03008915 C/ Ferrocaril, 22, 03570 La Vila Joiosa **Tel** 966870140 **Fax** 966870141 http://portal.edu.gva.es/iesmarcoszaragoza

1 ML– ÁRBOLES DE DECISIÓN

Los problemas de tipo de **clasificación** generalmente son aquellos en los que intentamos predecir los valores de una variable dependiente categórica (clase, pertenencia a grupos, etc.) a partir de una o más variables predictoras continuas.

Por ejemplo, podemos estar interesados en predecir quién se graduará o no de la universidad, o quién renovará o no una suscripción. Estos serían ejemplos de problemas simples de clasificación binaria, donde la variable dependiente categórica solo puede asumir dos valores distintos y mutuamente excluyentes.

En otros casos, podríamos estar interesados en predecir cuál de los múltiples productos de consumo alternativos diferentes (por ejemplo, marcas de automóviles) decide comprar una persona, o qué tipo de falla ocurre con diferentes tipos de motores. En esos casos, existen múltiples categorías o clases para la variable dependiente categórica.

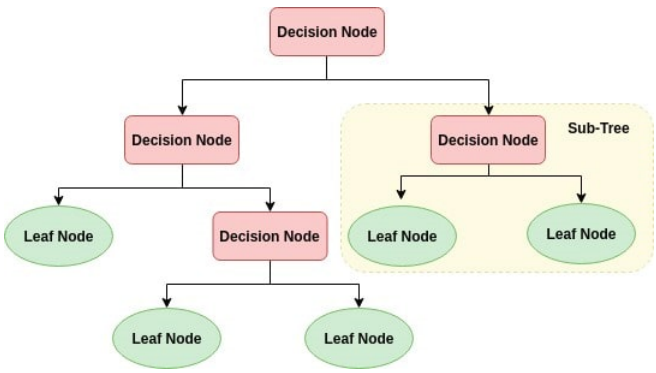
Un árbol de decisión en Machine Learning es una estructura de árbol similar a un diagrama de flujo donde un nodo interno representa una característica (o atributo), la rama representa una regla de decisión y cada nodo hoja representa el resultado.

El nodo superior en un árbol de decisión en Machine Learning se conoce como el nodo raíz. Aprende a particionar en función del valor del atributo. Divide el árbol de una manera recursiva llamada partición recursiva.

Esta estructura tipo diagrama de flujo lo ayuda a tomar decisiones. Es una visualización como un diagrama de flujo que imita fácilmente el pensamiento a nivel humano. Es por eso que los árboles de decisión son fáciles de entender e interpretar.

Los árboles de decisión clasifican los ejemplos clasificándolos por el árbol desde la raíz hasta algún nodo hoja, con el nodo hoja proporcionando la **clasificación** al ejemplo, este enfoque se llama **Enfoque de arriba hacia abajo**.

Cada nodo en el árbol actúa como un caso de prueba para algún atributo, y cada borde que desciende de ese nodo corresponde a una de las posibles respuestas al caso de prueba. Este proceso es recursivo y se repite para cada subárbol enraizado en los nuevos nodos.



03008915 C/ Ferrocaril, 22, 03570 La Vila Joiosa **Tel** 966870140 **Fax** 966870141 http://portal.edu.gva.es/iesmarcoszaragoza

La idea básica detrás de cualquier algoritmo de árbol de decisión es el siguiente:

1. Seleccionar el mejor atributo utilizando Medidas de selección de atributos (ASM) para dividir los registros.
2. Hacer que ese atributo sea un nodo de decisión y dividir el conjunto de datos en subconjuntos más pequeños. recursivamente para cada nodo hasta que una de las condiciones coincida:
   - Todas las tuplas pertenecen al mismo valor de atributo.
   - No quedan más atributos.
   - No hay más instancias.

1.1 Medidas de selección de atributos

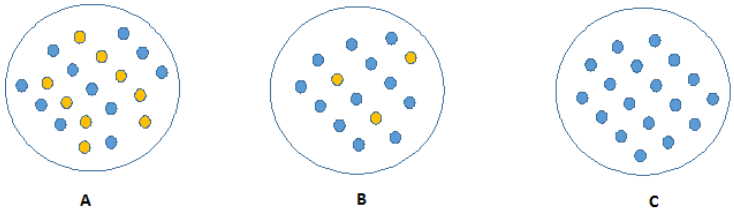
La medida de selección de atributos es una heurística para seleccionar el criterio de división que divide los datos de la mejor manera posible. También se conoce como reglas de división porque nos ayuda a determinar puntos de interrupción para tuplas en un nodo dado.

Las medidas de selección más populares son Ganancia de información, Proporción de ganancia e Índice de Gini.

1.1.1 Ganancia de información:

La ganancia de información es una propiedad estadística que mide qué tan bien un atributo dado separa los ejemplos de entrenamiento de acuerdo con su clasificación objetivo.

Si observamos la imagen, podemos decir que C es un nodo puro, B es menos impuro y A es más impuro.

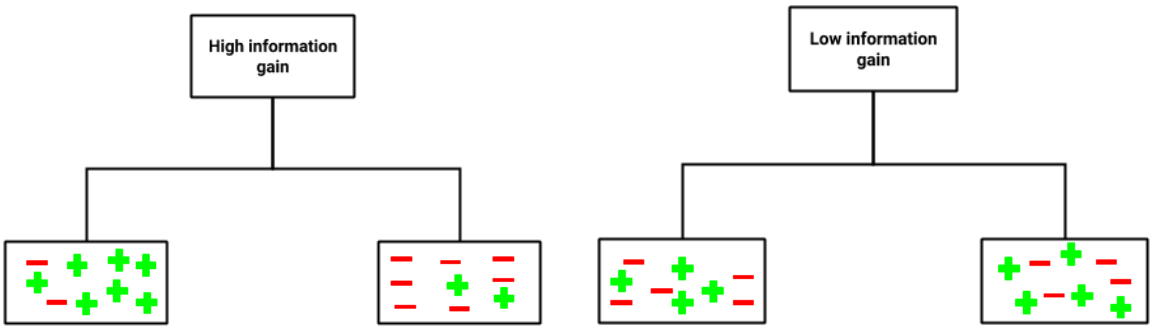


Ahora, podemos llegar a la conclusión de que un nodo menos impuro requiere menos información para describirlo. Y, el nodo más impuro requiere más información.

03008915 C/ Ferrocaril, 22, 03570 La Vila Joiosa **Tel** 966870140 **Fax** 966870141 http://portal.edu.gva.es/iesmarcoszaragoza

En otro ejemplo considerando la ganancia de información, en la figura a continuación, podemos ver que un atributo con baja ganancia de información (derecha) divide los datos de manera relativamente uniforme y como resultado no nos acerca más a una decisión.

Mientras que un atributo con alta ganancia de información (izquierda) divide los datos en grupos con un número desigual de positivos y negativos y, como resultado, ayuda a separarlos entre sí.

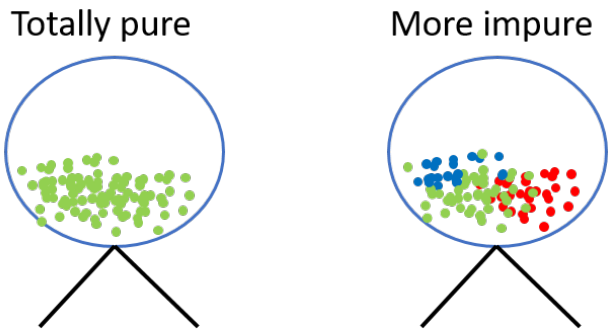


Para poder calcular la ganancia de información, primero tenemos que introducir el término ***entropía*** de un conjunto de datos.

Shannon inventó El concepto de entropía, que **mide la impureza del conjunto de entrada**. En física y matemáticas, la entropía se conoce como aleatoriedad o impureza en el sistema. En teoría de la información, se refiere a la impureza en un grupo de ejemplos. La ganancia de información es una disminución de la entropía.

La idea detrás de la entropía es, en términos simplificados, la siguiente: Imaginad que se tiene una rueda de lotería que incluye 100 bolas verdes. Se puede decir que el conjunto de bolas dentro de la rueda de lotería es totalmente puro porque solo se incluyen bolas verdes.

Para expresar esto en la terminología de entropía, este conjunto de bolas tiene una entropía de 0 (también podemos decir impureza cero). Considerar ahora, 30 de estas bolas son reemplazadas por bolas rojas y 20 por bolas azules.



03008915 C/ Ferrocaril, 22, 03570 La Vila Joiosa **Tel** 966870140 **Fax** 966870141 http://portal.edu.gva.es/iesmarcoszaragoza

Si ahora se saca otra bola de la rueda de lotería, la probabilidad de recibir una bola verde se ha reducido de 1.0 a 0.5. Como la impureza aumentó, la pureza disminuyó, por tanto, también aumentó la entropía.

Podemos decir que cuanto más "impuro" sea un conjunto de datos, mayor será la entropía y menos "impuro" un conjunto de datos, menor será la entropía

Se tendrá en cuenta que la entropía es 0 si todos los miembros de S pertenecen a la misma clase. Por ejemplo, si todos los miembros son positivos, *Entropía (S)* = 0.
La entropía es 1 cuando la muestra contiene el mismo número de ejemplos positivos y negativos. Si la muestra contiene un número desigual de ejemplos positivos y negativos, la entropía está entre 0 y 1.

1.2 Gini

Gini dice que, si seleccionamos dos elementos de una población al azar, entonces deben ser de la misma clase y la probabilidad de esto es 1 si la población es pura.

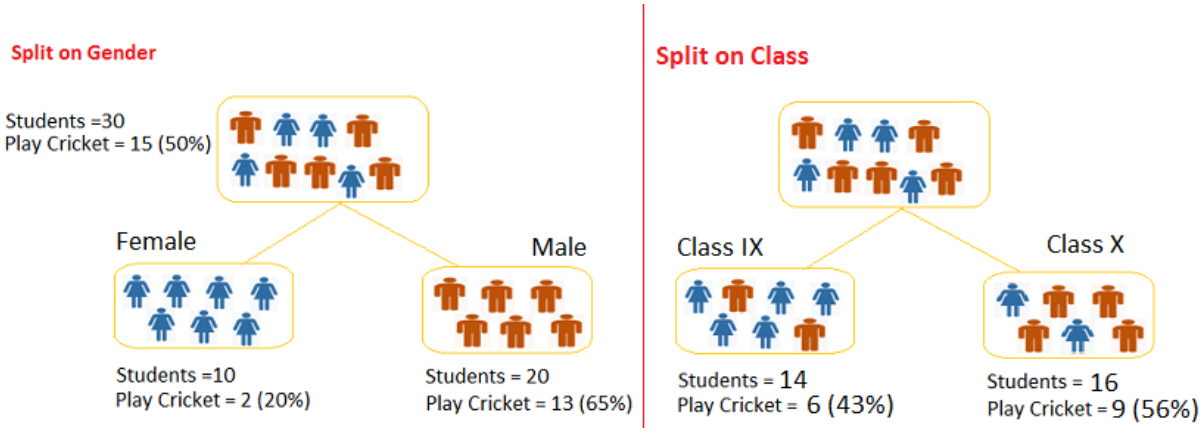
Funciona con la variable objetivo-categórica "Éxito" o "Fracaso".

Realiza solo divisiones binarias

Cuanto mayor sea el valor de Gini, mayor será la homogeneidad.

CART (árbol de **clasificación** y **regresión**) utiliza el método Gini para crear divisiones binarias.

Ejemplo:
Refiriéndose al ejemplo utilizado anteriormente, donde queremos segregar a los estudiantes en función de la variable objetivo (jugar cricket o no). En la siguiente instantánea, dividimos la población usando dos variables de entrada de género y clase. Ahora, quiero identificar qué división está produciendo subnodos más homogéneos usando Gini.



03008915 C/ Ferrocaril, 22, 03570 La Vila Joiosa **Tel** 966870140 **Fax** 966870141 http://portal.edu.gva.es/iesmarcoszaragoza

División en género:

1. Calcular, Gini para el subnodo femenino= `(0.2) * (0.2) + (0.8 ) * (0.8) = 0.68`
2. Gini para sub-nodo Masculino = `(0.65) * (0.65) + (0.35) * (0.35) = 0.55`
3. Calcular Gini ponderado para Split Gender = `(10/30) * 0.68+ (20/30) * 0.55 = **0.59**`

Similar para Split on Class:

1. Gini para sub-nodo Clase IX = `(0.43) * (0.43) + (0.57) * (0.57) = 0.51`
2. Gini para sub-nodo Clase X = `(0.56) * (0.56) + (0.44) * (0.44) = 0.51`
3. Calcular Gini ponderado para Clase dividida = `(14 /30)*0.51+(16/30)*0.51 = **0.51**`

Se puede ver que la puntuación de Gini para *Dividir en género* es mayor que *Dividir en Clase,* por lo tanto, la división de nodos tendrá lugar en Género.

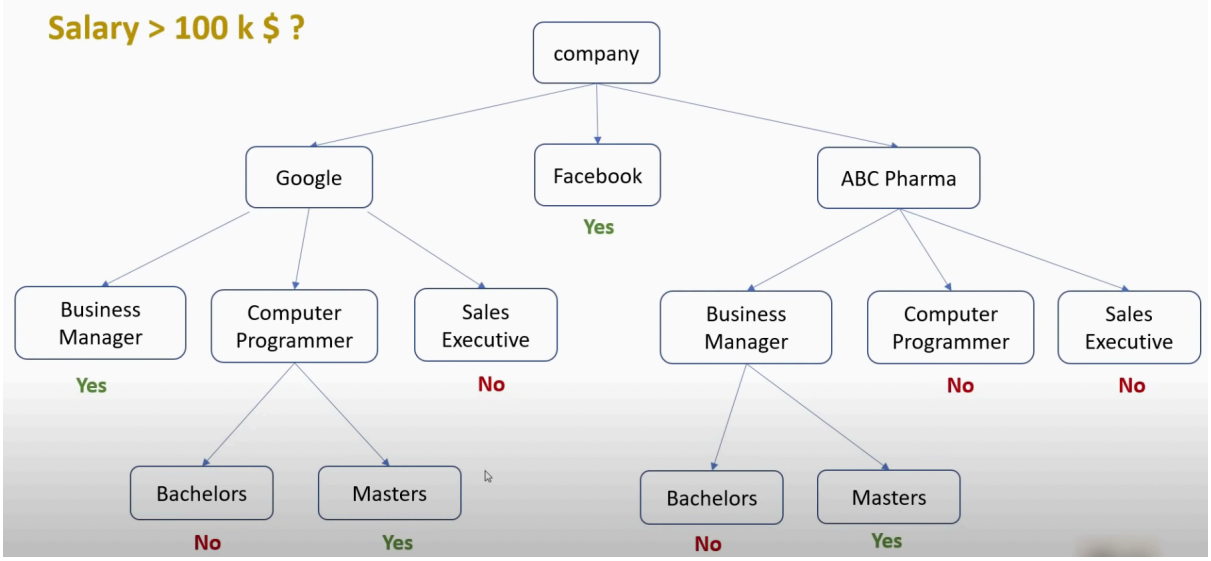
Gini es el parámetro predeterminado en el Algoritmo del árbol de decisión.

03008915 C/ Ferrocaril, 22, 03570 La Vila Joiosa **Tel** 966870140 **Fax** 966870141 http://portal.edu.gva.es/iesmarcoszaragoza

Ahora, imaginamos que tratamos de averiguar si un empleado gana más de 100k dólares dependiendo de la empresa, trabajo y estudios.
El dataset sería:

Company	Job	Degree	Salary_more_then_100k
google	sales executive	bachelors	0
google	sales executive	masters	0
google	business manager	bachelors	1
google	business manager	masters	1
google	computer programmer	bachelors	0
google	computer programmer	masters	1
abc pharma	sales executive	masters	0
abc pharma	computer programmer	bachelors	0
abc pharma	business manager	bachelors	0
abc pharma	business manager	masters	1
facebook	sales executive	bachelors	1
facebook	sales executive	masters	1
facebook	business manager	bachelors	1

Podemos imaginar un árbol de decisión como el que se muestra, pero dependerá de los atributos que seleccionemos para la toma de decisiones.



03008915 C/ Ferrocaril, 22, 03570 La Vila Joiosa **Tel** 966870140 **Fax** 966870141 http://portal.edu.gva.es/iesmarcoszaragoza

Ahora pasamos a la acción:

Antes de nada, los algoritmos de ML funcionan con variables numéricas, por tanto las pasamos a numéricas habiendo separado las características y el objetivo.

```
inputs = df.drop('salary_more_than_100k',axis=1)
target= df['salary_more_than_100k']
```

```
from sklearn.preprocessing import LabelEncoder
le_company=LabelEncoder()
inputs['company_n']= le_company.fit_transform(inputs['company'])
```

o bien:

```
inputs['Job_n']= inputs['Job'].astype('category').cat.codes
```

Eliminamos las columnas no numéricas:

```
inputs.drop(['Company','Job,'Degree'],axis=1,inPlace=True)
```

Y pasamos a entrenar el modelo:

```
from sklearn import tree
model = tree.DecisionTreeClassifier()
model.fit(inputs, target)
```

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
```

Y se predice...

```
model.predict([[2,1,0]])
```