Métodos No Lineales

Arboles de Decisión

Universidad Católica del Uruguay

1

Árboles de Decisión

- Muy utilizado y popular
- Aproxima funciones que toman valores discretos.
- La función aprendida se representa como un árbol
- Robusto ante datos con ruido
- Aprende expresiones disyuntivas: los árboles aprendidos se pueden también representar como reglas *if-then* (intuitivas)
- Numerosas aplicaciones: diagnósticos médicos, causas de fallo en equipos, evaluación de riesgos de créditos en la concesión de préstamos...
- Árbol de clasificación

3

3

Representación como árboles

- Cada nodo (no terminal) especifica un test de algún atributo de la instancia
- Cada rama corresponde a un posible valor del atributo
- Cada nodo terminal indica la clase en la que se clasifica
- Instancias no vistas se clasifican recorriendo el árbol: pasándoles el test en cada nodo, por orden desde el nodo raíz hasta algún nodo hoja, que da su clasificación

4

4

Ejemplo: ¿Vamos a jugar al tenis?

- Tarea: decidir si se va a jugar al tenis.
- Criterio: se va a jugar al tenis ...
  - si va a llover, sólo si no hay mucho viento.
  - si va a estar soleado pero no muy húmedo.
  - si va a estar nublado.
  - no en cualquier otro caso.

5

5

¿Vamos a jugar al tenis?

```
SI (EstadoDelTiempo== SOLEADO){
        SI (Humedad == ALTA)
                DEVOLVER NO;
        SINO, SI (Humedad == NORMAL)
                DEVOLVER SI;
} SINO SI(EstadoDelTiempo == CUBIERTO){
        DEVOLVER SI;
} SINO SI (EstadoDelTiempo == LLUVIOSO){
        SI (Viento == FUERTE)
                DEVOLVER NO;
        SINO SI (Viento == SUAVE)
                DEVOLVER SI;
}
```

6

6

Ejemplo: JugarTennis

- Clasificar las mañanas de sábado en si son o no adecuadas para jugar al tenis – supongamos que hemos creado (**¿?**) el siguiente árbol de decisión:

- **Instancia**: EstadoDelTiempo=SOLEADO, Temperatura=CALUROSO, Humedad=ALTA, Viento=FUERTE
- Entra por el camino izquierdo y se predice JugarTennis=No

7

7

Ejemplo: JugarTennis

- El árbol representa una disyunción de conjunciones de restricciones sobre los valores de los atributos de las instancias
- Un camino = una conjunción de *tests* de atributos
- Todo el árbol = disyunción de estas conjunciones
- Este árbol es:

(EstadoDelTiempo=SOLEADO ^ Humedad=Normal)
v (EstadoDelTiempo=CUBIERTO)
v (EstadoDelTiempo=LLUVIOSO ^ Viento=SUAVE)

8

8

Tipos de árboles

- Arboles de clasificación: valores de salida discretos
  - CLS, ID3, C4.5, ID4, ID5, C4.8, C5.0
- Arboles de regresión: valores de salida continuos
  - CART, M5, M5'

9

9

Algoritmo básico: ID3 "Iterative dicotomiser "[Quinlan, 1986]

- Basado en el algoritmo CLS (Concept Learning Systems) [Hunt et al., 1966], que usaba sólo atributos binarios
- Búsqueda ávida
- Construir el árbol de arriba a abajo, preguntando: ¿Qué ***atributo*** seleccionar como nodo ***raíz***?
- Se evalúa cada atributo para determinar cuán bien clasifica los ***ejemplos*** por sí mismo
- Se selecciona el ***mejor*** como nodo
- Repetir usando los ***ejemplos*** asociados con el nodo
- Parar cuando el árbol ***clasifica correctamente todos los ejemplos*** o cuando ***se han usado todos los atributos***
- Etiquetar el nodo hoja con la clase de los ejemplos

10

10

Algoritmo básico

Funcion APRENDER_ARBOL_DECISION (***ejemplos, atributos, valor_defecto***): ARBOL_DECISION
COM
si *ejemplos* está vacío **entonces devolver** *valor_defecto*
si no
si todos los elementos de ***ejemplos*** tienen la misma clasificación **entonces**
devolver la clasificación
si no si *atributos* esta vacío **entonces devolver** VALOR_MAYORIA (*ejemplos)*
si no
mejor = ELEGIR_ATRIBUTO (atributos, ejemplos)
arbol <- un nuevo árbol de decisión, cuya raíz es ***mejor***
m <- VALOR_MAYORIA(*ejemplos)*
para cada valor v_i de ***mejor*** hacer
ejemplos(i) <- {elementos de ejemplos con *mejor* = v_i}
subarbol = APRENDER_ARBOL_DECISION(*ejemplos(i)*, *atributos – mejor, m*)
añadir una rama a *arbol* con la etiqueta v_i y el subárbol *subarbol*
devolver *arbol*
FIN

11

11

ID3

- Paso clave: ¿cómo seleccionar el atributo?
- Nos gustaría el más útil para clasificar ejemplos; el que los separa bien
- ID3 escoge la variable más efectiva usando la ***ganancia de información*** (maximizarla)
- Mide cuán bien un atributo separa los ejemplos de entrenamiento de acuerdo a su clasificación objetivo, y selecciona el mejor.
- Reducción esperada en ***entropía*** (incertidumbre), causada al particionar los ejemplos de acuerdo a este atributo

12

12

Entropía o cantidad esperada de información

- Medida de la homogeneidad de un conjunto de muestras.
- En teoría de la información: medida de la incertidumbre sobre una fuente de mensajes.
- Sea una fuente **S** que puede producir ***n*** mensajes diferentes $\{m_1, m_2, \ldots, m_n\}$. Los mensajes son independientes, y la probabilidad de producir el mensaje m_i es p_i.
- Para tal fuente **S** con distribución de probabilidades de los mensajes $P = (p_1, p_2, \ldots p_n)$, la entropía E(P) es:

$$E(P) = -\sum_{i=1}^{n} p_i * \log_2(p_i)$$

13

13

Entropía

- Si un conjunto ***T*** de registros de una base de datos se particiona en *k* clases $\{C_1, C_2, \ldots, C_k\}$ sobre la base de un cierto atributo, entonces la **cantidad media de información necesaria** para identificar la clase de un registro es $E(P_T)$, donde P_T es la distribución de probabilidades de las clases:

$$P_T = \left(\frac{|C_1|}{|T|}, \frac{|C_2|}{|T|}, \ldots, \frac{|C_k|}{|T|}\right)$$

14

14

Entropía

- Dado un conjunto **S** con ejemplos positivos y negativos de un concepto objetivo, (problema de **2 clases**) la entropía del conjunto **S** con respecto a esta clasificación binaria es

$$\mathbf{E(S) = - p(}P\mathbf{)log_2\, p(}P\mathbf{) - p(}N\mathbf{)log_2\, p(}N\mathbf{)}$$

- La clase C_1 corresponde a *P – positivos-* y la clase C_2 corresponde a *N – negativos - .*

15

15

Pregunta: ¿cuál es la entropía del conjunto completo de Jugar Tenis?

ESTADO DEL TIEMPO	TEMPERATURA	HUMEDAD	VIENTO	¿JUGAR?
Cubierto	Caluroso	Alta	suave	SI
Cubierto	Caluroso	Normal	suave	SI
Soleado	Caluroso	Alta	suave	No
Soleado	Caluroso	Alta	fuerte	No
Cubierto	Frio	Normal	fuerte	SI
Lluvioso	Frio	Normal	suave	SI
Lluvioso	Frio	Normal	fuerte	No
Soleado	Frio	Normal	suave	SI
Cubierto	Templado	Alta	fuerte	SI
Lluvioso	Templado	Alta	suave	SI
Lluvioso	Templado	Normal	suave	SI
Lluvioso	Templado	Alta	fuerte	No
Soleado	Templado	Alta	suave	No
Soleado	Templado	Normal	fuerte	SI

a) 0.991
b) 0.940
c) 0.494
d) 0.302

16

16

Ejemplo

- Para los datos de "Jugar Tenis", donde *jugar* es el atributo de salida, tenemos para el conjunto de datos completo:

$$P_T = \left(\frac{9}{14}, \frac{5}{14}\right)$$

- Usando la ecuación de la entropía tenemos:

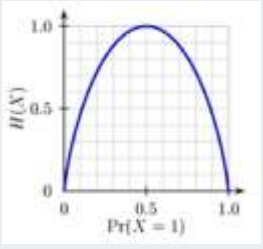
$$E(T) = E(P_T) = -\left(\frac{9}{14}\log\left(\frac{9}{14}\right) + \frac{5}{14}\log\left(\frac{5}{14}\right)\right) = 0.94$$

17

17

Entropía

- La entropía es 0 si la salida es ya conocida o el mensaje es invariante.
- La entropía es máxima si no tenemos conocimiento alguno sobre el sistema (o si cualquier resultado es igualmente posible)



Entropía de un sistema de clase 2

18

18

Ganancia de Información

- Particionamos el conjunto sobre la base de un atributo de entrada *X*, en subconjuntos $T_1, T_2, \ldots, T_n$.
- La información necesaria para identificar la clase de un elemento de T es la **media ponderada** de la información necesaria para identificar la clase de un elemento de cada subconjunto:

$$E(X,T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} E(T_i)$$

- Ganancia de Información = **E(T) – E(X,T)**
- ID3 calcula la ganancia de información para cada atributo y selecciona el que tenga **mayor ganancia**

19

19

Ganancia de Información

- La ganancia de información mide la **reducción esperada de la entropía**, o incertidumbre.

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

- Values(A) es el conjunto de todos los posibles valores del atributo **A** , y S_v el subconjunto de **S** para el cual el atributo **A** tiene valor **v**

$S_v = \{s \text{ in } S \mid A(s) = v\}$.

- El primer término es entonces solamente la entropía del conjunto **S** original.
- El segundo término es el valor esperado de la entropía luego de particionar **S** utilizando el atributo **A**

20

20

Ejemplo:

- Calcular la ganancia de información debida a particionar el conjunto de acuerdo al atributo Temperatura
- Temperatura tiene tres valores
  - Frio
  - Templado
  - Caluroso
  - $|T_{Frio}| = 4$, $|T_{Templado}| = 6$, $|T_{Caluroso}| = 4$

$$E(\text{temperatura},T) = \frac{4}{14} E(T_{Frio}) + \frac{6}{14} E(T_{Templado}) + \frac{4}{14} E(T_{Caluroso})$$

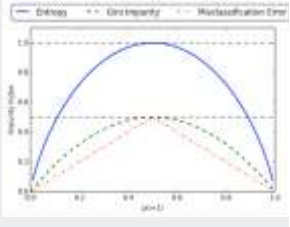
Ganancia (temperatura, T) = 0.940 - 0.911= **0.029 bits**

21

21

Criterios de "impureza" usados para el split

Cada split trata de hacer el nodo hijo más puro



Gini

Ganancia de información (Entropía)

Error de clasificación

22

Pre-Pruning – criterios para detener el algoritmo

En datasets reales, no es muy probable que obtengamos nodos terminales 100% homogéneos. Necesitamos indicar al algoritmo *cuándo parar:*

- Ningún atributo satisface un umbral de ganancia de información mínimo
- Se ha alcanzado una profundidad máxima
- Hay menos ejemplos que un cierto mínimo en el subárbol actual

23

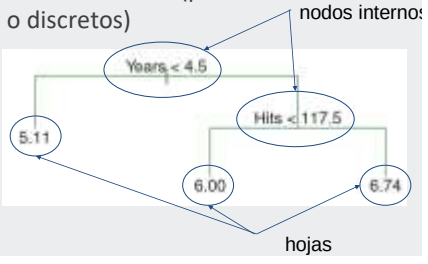
Pruning - poda

- Permitir al árbol crecer hasta el máximo, y *después* podar las ramas que no cambien efectivamente el error de clasificación – *post-pruning*
- A veces puede ser una mejor opción
- Requiere cómputos adicionales

24

Árboles de decisión - CART

- CART: Classification And Regression Tree
- El modelo de predicción se representa mediante un árbol binario (predictores continuos o discretos)

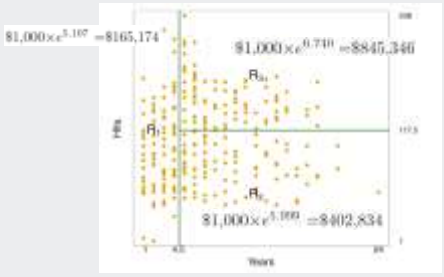


Ejemplo: predicción del salario de un bateador
Predictores: Años de jugador y hits de la temporada anterior (salario transformado mediante logaritmo, en miles)

25

Árboles de decisión para regresión

- Segmentación producida en el espacio de predictores:



26

Árboles de decisión para regresión

- En regresión, el valor definido para cada región es el promedio de todos los elementos del entrenamiento que caen en la hoja.
- Ventajas:
  - son sencillos de interpretar
  - provee información intrínseca de cuáles son los predictores más informativos
- Desventajas
  - rendimiento inferior a otros clasificadores
  - muy dependiente de los datos que se usan en el entrenamiento

27

CART - Árbol de regresión- Construcción

- ¿Cómo determinar las variables y los valores de decisión para los nodos internos?
- Dado el conjunto de datos S, para <u>cada</u> predictor considerar todos los valores de decisión que divida a los datos en S1 y S2 y que minimiza:

$$\text{SSE} = \sum_{i \in S_1} (y_i - \bar{y}_1)^2 + \sum_{i \in S_2} (y_i - \bar{y}_2)^2$$

valor medio de los elementos que caen en S1 y S2 respectivamente

se busca que los valores en S1 y S2 sean lo más homogéneos posible

28

CART - Árbol de regresión
Construcción

- El mismo procedimiento es aplicado recursivamente con S1 y S2 hasta que cada subconjunto tenga un tamaño definido (ej: 20 elementos o menos). Este procedimiento es tedioso de aplicar en forma exhaustiva.
- Enfoque tradicional:
  - top-down (comienza por el tope del árbol-todos los elementos pertenecen a la misma región)
  - greedy: el mejor particionamiento se realiza en cada paso (en lugar de ver hacia adelante y elegir un split que mejore el modelo pero en una instancia futura)

29

CART - Árbol de regresión
Poda

- El procedimiento puede producir árboles que sobreajustan a los datos de entrenamiento
- Solución: dejar crecer el árbol (lo denominamos T_0) y "podarlo", obteniendo el mejor subárbol posible
  - se puede evaluar cada subárbol candidato mediante cross-validation
  - sin embargo encontrar el subárbol indicado puede ser computacionalmente costoso
  - propuesta: cost–complexity prunning

30

CART - Árbol de regresión
Cost-complexity prunning

- Se define un parámetro α al cual corresponde un subárbol $T \subset T_0$ tal que:

$$\sum_{m=1}^{|T|} \sum_{i:\, x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

promedio de los valores de la variable dependiente que caen en el nodo terminal asociado a la región Rm

para cada región (nodo terminal u hoja)

para cada elemento que cae en la región (u hoja) Rm

- $|T|$: cantidad de nodos terminales de T

31

CART - Árbol de regresión
Cost-complexity prunning - Algoritmo

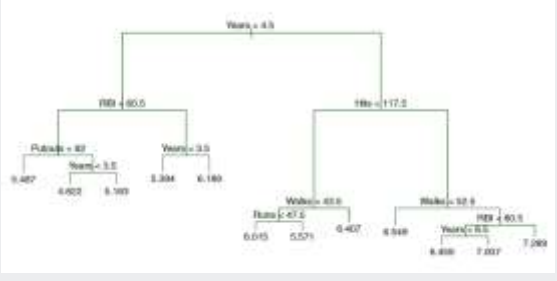
1. Desarrollar en forma completa un árbol (T_0)
2. Generar los diferentes subárboles resultantes de podar T_0 a partir de distintos valores de α
3. Para determinar α óptimo usar CV. Para cada k-fold k=1..*K*:
   a. calcular el cost-complexity prunning sobre el entrenamiento de los k-1 folds
   b. calcular el MSE sobre el k-fold

   Elegir el α asociado al MSE más bajo
4. Retornar el subárbol correspondiente al α óptimo

32

CART - Árbol de regresión
Ejemplo de Poda

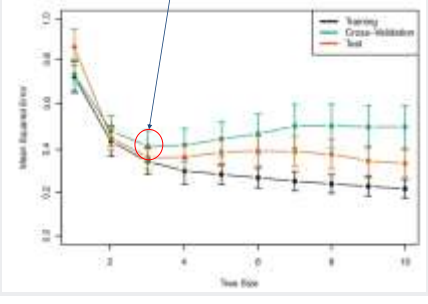
- Ejemplo: árbol de bateadores desarrollado en forma completa (9 atributos en dataset original)



33

CART - Árbol de regresión
Ejemplo de Poda

- el error mínimo en CV se da podando hasta tener 3 nodos terminales



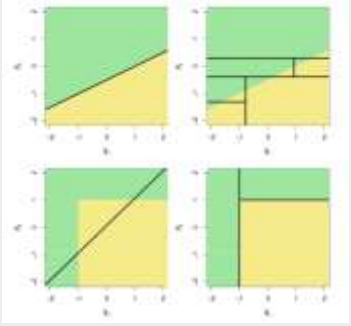
34

CART - Árbol de regresión

- Un árbol de decisión indica como valor de predicción de una hoja a la media de los valores que caen en ella.
- Cómo mejorarlo?
  - Modelo M5p (prime): en cada hoja realiza una regresión lineal ajustando los coeficientes con las tuplas que corresponden a la misma.

35

Árboles de decisión - Comparación



Ejemplo: modelos que se ajustan mejor a un problema particular.

36

CART - Árbol de regresión Poda

- El parámetro *cp* puede ser elegido mediante cross-validation (CV)
  - elegir el α_i que minimiza CV(α_i) (para diferentes valores de α_i)
- opción: one standard error rule (1SE Rule)
  - elige el subárbol tal que el error en CV está a una distancia de un error standard del árbol óptimo
  - http://www.stat.cmu.edu/~ryantibs/datamining/lectures/19-val2.pdf
  - mejora la generalización

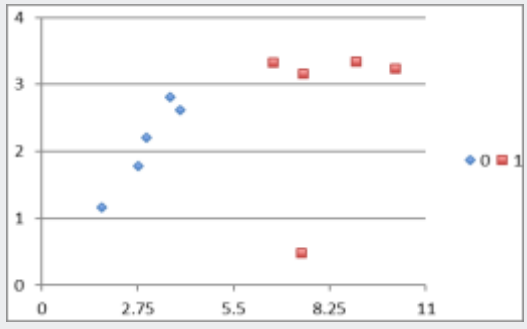
37

TA1

- "CART-dataset.csv"
- problema de clasificación binaria simple.
- Tenemos solamente dos variables de entrada (X1 yX2) y una sóla variable de salida (Y).
- El ejemplo está diseñado para que el algoritmo encuentre al menos dos puntos de división para clasificar el conjunto de entrenamiento

38

38



39

39

Aprendizaje del modelo CART

- Dividir los datos en un punto de division involucra separar todos los datos en ese nodo, en dos grupos, a la izquierda o derecha del punto de división.
- Si estamos trabajando en el primer punto de split del árbol, entonces todo el dataset estará involucrado.
- No nos interesa la clase a que pertenece el punto de división, sino solamente la composición de los datos asignados al subárbol izquierdo y derecho.
- Usamos una función de costo para evaluar la combinación de clases de la información de entrenamiento asignada a cada lado de la división.
- "índice GINI".

40

40

Indice Gini

- Calculamos el indice Gini para un nodo hijo:

$$G = 1 - (p_1^2 + p_2^2)$$

- p1 es la proporcion de instancias en el nodo con clase 1 y p2 para la clase 2.
- Ej: si el grupo IZQ. tiene 3 instancias con clase 0 y 4 con clase 1, entonces la proporción de instancias con clase 0 será 3/7 y la proporción con clase 1 será 4/7
- Abrir el archivo "gini.xlsx" para observar varios escenarios con distintas proporciones de instancias de dos clases y los correspondientes valores del indice Gini.

41

41

Gini

- Cuando el grupo tiene una mezcla 50-50 el Gini es 0.5, peor scenario possible
- En el ultimo ejemplo, vemos que todas las instancias caen en una sóla clase: el Gini es 0, un ejemplo de división perfecta

42

42

Calculo de indice Gini

- El cálculo del índice Gini para un punto de división: calcular el Gini para cada nodo hijo y ponderar los valores por la cantidad de instancias en el nodo padre

$$G = ((1 - (g1_1^2 + g1_2^2)) \times \frac{n_{g1}}{n}) + ((1 - (g2_1^2 + g2_2^2)) \times \frac{n_{g2}}{n})$$

- $g1_1$ es la proporción de instancias en el grupo 1 para la clase 1, $g1_2$ para la clase 2
- $g2_1$ es la proporción de instancias en el grupo 2 y clase 1, $g2_2$ grupo 2 y clase 2,
- n_{g1} y n_{g2} son los números totales de instancias en los grupos 1 y 2
- n es el numero total que queremos agrupar, del nodo padre

43

43

Primer punto de división candidato

- X1 = 2.7712
  - Si X1 < 2.7712 entonces IZQ
  - Si X1 >= 2.7712 entonces DER

División # 1		
2.771244718		
X1	Y	grupo
2.771244718		0 DER
1.728571309		0 IZQ
3.678138986		0 DER
3.961843753		0 DER
2.999999999		0 DER
7.497594586		1 DER
4.962293295		1 DER
7.444542326		1 DER
10.12493903		1 DER
8.682257354		1 DER

Para el grupo IZQ, las proporciones son:
- Y=0 : 1/1 = 1
- Y=1 : 0/1 = 0

Para el grupo DER, las proporciones son:
- Y=0 : 4/9 = 0.4444
- Y=1 : 5/9 = 0.5555

$$Gini(X1 = 2.7712) = (1 - (\frac{1^2}{1} + \frac{0^2}{1})) \times \frac{1}{10} + (1 - (\frac{4^2}{9} + \frac{5^2}{9})) \times \frac{9}{10}$$

44

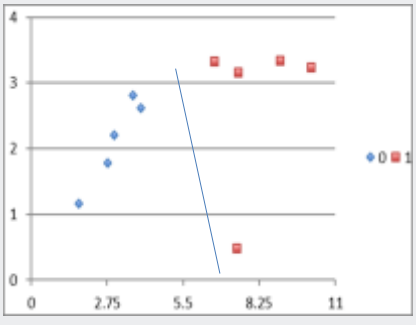
44

Cuentas de Clases					
	IZQ	DER	PADRE		
Y=0	1	4	5		
Y=1	0	5	5		
CUENTAS	1	9	10		
Gini					
	IZQ	DER	Gini IZQ	Gini DER	Gini izq
Y=0	1	0.444444444	0	0.44444444	0.44444444
Y=1	0	0.555555556			
Peso	0.1	0.9			

45

45

Mejor punto de división candidato



46

46

Mejor punto de división candidato

- Un valor possible de X1 sería entonces el del ultimo ejemplo, X1 = 6.6422.
  - Si X1 < 6.6422 entonces IZQ
  - Si X1 >= 6.6422 entonces DER
- Aplica el procedimiento anterior para agrupar las instancias y calcular el índice Gini correspondiente.

47

47

CART - Árbol de clasificación

- Un árbol de decisión indica como valor de predicción de una hoja a la media de los valores que caen en ella.
- Cómo mejorarlo?

48

Revisión de parámetros – criterios de división ("split")

- Para variables objetivo categóricas:
  - Ganancia de información (1 - entropía)
  - Tasa o relación de ganancia
  - Impureza Gini (1 – índice Gini)
  - Chi-cuadrado
- Para variables objetivo continuas:
  - Exactitud, varianza
- ¿Cuáles son los operadores de RapidMiner que implementan estas variaciones?
- Ver buen tutorial resumen en https://www.analyticsvidhya.com/blog/2020/06/4-ways-split-decision-tree/

49

TA3 – EJ2 – Arbol de decisión de regresión en

- Utilizar el dataset “housing” de predicción de mediana de valor de casas
- Analizar los atributos, estadísticas, variable objetivo
- Crear un proceso con cross validation para entrenar un árbol de decision
- Agregar el DT, apply model y performance (regression).
- Seleccionar los parámetros apropiados para el DT y los indicadores del performance
- Ejecutar y estudiar el modelo generado

50

Revisión de parámetros que podemos optimizar

- Criterios para “split”
  - Ganancia de información (entropía)
  - Tasa o relación de ganancia
  - Índice Gini
  - Exactitud
- Profundidad máxima del árbol
- Ganancia minima para split
- Tamaño mínimo para split
- Tamaño mínimo de hoja

51

TA3 – EJ3 – Optimización de parámetros del Arbol de Decisión

- Utilizar el proceso RM desarrollado en el TA3- EJ1
- Revisar el context, los atributos, estadísticas, variable objetivo
- Agregar un operador (subproceso) “Optimize parameters”
- Dentro de éste, agregar un operador cross validation para entrenar el árbol de decision, mover los bloques que correspondan al mismo,, apply model y performance (classification).
- Agregar un operador “log” y configurar para que tenga al menos iteraciones, performance, y los parámetros objeto de optimización
- Seleccionar los parámetros a optimizar y los intervalos de evaluación (**estimar la cantidad de iteraciones totales que se harán!!!!**)
- Registrar los resultados, modelo y rendimiento final alcanzado.

https://academy.rapidminer.com/learn/video/optimization-of-the-model-parameters

52

Demo AutoModel para observar la optimización

53