Guía para árboles de decisión.

# **Parte 1. Preprocesamiento** (súper rápidito)**.**

1. Queremos entender cómo afecta la normalización de la variables continuas en un árbol de decisión (si se anima, piénselo unos minutos antes a ver que espera)
   1. Redefinir el preprocesamiento de datos eliminando el scaling de las variables continuas.
   2. Evaluar performance del modelo, con parámetros por default y CV = 5 (hágalo simple, saque provecho de la función cross\_val\_score(). Compare los resultados con el modelo que normaliza las variables continuas.

# **Parte 2. Exploración de hyperparámetros univariada**

1. Vimos que la máxima profundidad del árbol permitida es un parámetro que si se lo libera demasiado se incrementa en exceso la complejidad de la frontera de decisión y el modelo sobrejusta completamente. En el gráfico se ve con claridad que ya una profundidad de 5 da un sobreajuste extremo (average\_prec = 1 en training y 0.5 en evaluation). Además el testing apenas pasa el avg\_precision =0.6.
   1. Repita el barrido pero seteando splitter = ‘random’}

Hint:

pipeline.named\_steps['clasificador'].set\_params(\*\*{'splitter':'random'})

Mejoró la performance en el test? Discuta que le parece que pudo haber pasado.

* 1. Y si además setea el criterio de entropía? Sigue siendo ‘mucho’ una profundidad de 5?

También vimos que la complejidad aumenta en tanto reducimos el umbral que establece cuál es el número mínimo de muestras requeridas en un nodo para que éste pueda ser dividido (min\_samples\_split).

* 1. Cuál es el valor por default de min\_samples split? Donde se ubica en la curva que trazamos?
  2. Cuál es el Max Depth por default del árbol?
  3. Fije por ejemplo min\_sample\_split = 0.5 y vuelva a barrer la curva de max\_depth. Cambió algo? Hasta donde llega ahora el score en el training?
     1. Hint, ejecute primero la sig línea: pipeline.named\_steps['clasificador'].set\_params(\*\*{'min\_samples\_split': 0.5})
  4. Podemos pensarlos como hyperparámetros independientes?
  5. Qué comportamiento espera con el seteo por default del árbol?

1. Vuelva a la config inicial del pipeline 'min\_samples\_split':2
   1. Haga un barrido del parámetro de postpruning ccp\_alpha. Explore en qué rango (dentro de 0.0 y 1.0) tiene sentido moverlo en este dataset. El default es 0.0 (no postpruning) el otro extremo representa un pruneado total (i.e. ud ha recortado todo el árbol y se quedó con un único nodo, el root del árbol).
2. Considere ahora el número mínimo de muestras admitido en un nodo ‘hoja’ del árbol. Realice un barrido de ese parámetro para ganar intuición sobre él (se mueve entre 0 y 0.5, o un entero en su defecto)
   1. Nota alguna ventana de mejora en el test?
3. Pruebe barrer max\_features. Pero ojo que es muy sensible al seteo que tenga de min\_sample\_split

Osea, ojo con la exploración univariada de hyperparámetros. Está bien para ganar una primera impresión del problema. Pero ojo, porque un barrido univariado es un corte del espacio N-dimencional de hyperparámetros, sólo eso. Lo que ve para un hyperparámetro es condicional al efecto de cómo están seteados los otros hyperparámetros.

**Gridsearch**

1. Con exactamente el mismo espacio de búsqueda intente:
   1. Tome el gridsearch y cambie el número de folds de 2 a 3, 5 y luego a 10 (cv = 10) -- pueden dividirse entre los integrantes del grupo--. Mejora? Analícelo no solo con la performance media, mire el desvío y, mire el valor obtenido en el test.