(S,A)**
Probabilidades del atributo Vuela Sí				0,29
Mamifero	1/2	0,50	0,50	
Ave	1/2	0,50	0,50	
Reptil	0	0,00	0,00	
			1,00	**Entropia(S,A)**

ID3: Ejemplo

$$-p_i \log_2 p_i$$

Probabilidades del atributo Vuela NO				0,71
Mamifero	2/5	0,40	0,53	
Ave	1/5	0,20	0,46	
Reptil	2/5	0,40	0,53	
			1,52	**Entropia(S,A)**
Probabilidades del atributo Vuela Sí				0,29
Mamifero	1/2	0,50	0,50	
Ave	1/2	0,50	0,50	
Reptil	0	0,00	0,00	
			1,00	**Entropia(S,A)**

ID3: Ejemplo

Probabilidades del atributo Piel: PLUMA				
Mamifero	0/2	0,00	0,00	
Ave	2/2	1,00	0,00	
Reptil	0/2	0,00	0,00	
			0,00	**Entropia(S,A)**
Probabilidades del atributo Piel: PELO				
Mamifero	3/3	1,00	0,00	
Ave	0	0,00	0,00	
Reptil	0	0,00	0,00	
			0,00	**Entropia(S,A)**
Probabilidades del atributo Piel: Escamas				
Mamifero	0/2	0,00	0,00	
Ave	0/2	0,00	0,00	
Reptil	2/2	1,00	0,00	
			0,00	**Entropia(S,A)**

ID3: Ejemplo

Probabilidades del atributo Piel: PLUMA				0,29
Mamifero	0/2	0,00	0,00	
Ave	2/2	1,00	0,00	
Reptil	0/2	0,00	0,00	
			0,00	**Entropia(S,A)**
Probabilidades del atributo Piel: PELO				0,43
Mamifero	3/3	1,00	0,00	
Ave	0	0,00	0,00	
Reptil	0	0,00	0,00	
			0,00	**Entropia(S,A)**
Probabilidades del atributo Piel: Escamas				0,29
Mamifero	0/2	0,00	0,00	
Ave	0/2	0,00	0,00	
Reptil	2/2	1,00	0,00	
			0,00	**Entropia(S,A)**

ID3: Ejemplo

Probabilidades del atributo Nacimiento: Huevo				
Mamifero	0	0,00		
Ave	2/4	0,50		
Reptil	2/4	0,50		
				Entropia(S,A)
Probabilidades del atributo Nacimiento: Placenta				
Mamifero	3/3	1,00		
Ave	0	0,00		
Reptil	0	0,00		
			0,00	**Entropia(S,A)**

ID3: Ejemplo

Probabilidades del atributo Nacimiento: Huevo				0,57
Mamifero	0	0,00	0,00	
Ave	2/4	0,50	0,50	
Reptil	2/4	0,50	0,50	
			1,00	**Entropia(S,A)**
Probabilidades del atributo Nacimiento: Placenta				0,43
Mamifero	3/3	1,00	0,00	
Ave	0	0,00	0,00	
Reptil	0	0,00	0,00	
			0,00	**Entropia(S,A)**

ID3: Ejemplo

1.56

$$\text{Gan Inf}(S, A) = \text{Entropia}(S) - \sum_{v \in V(A)} \frac{|Sv|}{|S|} \boxed{\text{Entropia}(Sv)}$$

= 1.56 - (0.29*1 + 0.71 * 1.52) = 0.1908

Sí p=0.29 E=1

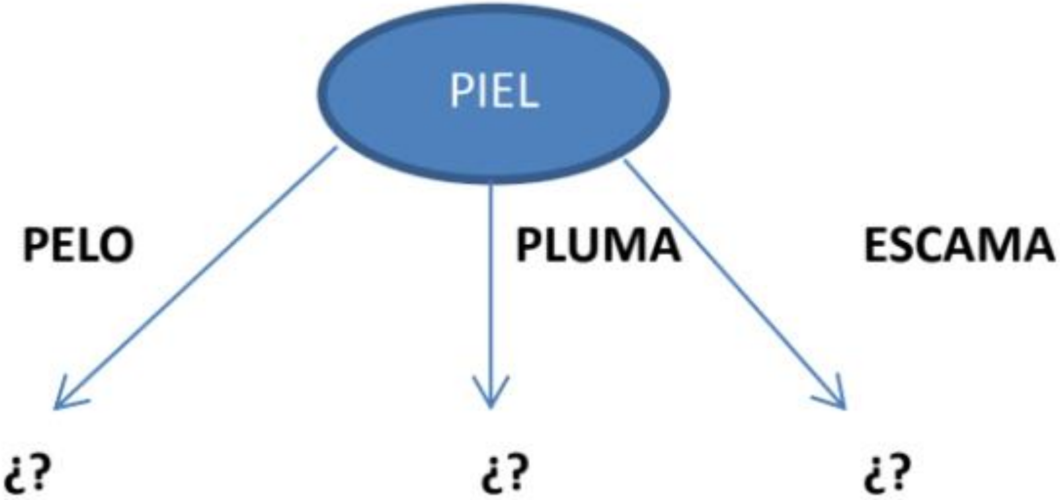
Atributo: vuela

No p=0.71 E=1.52

ID3: Ejemplo

Atributos	Ganancia de información
Vuela	0.191
Piel	1.557
Nacimiento	0.985

ID3: Ejemplo



ID3: Ejemplo

Animal	ATRIBUTO			Clase
	Vuela	Piel	Nacimiento	
vaca	no	pelo	placenta	Mamifero
cerdo	no	pelo	placenta	Mamifero
~~paloma~~	~~sí~~	~~pluma~~	~~huevo~~	~~Ave~~
murcielago	sí	pelo	placenta	Mamifero
~~gallina~~	~~no~~	~~pluma~~	~~huevo~~	~~Ave~~
~~iguana~~	~~no~~	~~escamas~~	~~huevo~~	~~Reptil~~
~~cocodrilo~~	~~no~~	~~escamas~~	~~huevo~~	~~Reptil~~

ID3: Ejemplo

Probabilidades de la clase PELO		Probabilidad	"-P*Log2(P)"
Mamifero	3/3	1	0
Ave	0/3	0	0
Reptil	0/3	0	0

Entropía de la clase PELO: 0

ID3: Ejemplo

Probabilidades de la clase PLUMA		Probabilidad	"-P*Log2(P)"
Mamifero	0/2	0	0
Ave	2/2	1	0
Reptil	0/2	0	0

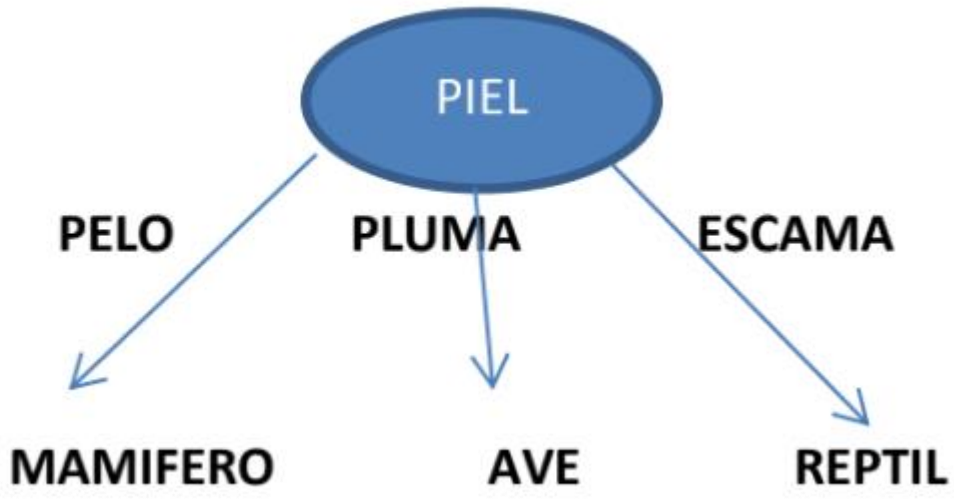
Entropía de la clase PLUMA: 0

ID3: Ejemplo

Probabilidades de la clase ESCAMA		Probabilidad	"-P*Log2(P)"
Mamifero	0/2	0	0
Ave	0/2	0	0
Reptil	2/2	1	0

Entropía de la clase ESCAMA: 0

ID3: Ejemplo



ID3: Recapitulando

- Un nodo de decisión está asociado a uno de los atributos y tiene 2 o más ramas que salen de él, cada una de ellas representando los posibles valores que puede tomar el atributo asociado.

- Un nodo-respuesta está asociado a la clasificación que se quiere proporcionar,y nos devuelve la decisión del árbol con respecto al ejemplo de entrada.

Impureza de Gini

Impureza de Gini

La impureza de Gini es una medida de cuán a menudo un elemento elegido aleatoriamente del conjunto sería etiquetado incorrectamente si fue etiquetado de manera aleatoria de acuerdo a la distribución de las etiquetas en el subconjunto

Algunas implementaciones de árboles de decisión utilizan la **impureza de Gini** en lugar de la ganancia de información, ya que es más fácil de calcular (computacionalmente menos costosa)
Ejemplo: Scikit-learn

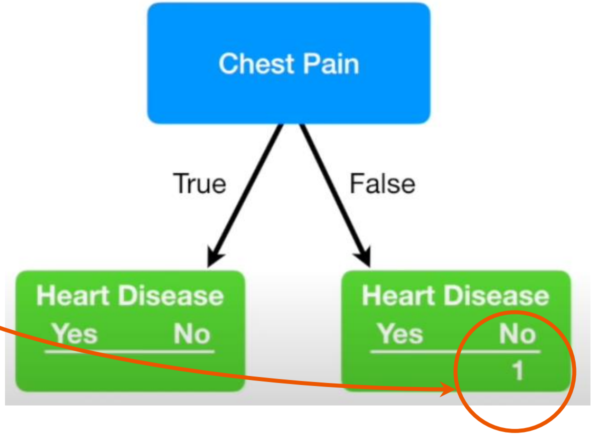
$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Impureza de Gini

Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc…	etc…	etc…	etc…

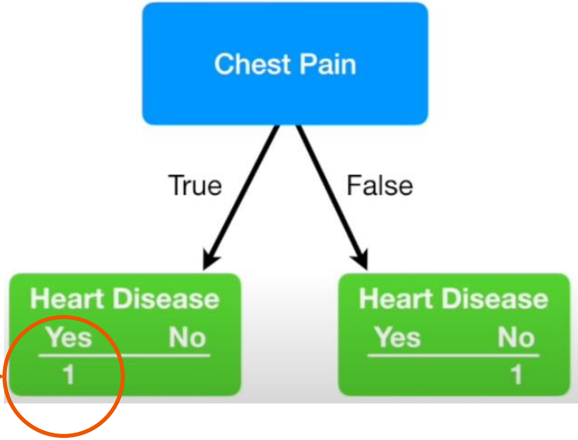
¿Cúal debería ser el mejor atributo para el nodo raíz?

- Chest Pain (dolor de pecho)
- Good Blood Circulation (buena circulación sanguínea)
- Blocked Arteries (arterias bloqueadas)



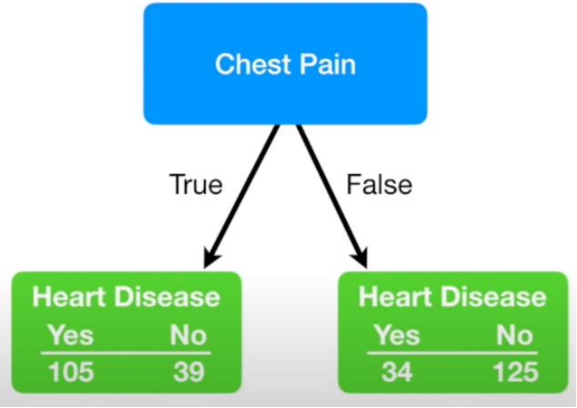
Impureza de Gini

Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc...	etc...	etc...	etc...



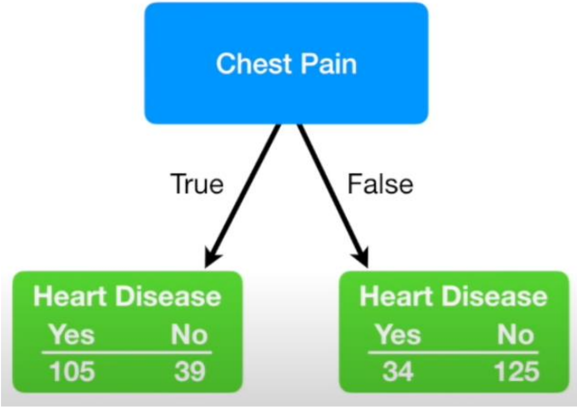
Impureza de Gini

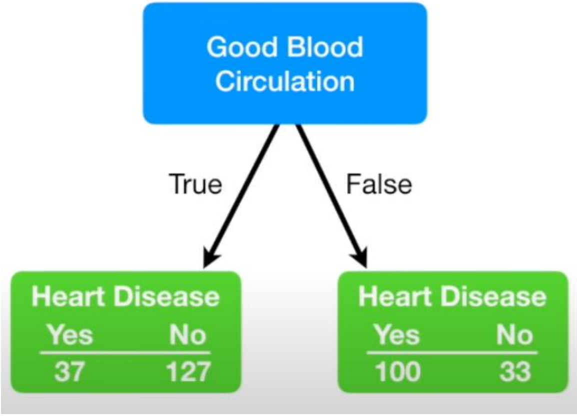
Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc...	etc...	etc...	etc...



Impureza de Gini

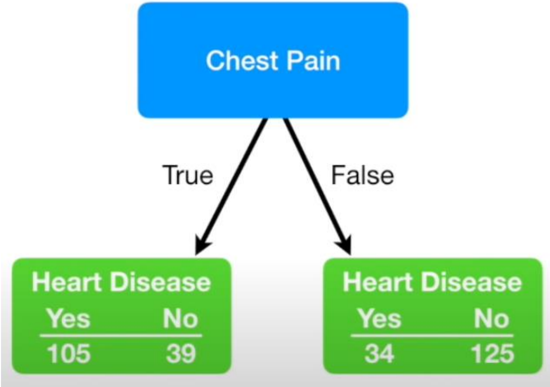
Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc...	etc...	etc...	etc...

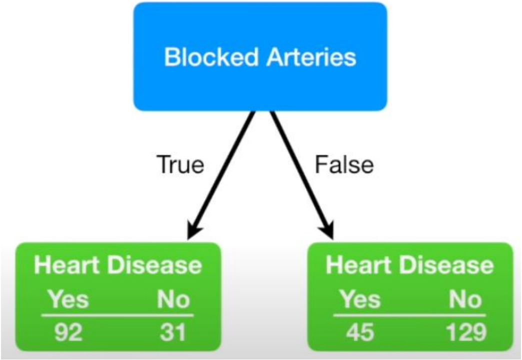


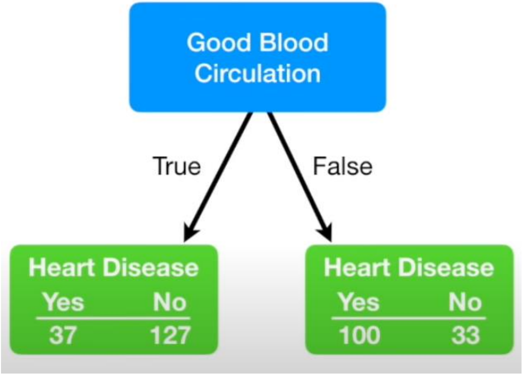


Impureza de Gini

Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc...	etc...	etc...	etc...



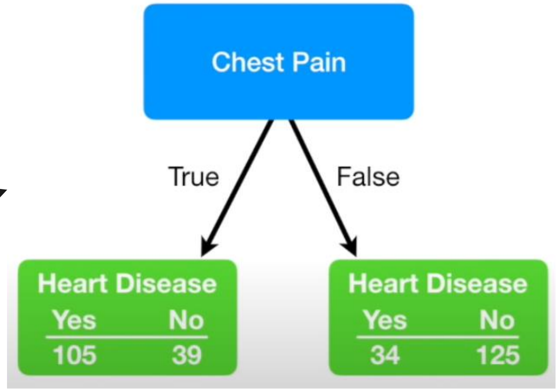


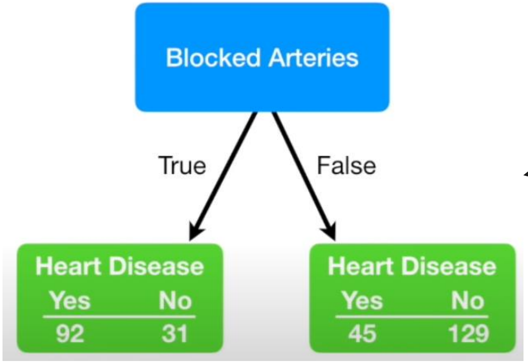


Impureza de Gini

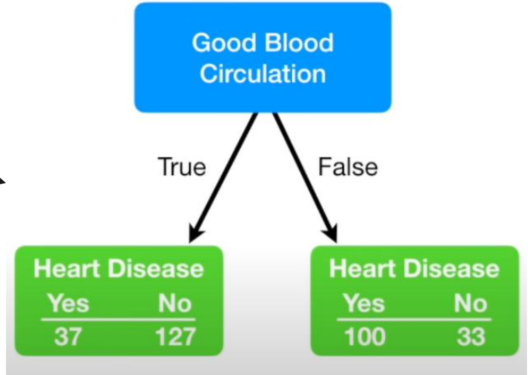
¿Qué también clasifica cada uno de estos árboles?

True, tiene la mayoría de “Yes”, y “False” la mayoría de “No”. Pero no clasifica correctamente al 100% de los casos



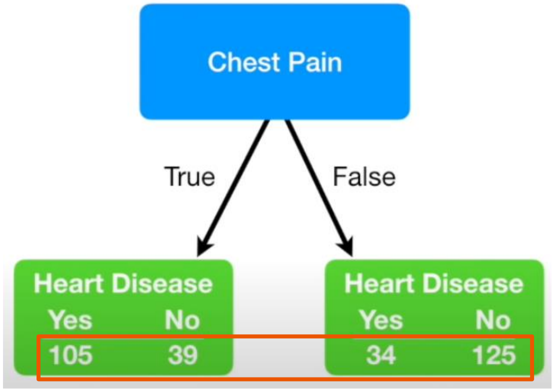


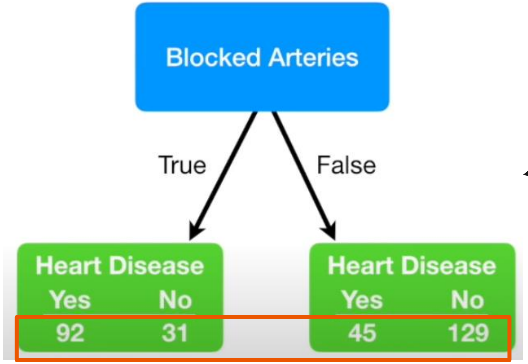
Estos árboles tampoco clasifican correctamente al 100% de los casos. ¿Cúal es mejor?



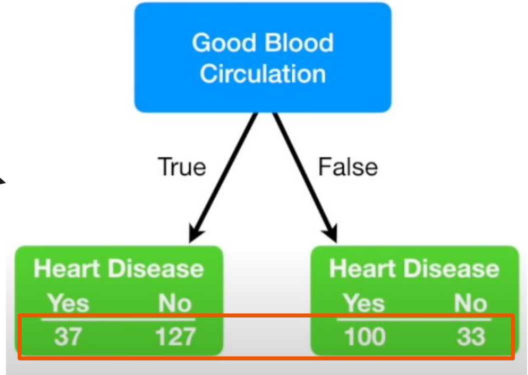
Impureza de Gini

Como ninguno de los nodos hoja es 100%: “Yes Heart Disease” o bien 100% “No Heart Disease”. Ninguno es puro, es decir que son nodos: **impuros**



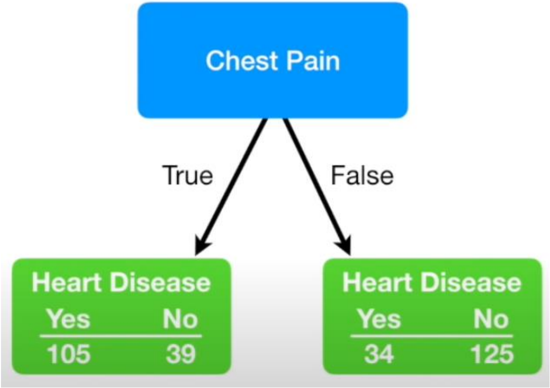


El total no siempre coincide, ya que para valores faltantes (nulos), se omite el registro



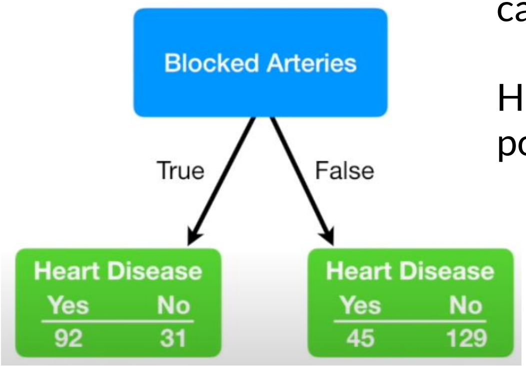
Impureza de Gini

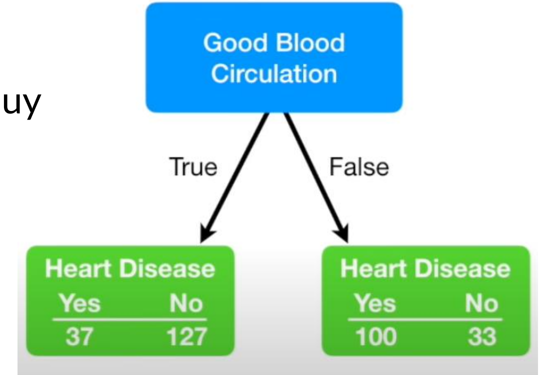
Como ninguno de los nodos hoja es 100%: “Yes Heart Disease” o bien 100% “No Heart Disease”. Ninguno es puro, es decir que son nodos: **impuros**



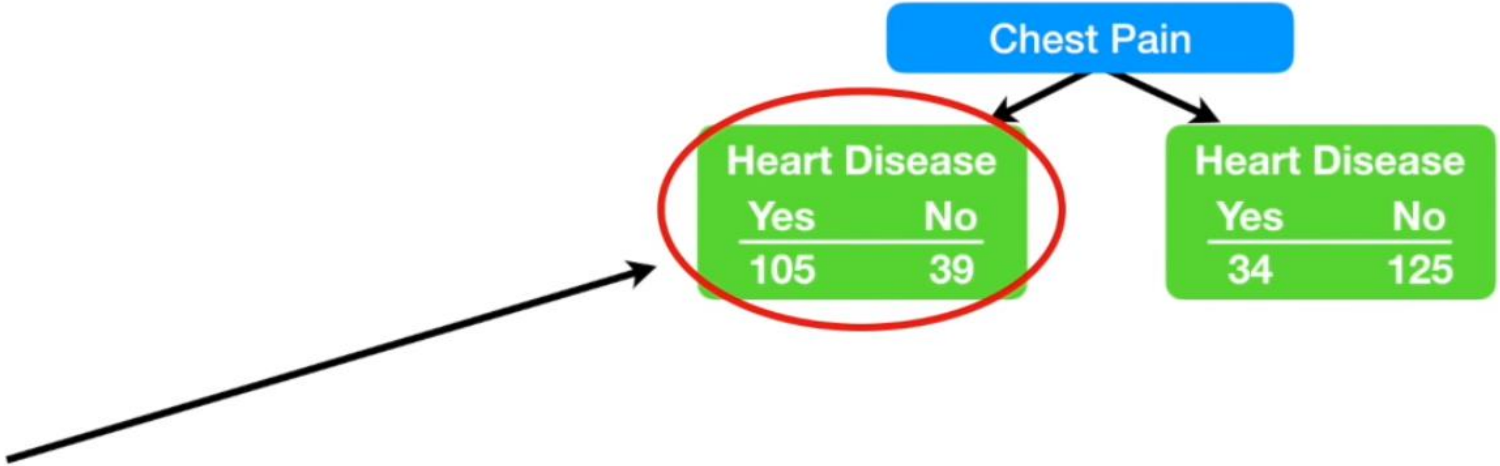
Para saber cual clasifica mejor, tenemos que medir la impureza de cada nodo y de cada árbol.

Hay varias formas de medir esto, una muy popular es llamada: **Gini**



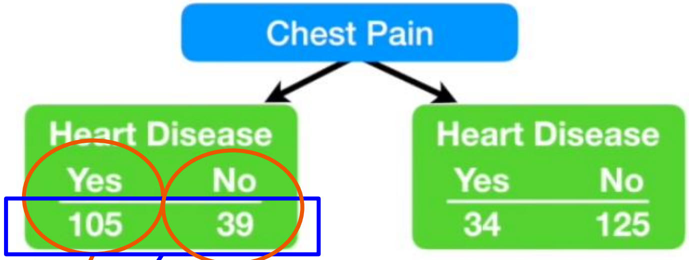


Impureza de Gini



For this leaf, the Gini impurity $= 1 -$ (the probability of "yes")^2 - (the probability of "no")^2

Impureza de Gini

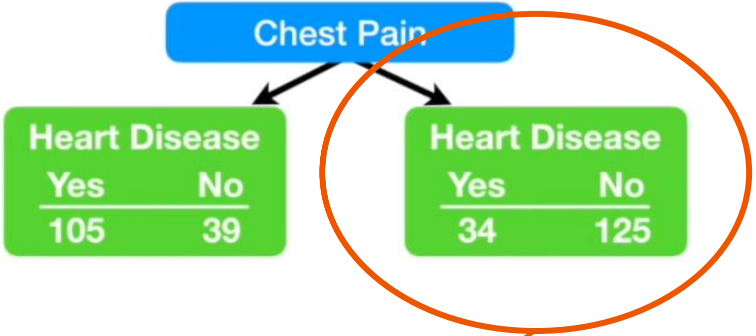


For this leaf, the Gini impurity = 1 - (the probability of "yes")^2 - (the probability of "no")^2

$$= 1 - \left(\frac{105}{\boxed{105 + 39}}\right)^2 - \left(\frac{39}{105 + 39}\right)^2$$

$$= 0.395$$

Impureza de Gini



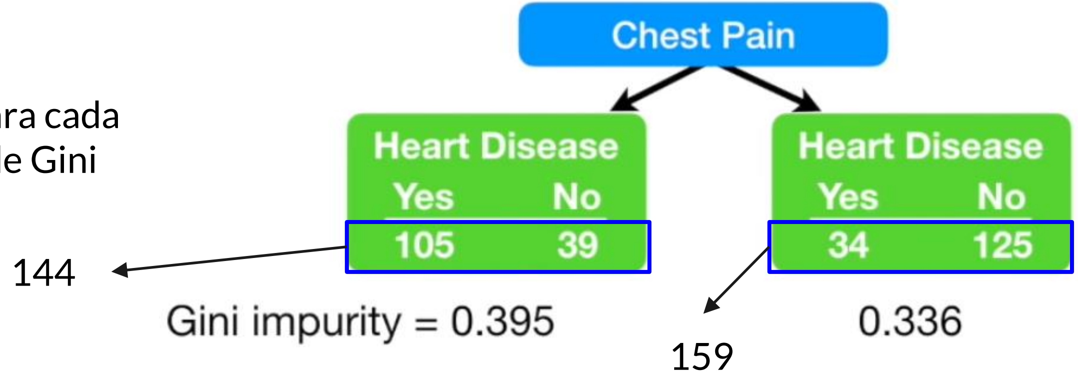
$$= 1 - (\text{the probability of “yes”})^2 - (\text{the probability of “no”})^2$$

$$= 1 - \left(\frac{34}{34 + 125}\right)^2 - \left(\frac{125}{34 + 125}\right)^2$$

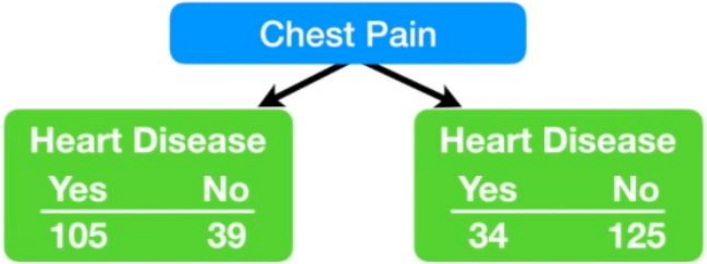
$$= 0.336$$

Impureza de Gini

Ahora que tenemos la impureza de Gini para cada nodo hoja, podemos calcular la impureza de Gini total, para el nodo raíz: “Chest Pain”.
Se trata de el promedio ponderado.



Impureza de Gini



Gini impurity = 0.395 0.336

Gini impurity for Chest Pain = weighted average of Gini impurities for the leaf nodes

$$= \left(\frac{144}{144 + 159}\right) 0.395 + \left(\frac{159}{144 + 159}\right) 0.336$$

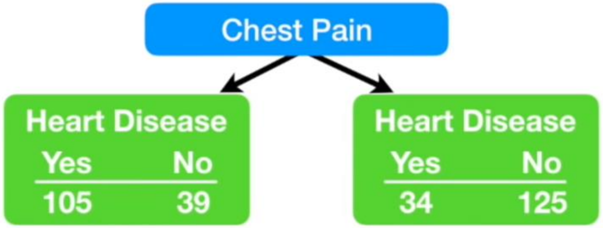
$$= 0.364$$

Impureza de Gini

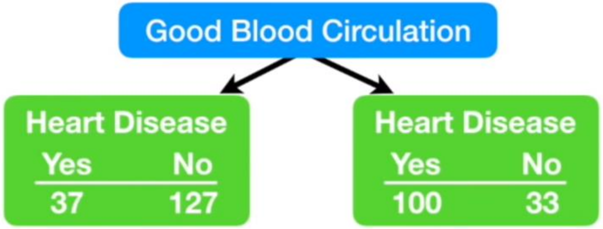
Cómo “Good blood circulation” tiene la menor impureza, utilizamos este atributo para el nodo raíz.

Luego procedemos de igual forma que en el caso de la ganancias de información pero calculando la impureza de Gini

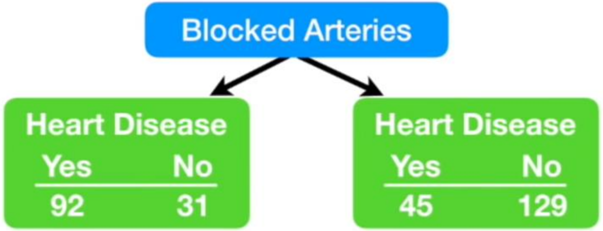
Gini impurity for Chest Pain = 0.364



Gini impurity for Good Blood Circulation = 0.360



Gini impurity for Blocked Arteries = 0.381



C4.5

C4.5

Se hicieron mejoras a ID3

- Campos numéricos, rangos continuous
- Datos faltantes
- Poda

C4.5: Datos faltantes

Datos faltantes:

Manejo de los datos de formación con valores de atributos faltantes - C4.5 permite valores de los atributos para ser marcado como “?” para faltantes. Los valores faltantes de los atributos simplemente no se usan en los cálculos de la ganancia y la entropía.

C4.5: Campos numéricos, o rangos continuous

Campos numéricos, o rangos continuous:

Si un atributo A, tiene un rango continuo de valores. El algoritmo puede, dinámicamente crear un campo **Booleano** tal que si A < C **Ac = TRUE**, sino **Ac = FALSE.**

¿Cómo encontrar ese umbral C?

Vamos a cortar el rango de forma que **C** nos quede con la mayor ganancia de información.

C4.5: Campos numéricos, o rangos continuous

Ordenamos **A** de menor a mayor, por ejemplo.

Identificamos los valores adyacentes (de la clase que es nuestra salida)

Detectamos cuando hay un cambio de valor de salida, entonces en esos límites seguramente están nuestros **C**i candidatos.

Creamos varios **Ci**, que dividen en dos el rango. Para cada uno de estos rangos calculamos la ganancia de información. Nos quedamos con el que nos

da el mejor resultado. (Se puede también quedarse con los N mejor.)

C4.5: Poda

El método de poda del árbol consiste en:

- generar el árbol tal y como hemos visto
- A continuación, analizar recursivamente, y desde las hojas, qué preguntas (nodos interiores) se pueden eliminar sin que se incremente el error de clasificación con el conjunto de test.

(Si hay ruido el árbol tendrá un error, es decir cantidad de casos mal clasificados)

Error= Casos bien clasificados / Casos Totales

C4.5: Poda

El método de poda del árbol consiste en:

- Se elimina un nodo interior cuyos sucesores son todos nodos hoja.
- Se vuelve a calcular el error que se comete con este nuevo árbol sobre el conjunto de test.
- Si este error es menor que el error anterior, entonces se elimina el nodo y todos sus sucesores(hojas)
- Se repite.

Random Forest

Random Forest

“Muchos estimadores mediocres, promediados pueden ser muy buenos”

Bootstrap aggregating:

Es una técnica, o meta-algoritmo que dice lo siguiente:

Dado un conjunto de entrenamiento D, de tamaño n, la técnica de ***bagging*** generará **m** nuevos conjuntos de entrenamiento D1,... Di,..., Dm cada uno de tamaño **n'** tomando muestras aleatorias de D.

Y en general **n'<n**. Siendo **n'** aproximadamente un **2/3** de n.

Bootstrap aggregating:

Suponiendo que tenemos un conjunto de datos, como el que sigue:

- 8 Atributos
- 1 Clase que queremos saber
- 9000 registros

Bootstrap aggregating:

9000

F-1	F-2	F-3	F-4	F-5	F-6	F-7	F-8	CLASS
Fa1	Fb1	Fc1	Fd1	Fe1	Ff1	Fg1	Fh1	Class-A
Fa2	Fb2	Fc2	Fd2	Fe2	Ff2	Fg2	Fh2	Class-A
Fa3	Fb3	Fc3	Fd3	Fe3	Ff3	Fg3	Fh3	Class-B
Fa4	Fb4	Fc4	Fd4	Fe4	Ff4	Fg4	Fh4	Class-B
Fa5	Fb5	Fc5	Fd5	Fe5	Ff5	Fg5	Fh5	Class-B
Fa6	Fb6	Fc6	Fd6	Fe6	Ff6	Fg6	Fh6	Class-A
Fa7	Fb7	Fc7	Fd7	Fe7	Ff7	Fg7	Fh7	Class-B
Fa8	Fb8	Fc8	Fd8	Fe8	Ff8	Fg8	Fh8	Class-A
Fa9	Fb9	Fc9	Fd9	Fe9	Ff9	Fg9	Fh9	Class-A
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Fai	Fbi	Fci	Fdi	Fei	Ffi	Fgi	Fhi	Class-B
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Fa9000	Fb9000	Fc9000	Fd9000	Fe9000	Ff9000	Fg9000	Fh9000	Class-A

Bootstrap aggregating:

Hacemos **m** sub-tablas tomando solo algunas filas, de forma aleatoria, ¿cuantas tomamos? => 2/3 es decir 6000

m

6000

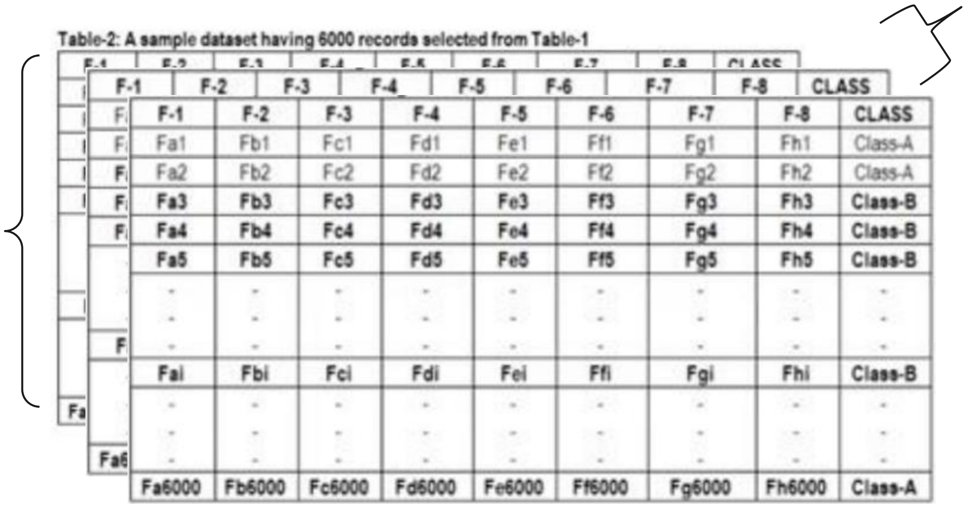


Table-2: A sample dataset having 6000 records selected from Table-1

F-1	F-2	F-3	F-4	F-5	F-6	F-7	F-8	CLASS
Fa1	Fb1	Fc1	Fd1	Fe1	Ff1	Fg1	Fh1	Class-A
Fa2	Fb2	Fc2	Fd2	Fe2	Ff2	Fg2	Fh2	Class-A
Fa3	Fb3	Fc3	Fd3	Fe3	Ff3	Fg3	Fh3	Class-B
Fa4	Fb4	Fc4	Fd4	Fe4	Ff4	Fg4	Fh4	Class-B
Fa5	Fb5	Fc5	Fd5	Fe5	Ff5	Fg5	Fh5	Class-B
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Fai	Fbi	Fci	Fdi	Fei	Ffi	Fgi	Fhi	Class-B
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Fa6000	Fb6000	Fc6000	Fd6000	Fe6000	Ff6000	Fg6000	Fh6000	Class-A

Bootstrap aggregating:

Attribute bagging (o ***random subspace***):

Luego para cada una de las **m** tablas, escogemos sólo algunos atributos (**COLUMNAS**) de forma aleatoria también.

¿Con cuantas nos quedamos? =>Con **RAÍZ CUADRADA** del número de atributos.

Cómo acá hay 8 atributos, tomamos **Round($\sqrt{8}$) = 3**

(Esto es recomendable sobre todo cuando hay muchas columnas)

Bootstrap aggregating:

Attribute bagging (o ***random subspace***):

Table-3: A sample dataset having 6000 records, 3-features, selected from Table-2

F-1	F-5	F-7	CLASS
Fa1	Fe1	Fg1	Class-A
Fa2	Fe2	Fg2	Class-A
Fa3	Fe3	Fg3	Class-B
Fa4	Fe4	Fg4	Class-B
Fa5	Fe5	Fg5	Class-B
. . .	. . .	. . .	. . .
Fai	Fei	Fgi	Class-B
. . .	. . .	. . .	. . .
Fa6000	Fe6000	Fg6000	Class-A

Random Forest:

*Ahora tenemos **m** tablas reducidas en atributos y para cada una de ellas entrenamos un árbol*

- *Para cada árbol calculamos su matriz de confusión:*

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Random Forest:

Entonces calculamos la tasa de error de cada árbol, esto es:
FALSOS POSITIVOS + FALSOS NEGATIVOS SOBRE EL TOTAL DE EJEMPLOS CLASIFICADOS
(Sobre un conjunto de entrenamiento o sobre un porcentaje del conjunto utilizado)

- Nos quedamos con el mejor árbol de los **m**, y repetimos el proceso K veces.
- Al final vamos a tener K árboles.

(FP + FN) / TOTAL = **Tasa de error**

Random Forest:

¿Cómo clasificar una vez finalizado el proceso?

Luego para clasificar un nuevo ejemplo hacemos lo siguiente:

Ejecutamos el caso que queremos probar por cada uno de los **K** arboles
Si la mayoría de los casos devolvió que la categoría final es **CATEGORIA-A**, entonces nos quedamos con ese valor de salida.

Random Forest: