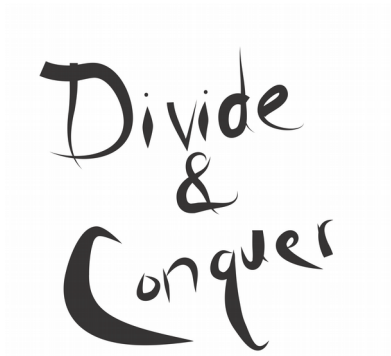
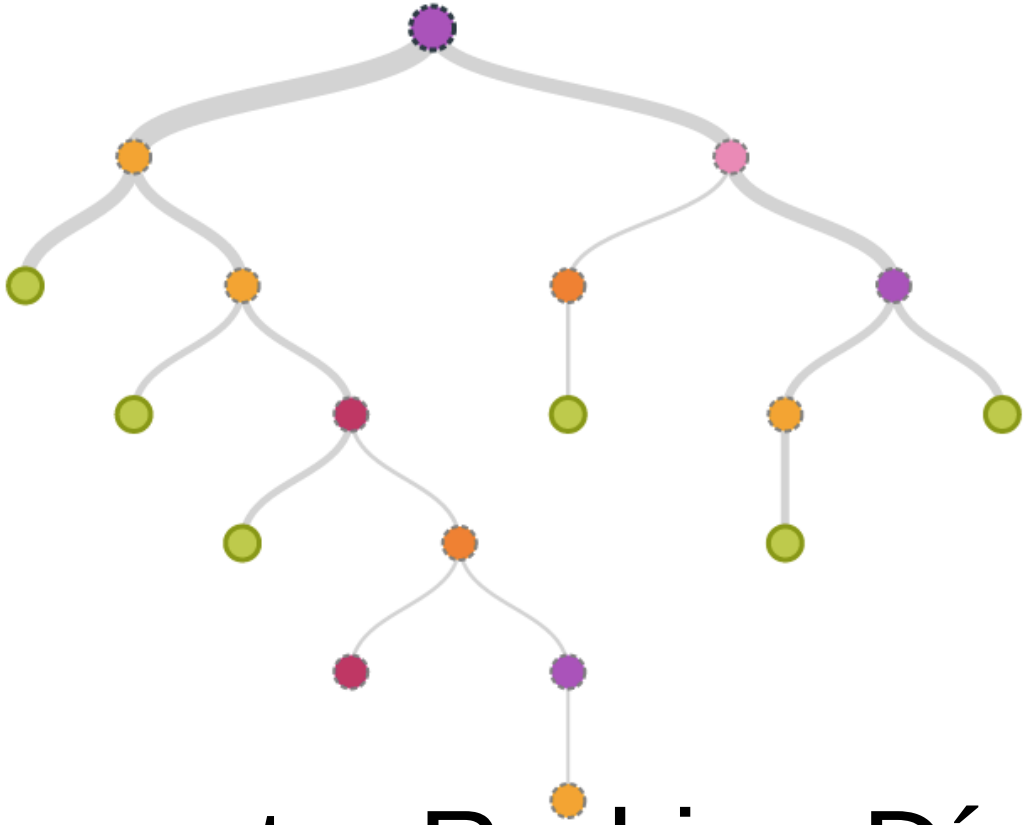


Tema 5.1

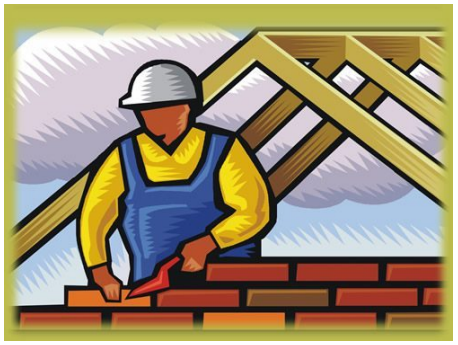
Dividir y conquistar- Clasificación empleando árboles de decisión



Ponente: Rodrigo Díaz

Introducción

La predicción del futuro de nuestras vidas puede reducirse a una serie de decisiones simples:



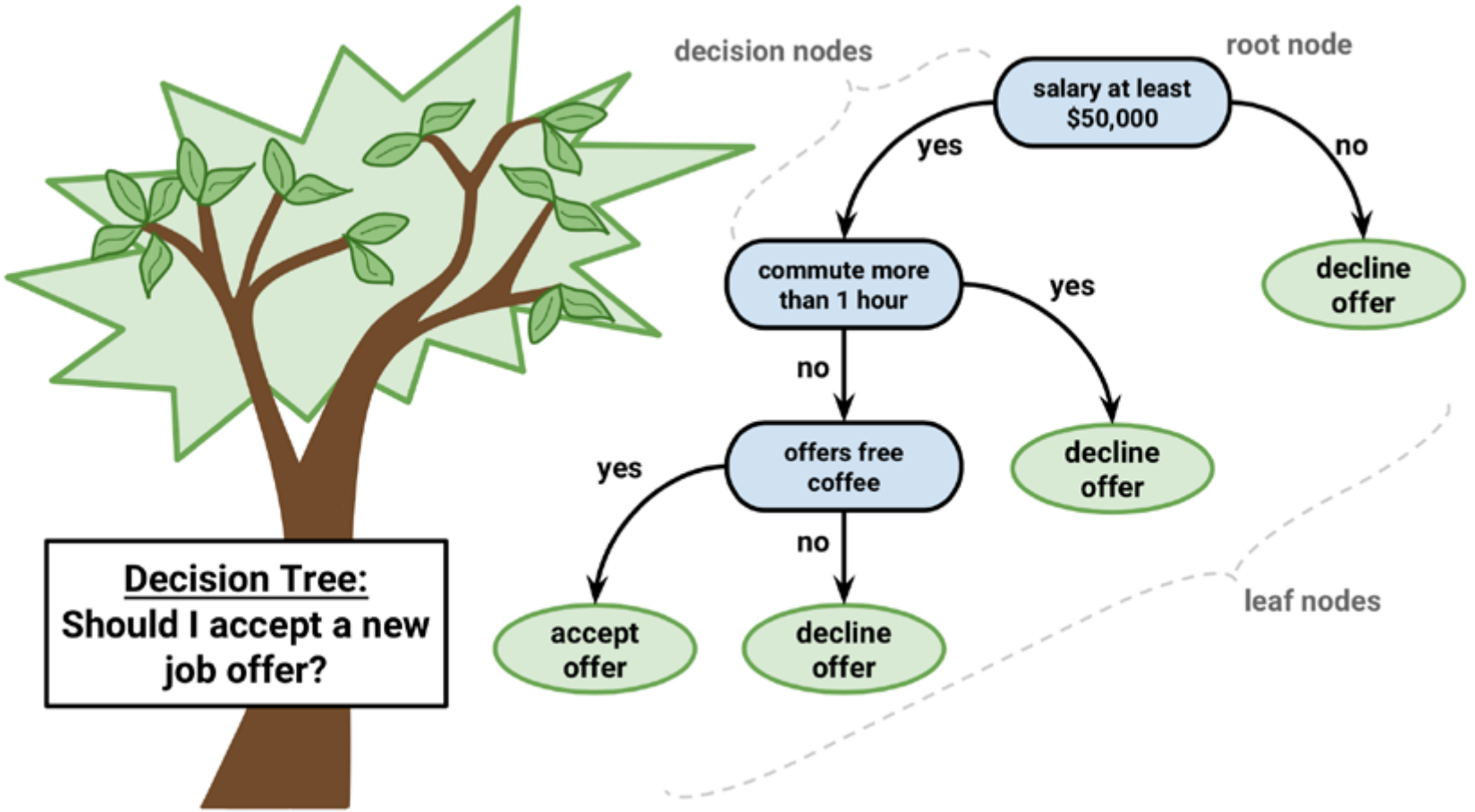
Aprenderemos….

Como los árboles de decisión (AD) “codisiosamente” dividen la data en segmentos de interes.

Los algoritmos más comunes usados en los AD.

Cómo usar estos algoritmos en problemas de la vida real, por ejemplo en préstamos bancarios.

AD son clasificadores robustos que utilizan estructura parecida a un árbol para modelar relaciones entre características y posibles resultados potenciales.



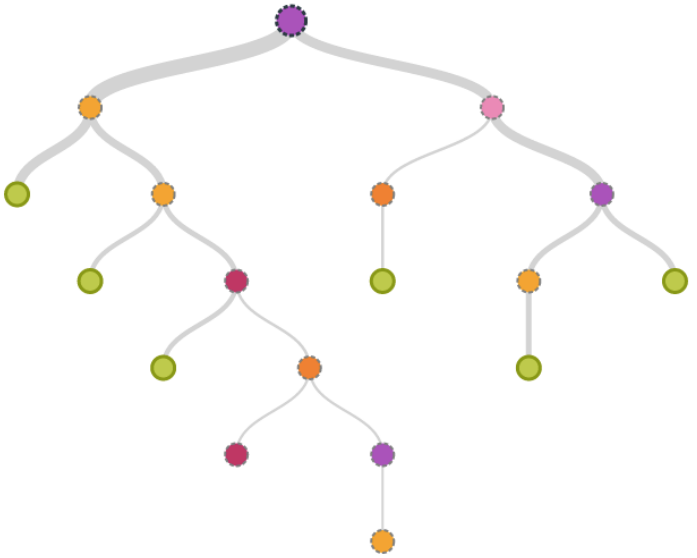
Se tiene la ventaja que en algunos casos el algoritmo resulta en una estructura comprensible para el ser humano, ofrenciendo información sobre por qué el modelo funciona o no para determinada tarea y haciendolo ideal para compartirlo en conferencias o prácticas de negocios.

Esta técnica tiene muchos usos, siendo quizás la más popular en ML, se aplica muy bien a distittos tipos de data y ofrece soluciones “fuera de caja”.

Por otro lado, no siempre se ajusta bien a todos los escenarios. No funciona bien cuando la data tiene demasiadas características nominales con muchos niveles. Tampoco para muchas varialbes numéricas. Esto porque podrían sobreajustar la data (aunque se puede corregir en parte).

Divide y vencerás

AD utiliza el particionamiento recursivo. Es decir, la división de la data en subgrupos, los cuales son a su vez reducidos en subgrupos.

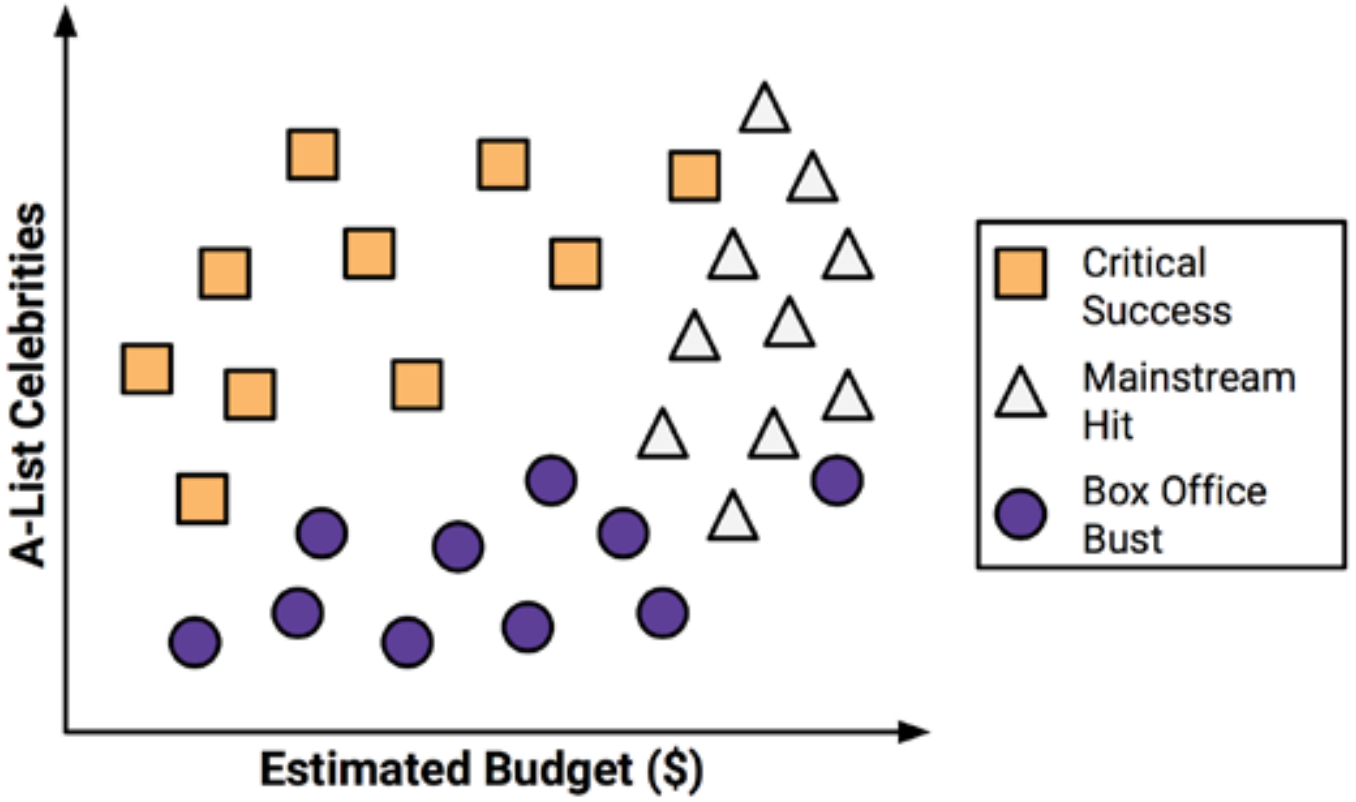


El proceso se detiene cuando el algoritmo determina que el subgrupo es suficientemente homogenio (u otro criterio de detención).

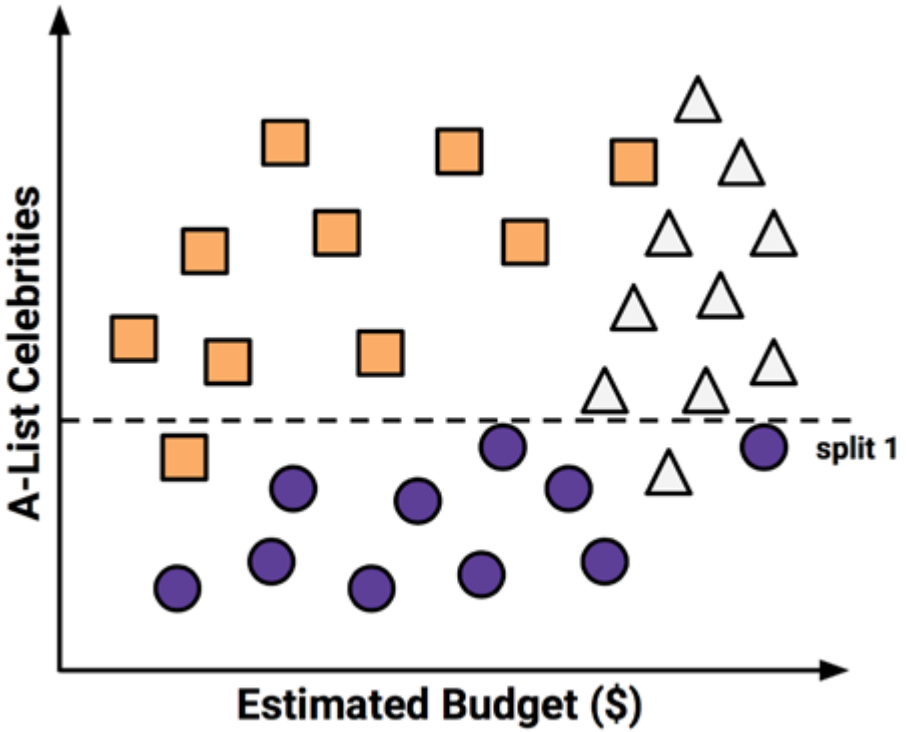
1. Todos (o casi) de las muestras en el nodo tienen la misma clase.
2.No hay más característicasa que distingan entre las muestras.
3. El árbol ha crecido hasta un limite pre-establecido.

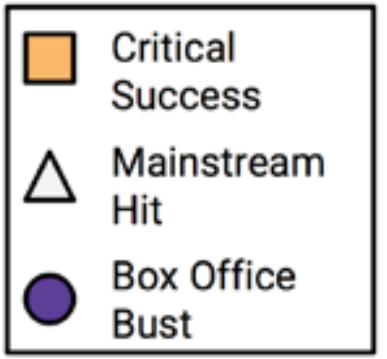
Ejemplo de estudio en Hollywood

Queremos predecir si una película va a ser un: Critical Success, Mainstream Hit, or Box Office Bust.



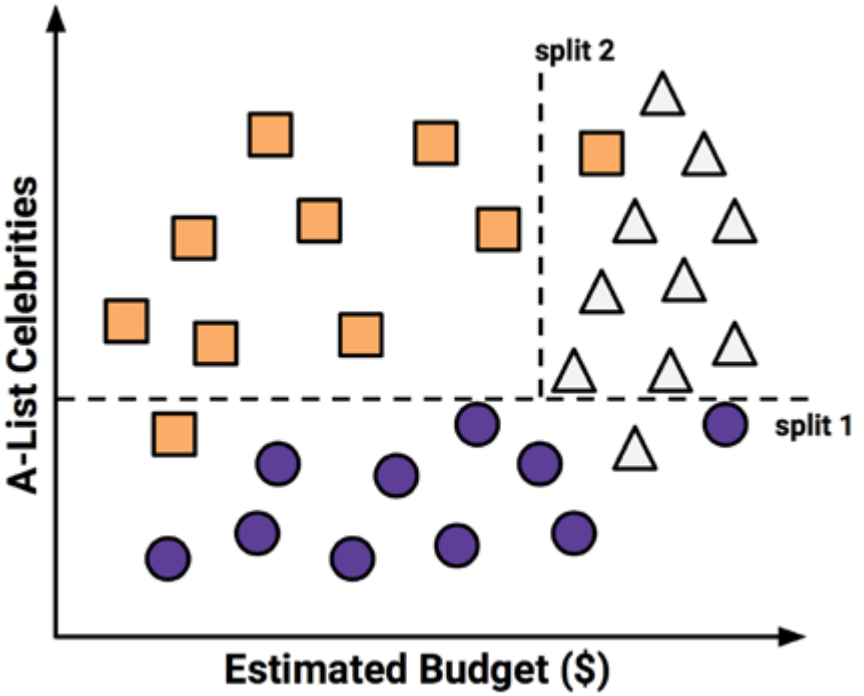
Usamos la estrategia de Julio César





Dividimos en función del número de celebridades

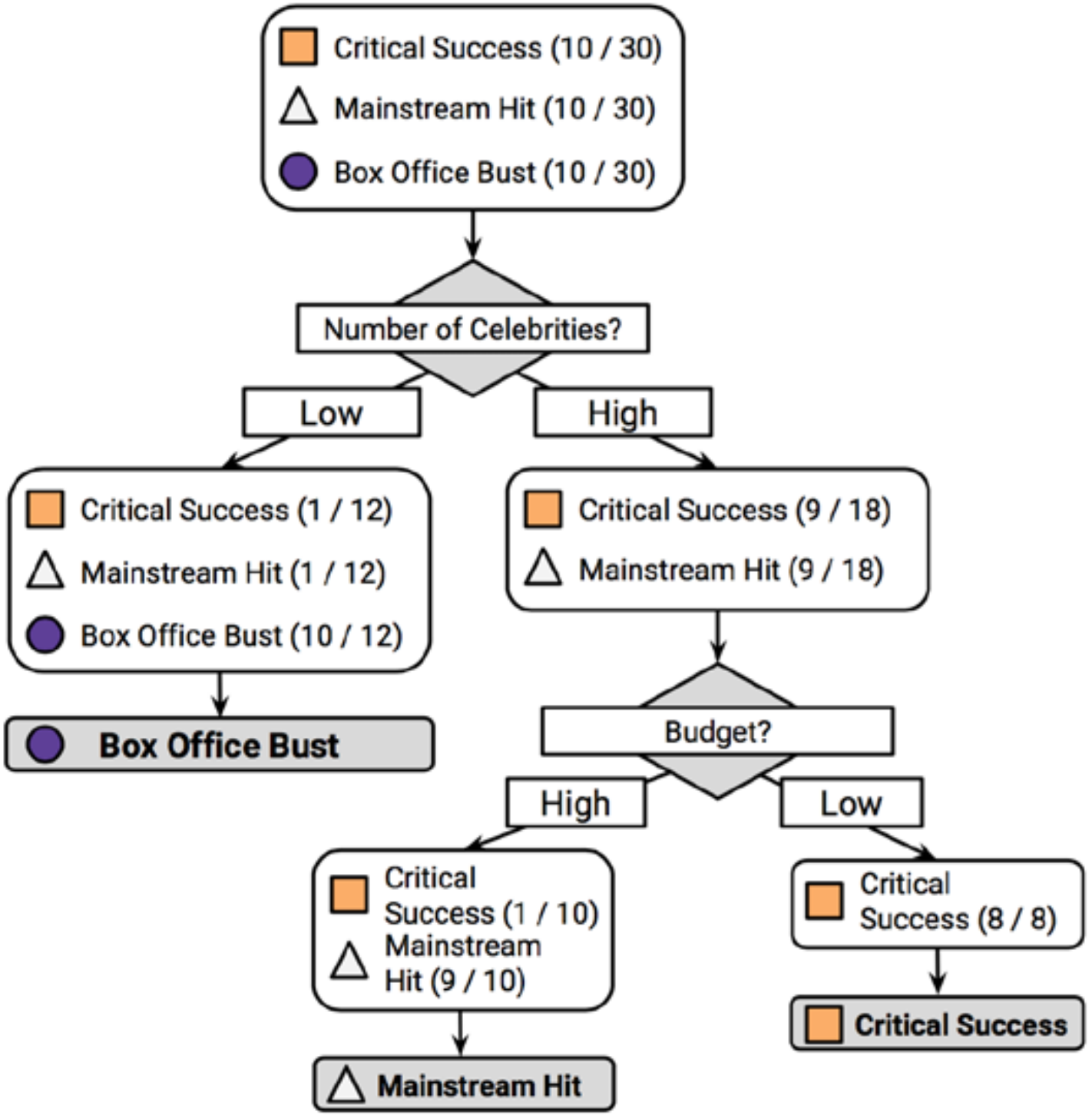
En función del presupuesto



Podriamos seguir diviendo pero no es aconsejable sobreajutar nuestro modelo.

Nota importante:

Se llama la atención a la posibilidad de dividir la data mas limpiamente con una línea diagonal. Sin embargo, el modelo lo hace divisiones paralelas a los ejes. Es es una de las limitaciones, al tomar en cuenta 1 característica a la vez.


Critical Success (10 / 30)
Mainstream Hit (10 / 30)
Box Office Bust (10 / 30)
Number of Celebrities?
Low
High
Critical Success (1 / 12)
Mainstream Hit (1 / 12)
Box Office Bust (10 / 12)
Box Office Bust
Critical Success (9 / 18)
Mainstream Hit (9 / 18)
Budget?
High
Low
Critical Success (1 / 10)
Mainstream Hit (9 / 10)
Mainstream Hit
Critical Success (8 / 8)
Critical Success

Debido a que la data de la vida real suele contener muchas características, se puede emplear un algoritmo popular denominado C5.0.

Creado por Ross Quinlan ID3-→ C4.5-→ C5.0

Es el algoritmo standar a la hora de producir AD ya que trabaja bien en la mayoría de los casos.

Sus desventajas son relativamente pocas y se pueden evitar ampliamente.

Strengths	Weaknesses
• An all-purpose classifier that does well on most problems • Highly automatic learning process, which can handle numeric or nominal features, as well as missing data • Excludes unimportant features • Can be used on both small and large datasets • Results in a model that can be interpreted without a mathematical background (for relatively small trees) • More efficient than other complex models	• Decision tree models are often biased toward splits on features having a large number of levels • It is easy to overfit or underfit the model • Can have trouble modeling some relationships due to reliance on axis-parallel splits • Small changes in the training data can result in large changes to decision logic • Large trees can be difficult to interpret and the decisions they make may seem counterintuitive

Escogiendo la mejor división:

La **pureza** de un subgrupo es el grado en que éste contiene a una sola clase. Se dice que es un subgrupo **puro** si sólo tiene una clase.

El algoritmo C5.0 utiliza el concepto de entropía (grado de desorden de un sistema) para cuantifcar la pureza de un grupo. Los AD intentan reducir la entropía con el fin de aumentar la homogeneidad dentro del grupo.

Tipicamente, la entropía es medida en bits. En caso de sólo dos clases posibles, la entropía asume valores entre 0 y 1. Es decir, para *n* clases, la entropía va desde 0 a log2 (*n*). Donde el valor mínimo indica homogeneidad absoluta, mientras el mayor valor significa que la data es totalmente diversa.

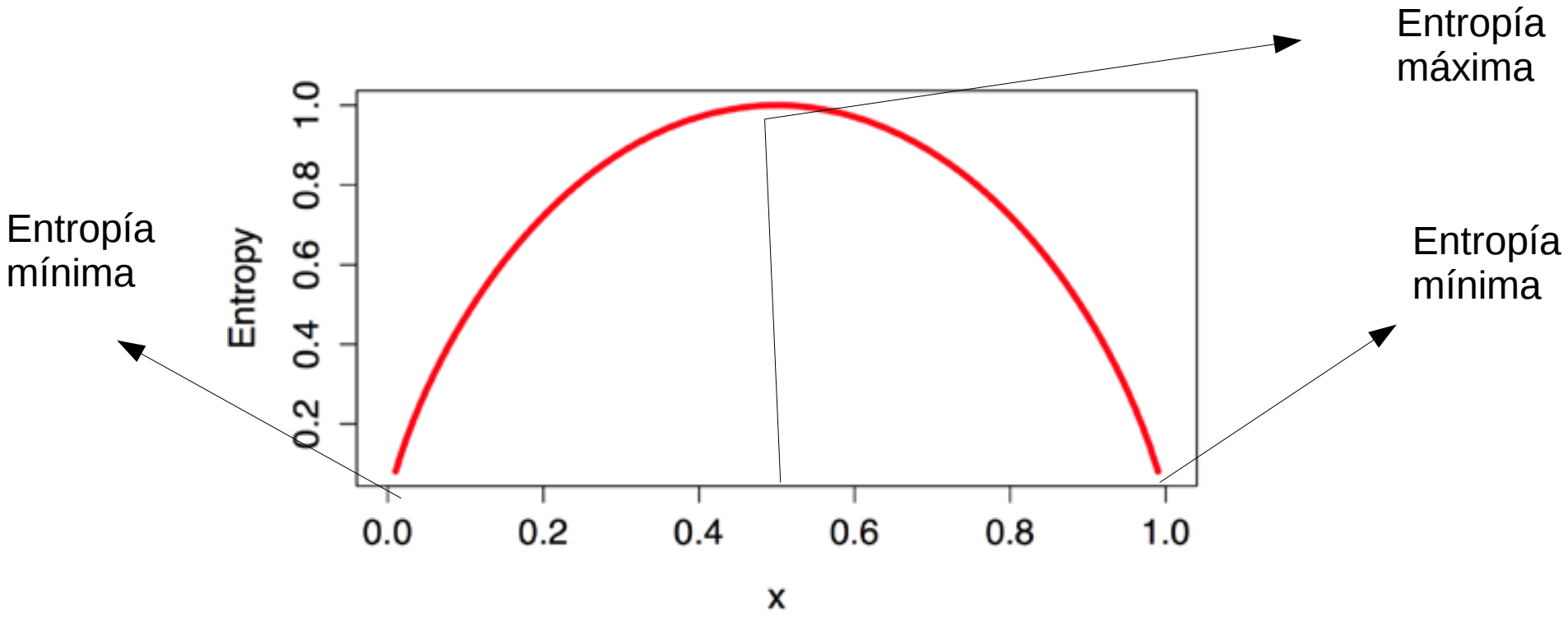
$$\text{Entropy}(S) = \sum_{i=1}^{c} -p_i \; log_2(p_i)$$

S=segmento de datos
c=número de niveles en la clase
Pi= proporción de valores del nivel
i de la clase.

For example, suppose we have a partition of data with two classes:
red (60 percent) and
white (40 percent). We can calculate the entropy as follows:

```
-0.60 * log2(0.60) - 0.40 * log2(0.40)
[1] 0.9709506
```

```
curve(-x * log2(x) - (1 - x) * log2(1 - x),
col = "red", xlab = "x", ylab = "Entropy", lwd = 4)
```



El algoritmo utiliza la entropia para determinar la característica óptima con la cual realizar la divisíón. Esto lo hace calculando el cambio en la homogeneidad para cada división posible.

La ganancia de información para una característica F es calculada como la diferencia de la entropía en el segemento antes de la partición y luego de la partición.

$$\mathrm{InfoGain}(F) = \mathrm{Entropy}(S_1) - \mathrm{Entropy}(S_2)$$

Más aún, es necesario conocer la entropía en todas las particiones creadas luego de la división.

De manera resumida, la entropía total resultante de una división es la sumatoria de la entropía en cada una de las *n* particiones pesada por la proporción de casos dentro de la partición *Wi.*

$$\text{Entropy}(S) = \sum_{i=1}^{n} w_i \ \text{Entropy}(P_i)$$

Mientras mayor ganancia de información, mejor sera la característica escogida para generar grupos homogéneos. Si fuese 0 no hay reducción de la entropía, mientras que el máximo de información se obtiene si la entropía no cambia antes y despues de la división.

La ecuación anterior asume variables nominales, pero tambien se puede aplicar para variables numéricas. Para esto se escogen secciones para dividir la variable, generándose categorias, con lo cual el modelo puede trabajar.

Existen otros criterios de división distintos al de ganancia de información. Algunos de éstos son: índice de Gini, estadístico de Chi cuadrado y ganancia de radio.

An Empirical Comparison of Selection Measures for Decision-Tree Induction. Machine Learning. 1989; 3:319-342.

Podando el AD

Debido a que podríamos dividir casi indefinidamente nuestro árbol, se causaría un sobreajuste de la data de entrenamiento, por lo que es necesario podar el AD para reducir su tamaño en orden de la mejor generalización de la data a evaluar.

Podemos detener el proceso de división cuando se hayan alcanzo determiando números de divisiones o cuando los nodos contengan poco número de casos. Estto se denomina **detención temprana** o **pre poda (early stopping** o **pre-pruning).** Sin embargo corremos el riesgo de perder patrones importantes que podrían surgir si el AD hubiera crecido más.

Una alternativa es la post poda (post-pruning), donde primero se deja intencionalmente crecer el AD y luego se podan los nodos en los niveles más convenientes. Generalmente esta opción es más efectiva, ya que podemos estar seguros que todas las estructuras importantes han sido descubiertas.

Esposito F, Malerba D, Semeraro G. A Comparative Analysis of Methods for Pruning Decision Trees. IEEE Transactions on Pattern. Analysis and Machine Intelligence. 1997;19: 476-491.

El algoritmo C5.0 tiene el beneficio automatizar este proceso de poda, con la final rducción de los nodos donde hay muy poco efecto para la clasificación. En algunos casos hay un re-arreglo del árbol, lo que se conoce como subtree raising y subtree replacement. El primero cuando se sube una rama de nivel y el segundo cuando una decisión es suplantada por una más sencilla.

Por ultimo, el balance entre sobre-ajustar y sub-ajustar el AD conlleva un poco de arte, pero si la exactitud del modelo es vital, requiere invertir tiempo en distintas opciones de poda. Como veremos en el ejemplo de R, la fortaleza del C5.0 es que se pueden ajustar ampliamte las opciones de entrenamiento.

Vamos a R…