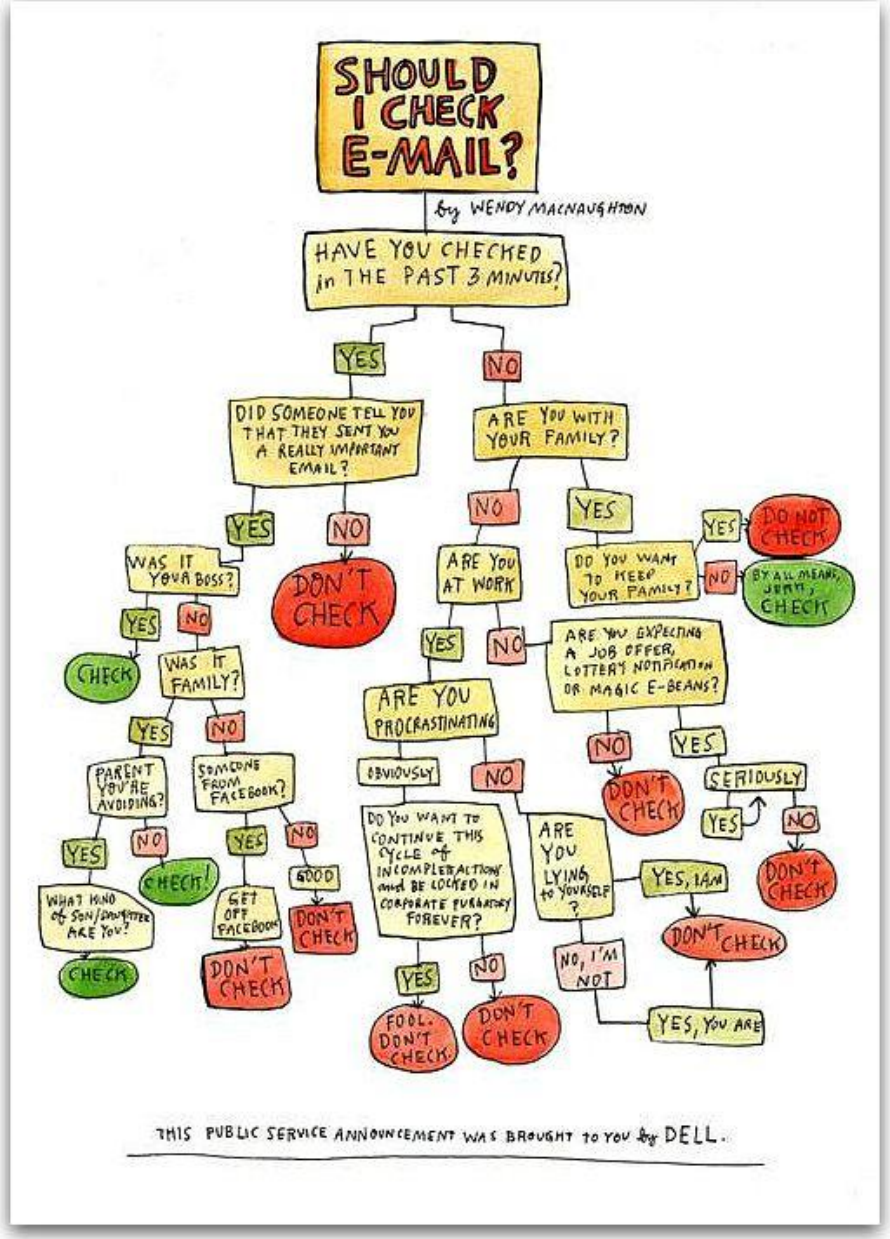
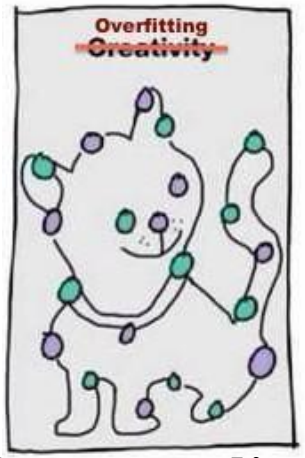
APRENDIZAJE AUTOMÁTICO



Clases anteriores

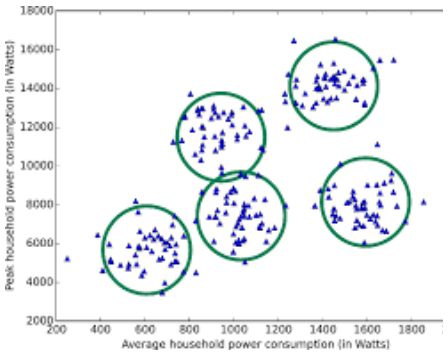
Aprendizaje automático

Protocolos

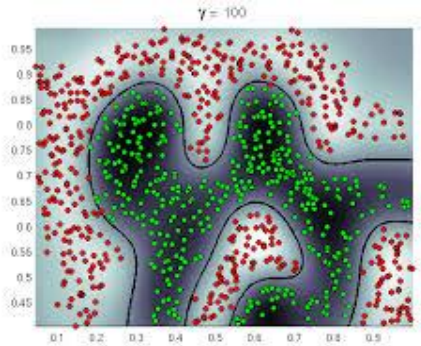


Sobre aprendizaje (Overfitting)

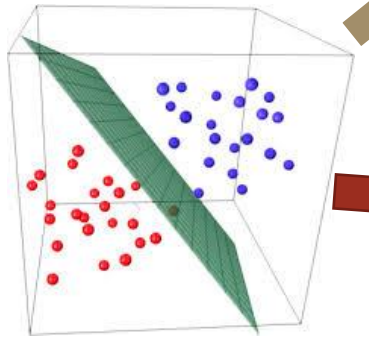
Métricas de Evaluación de la clasificación



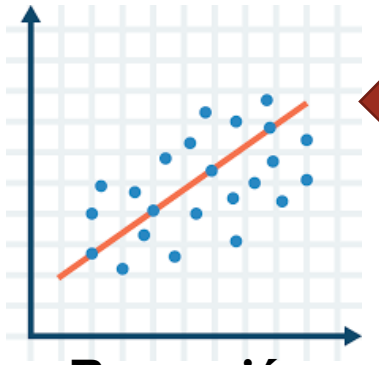
Aprendizaje no supervisado



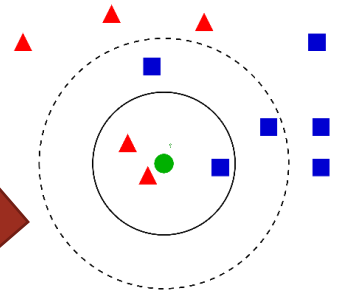
Aprendizaje supervisado



Clasificación



Regresión



KNN

EJEMPLO DE USO: PROSPECCIÓN DE CLIENTES

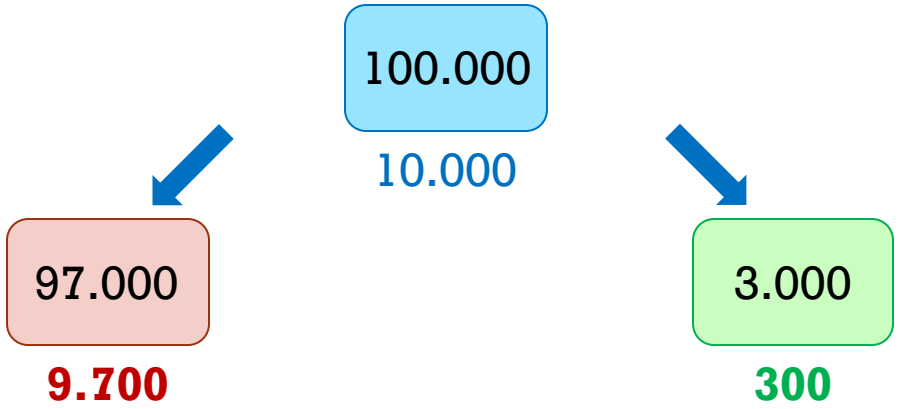
Una compañía de seguros quiere contactar los mejores clientes potenciales de una base de datos de **100.000** personas que acaban de adquirir para ofrecerles un plan. Cuentan con la información de campañas anteriores incluyendo diferentes características como edad, género y salarios, así como la indicación de si la oferta fue exitosa o no.

Teniendo en cuenta el costo del paquete de publicidad por correo, solo pueden contactar **10.000** clientes potenciales.

Sabemos de las campañas anteriores, que solo el **3%** de las personas contactadas compraban el plan, pero esta tasa varía considerablemente si empezamos a considerar sub poblaciones con características particulares (edad, …).

EJEMPLO DE USO: PROSPECCIÓN DE CLIENTES

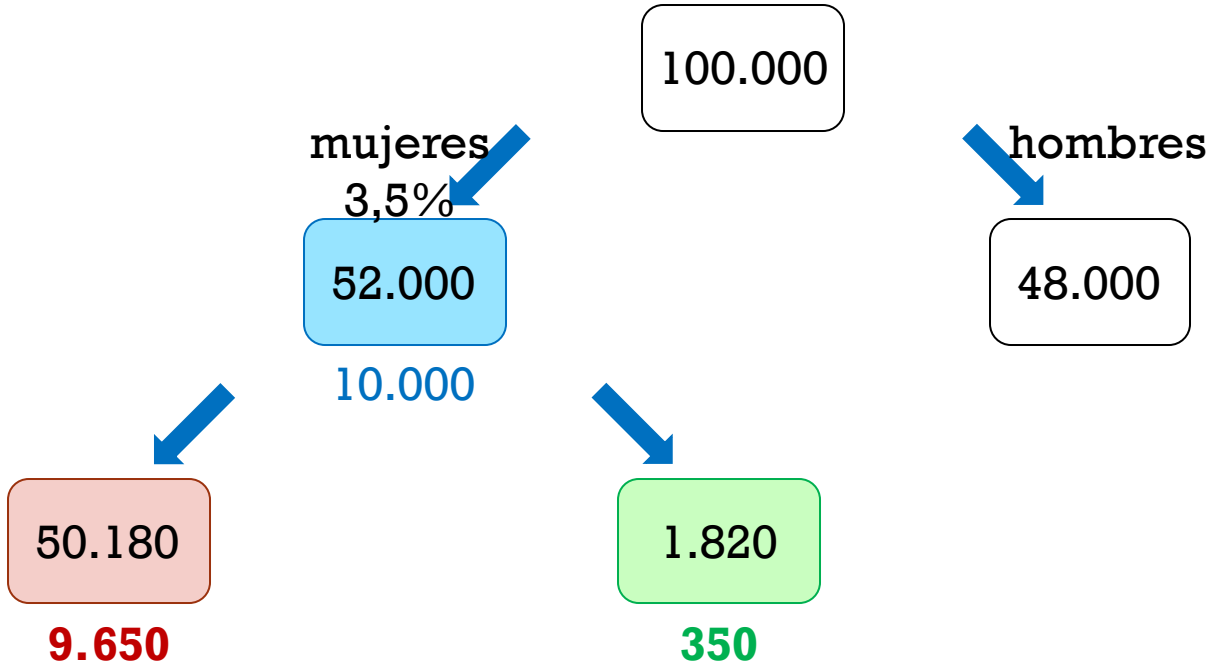
Escoja 10.000 clientes potenciales aleatoriamente



éxito: 3% (300)

EJEMPLO DE USO: PROSPECCIÓN DE CLIENTES

Las mujeres son más propensas a comprar seguros (3,5%) y hay 52.000 mujeres en la BD

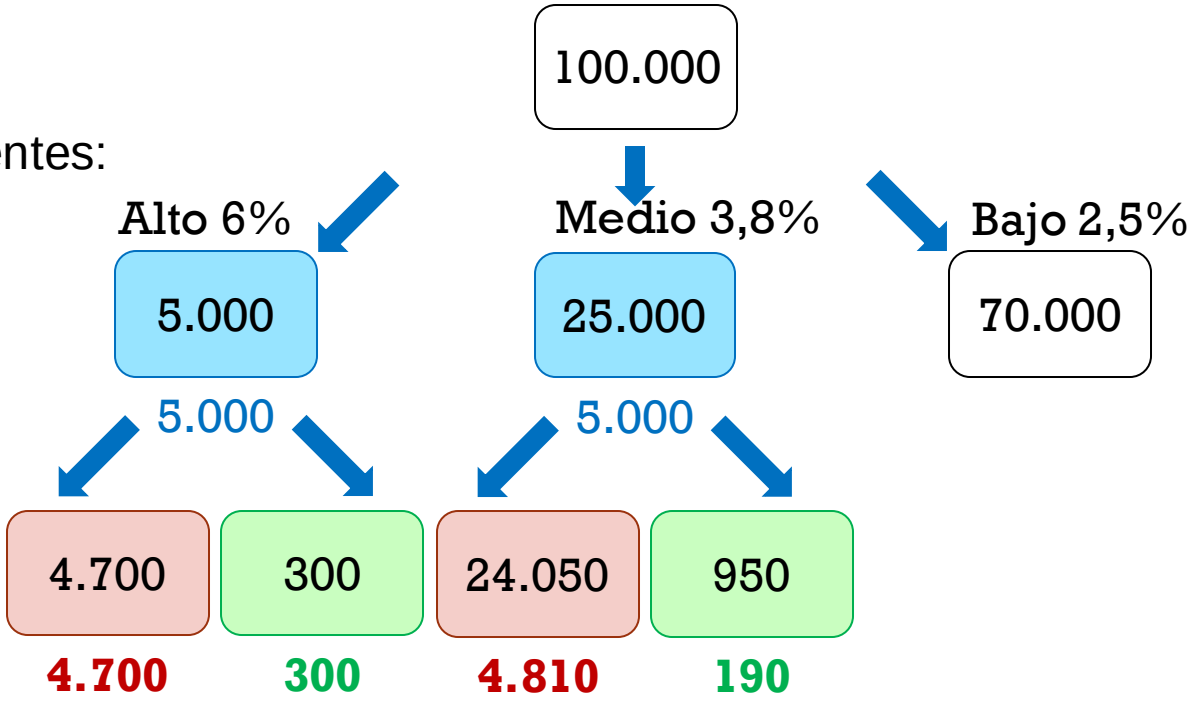


éxito: 3,5% (350)

EJEMPLO DE USO: PROSPECCIÓN DE CLIENTES

Las tasas de éxito por tipo de salario son las siguientes:

- Alto (5.000 en el grupo): 6%
- Medio (25.000 en el grupo): 3,8%
- Bajo (70.000 en el grupo): 2,5%



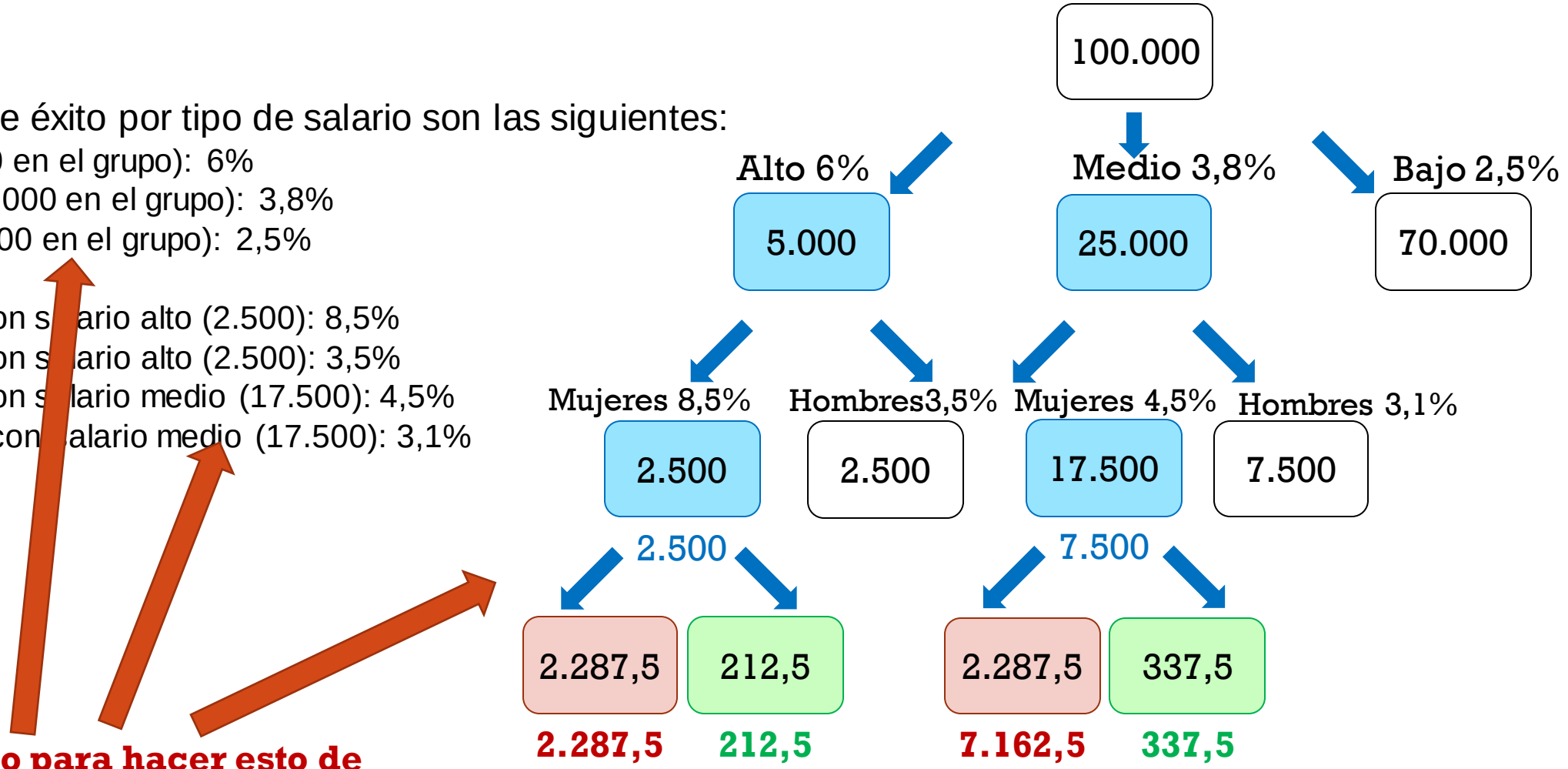
éxito: 4,9% (490)

EJEMPLO DE USO: PROSPECCIÓN DE CLIENTES

Las tasas de éxito por tipo de salario son las siguientes:

Alto (5.000 en el grupo): 6%
Medio (25.000 en el grupo): 3,8%
Bajo (70.000 en el grupo): 2,5%

Mujeres con salario alto (2.500): 8,5%
Hombre con salario alto (2.500): 3,5%
Mujeres con salario medio (17.500): 4,5%
Hombres con salario medio (17.500): 3,1%

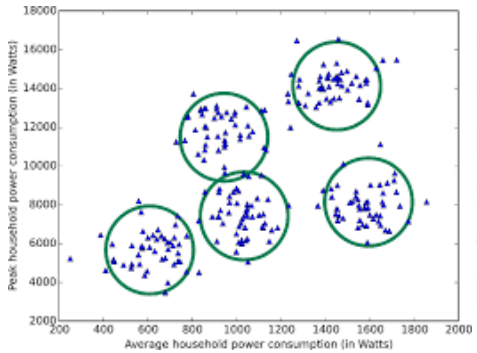


¿Cómo hago para hacer esto de una manera más inteligente?

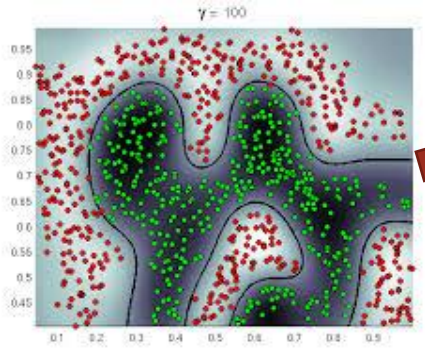
éxito: 5,5% (550)

AGENDA

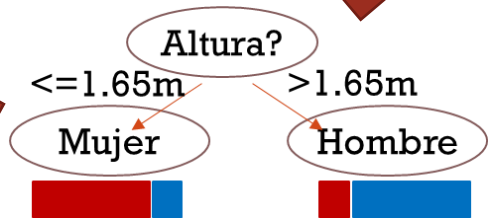
Aprendizaje automático



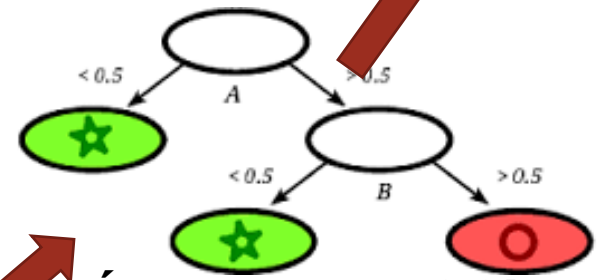
Aprendizaje no supervisado



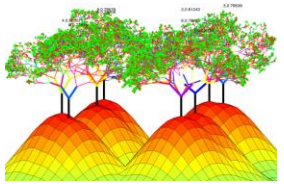
Aprendizaje supervisado



Decision Stump



Árboles de decisión



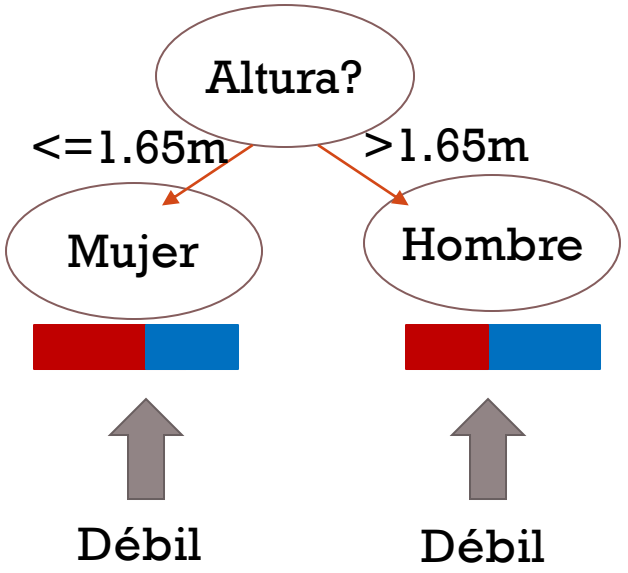
Random forest

Poda

DECISION STUMP

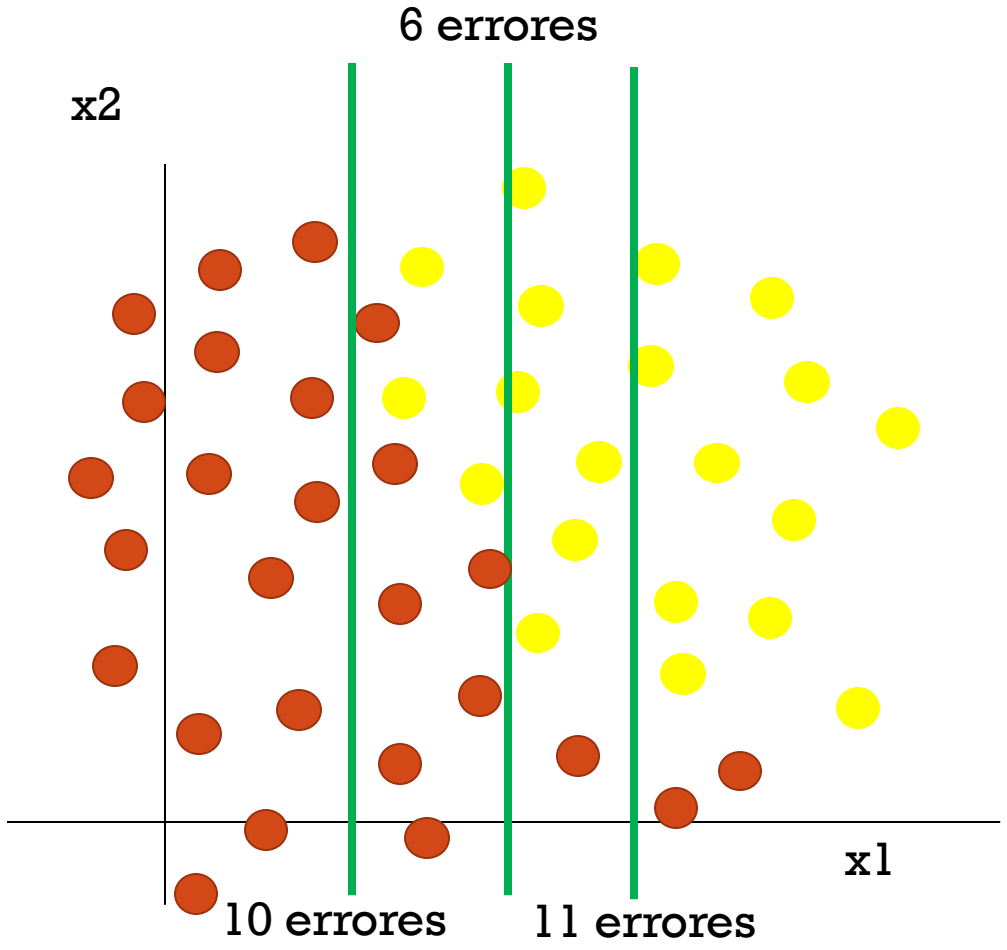
DECISION STUMP

- Busca el mejor particionamiento considerando **1 sola variable** predictiva
- Árbol de decisión de un solo nivel
- Es un "**very weak learner**", que produce una sola regla de decisión. Por ejemplo:
  - Las personas que miden mas de 1.65 metros son hombres, y las que no, mujeres
  - Los que consiguen trabajo en data science ganan 30% más que los que no
- Muy utilizado en modelos de ensamble (sobre todo **Boosting**)



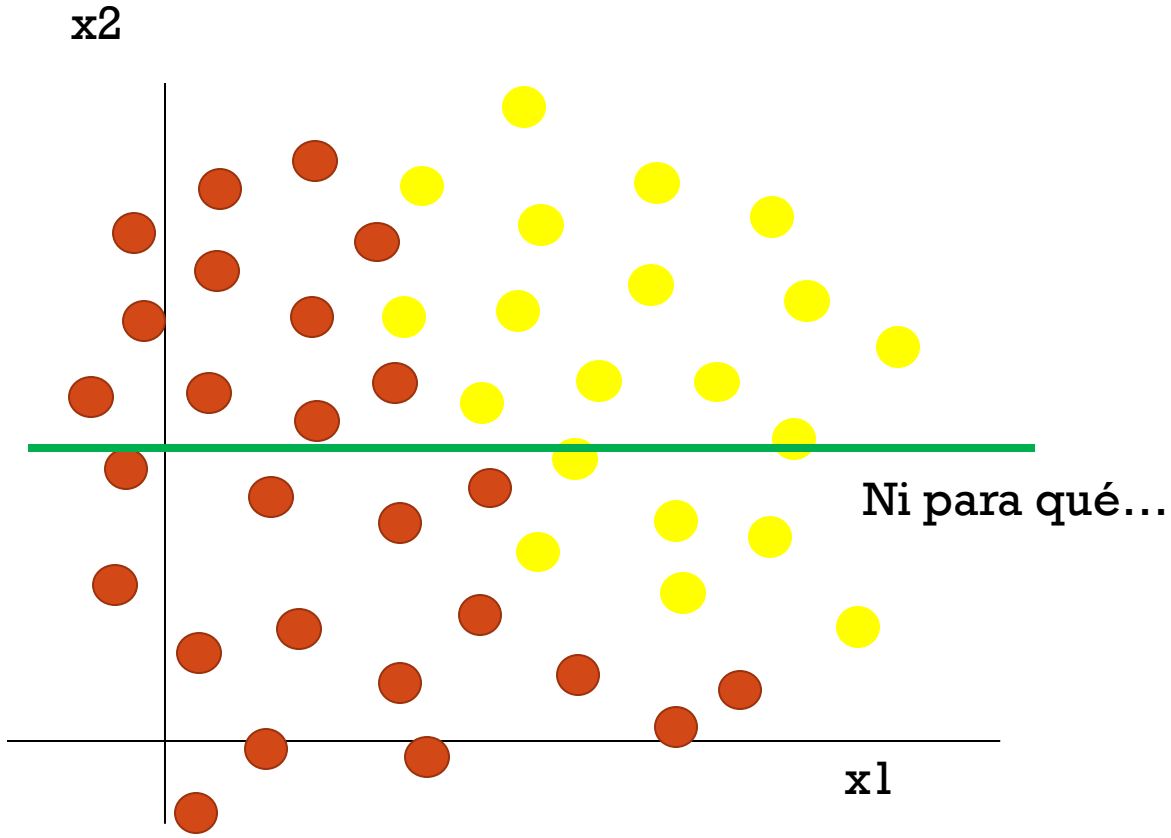
DECISION STUMP

- El particionamiento en las variables numéricas solo se puede realizar de manera perpendicular a los ejes
  - Se busca minimizar el error de clasificación/regresión
  - Es necesario buscar todos los particionamientos posibles en todas las variables predictivas
- ¿Cómo serían las reglas y el número de errores de los clasificadores siguientes?



DECISION STUMP

- El particionamiento en las variables numéricas solo se puede realizar de manera perpendicular a los ejes
  - Se busca minimizar el error de clasificación/regresión
  - Es necesario buscar todos los particionamientos posibles en todas las variables predictivas
- ¿Cómo serían las reglas y el número de errores de los clasificadores siguientes?



DECISION STUMP

- Las variables predictivas numéricas deben ser discretizadas
- Hay varias maneras de realizar el análisis del mejor punto de corte, utilizando diferentes métricas:
  - Ganancia o ratio de información (entropía)
  - Gini
  - CHAID
- Más adelante haremos un taller al respecto con la entropía condicional

¿Cuál particionamiento es mejor entre p1 y p2?

humidity	play (X)	p1	p2
54	yes	a	a
58	no	a	a
59	yes	a	a
60	yes	a	a
60	yes	a	a
62	yes	a	a
63	yes	b	a
80	yes	b	a
81	yes	b	a
89	no	b	b
90	no	b	b
90	no	b	b
90	no	b	b
92	yes	b	b

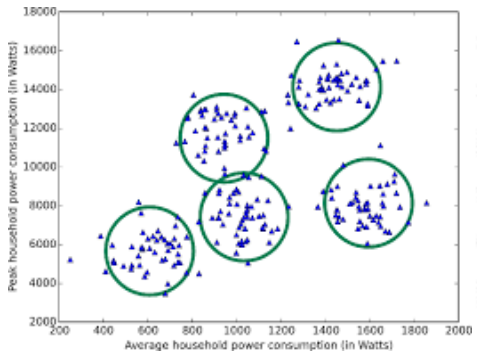
CLASIFICACIÓN
TALLER: CHURN DE CLIENTES

- DATASET: base de datos de 20000 clientes que han cancelado (churn) o no los servicios de una compañía. La idea es poder predecir en un futuro quiénes son los clientes más propensos a hacer churn, para poder desarrollar campañas que lo prevengan.
- En EXCEL definir el baseline de clasificación para el atributo objetivo LEAVE
- Encontrar particionamientos que permitan mejorar la tasa de correctitud del baseline

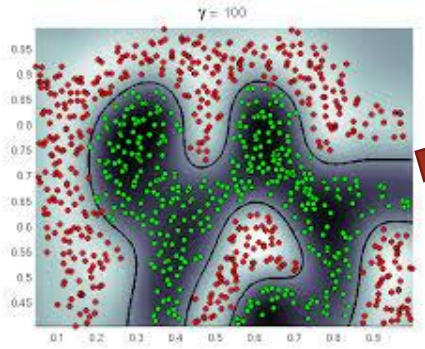
Descarguen los archivos del taller de clasificación de churn de clientes y ejecuten cada una de las 3 partes, que vamos a ir revisando una por una.

AGENDA

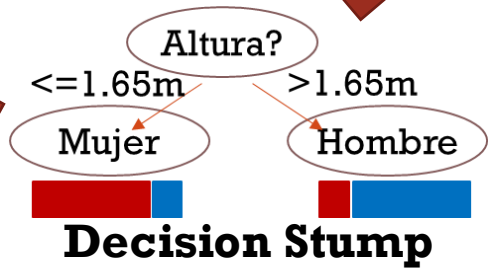
Aprendizaje automático



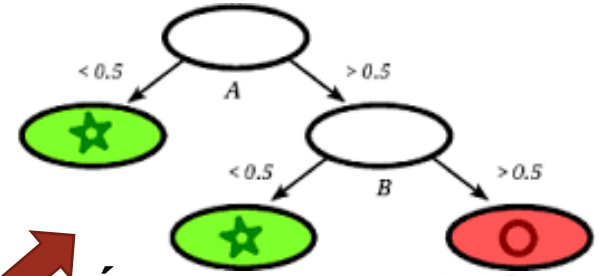
Aprendizaje no supervisado



Aprendizaje supervisado



Decision Stump

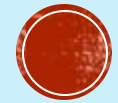


Árboles de decisión

ÁRBOLES DE DECISIÓN

TALLER DE PARTICIONAMIENTO

- Clasificador humano

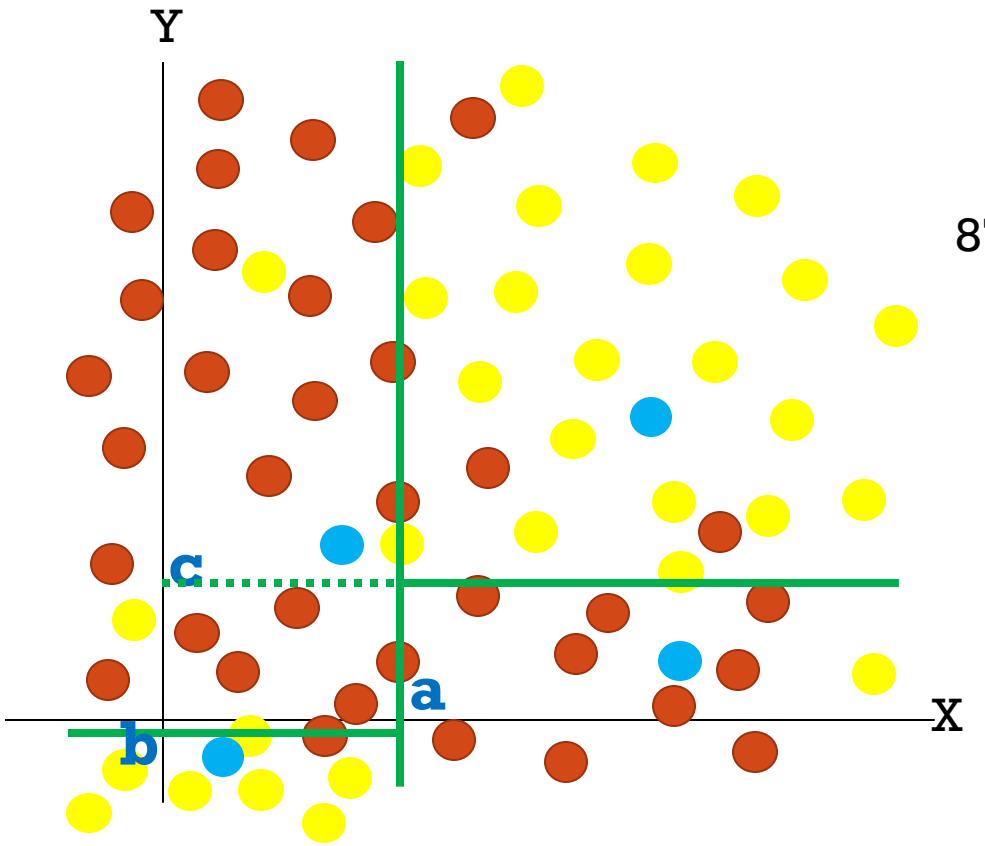


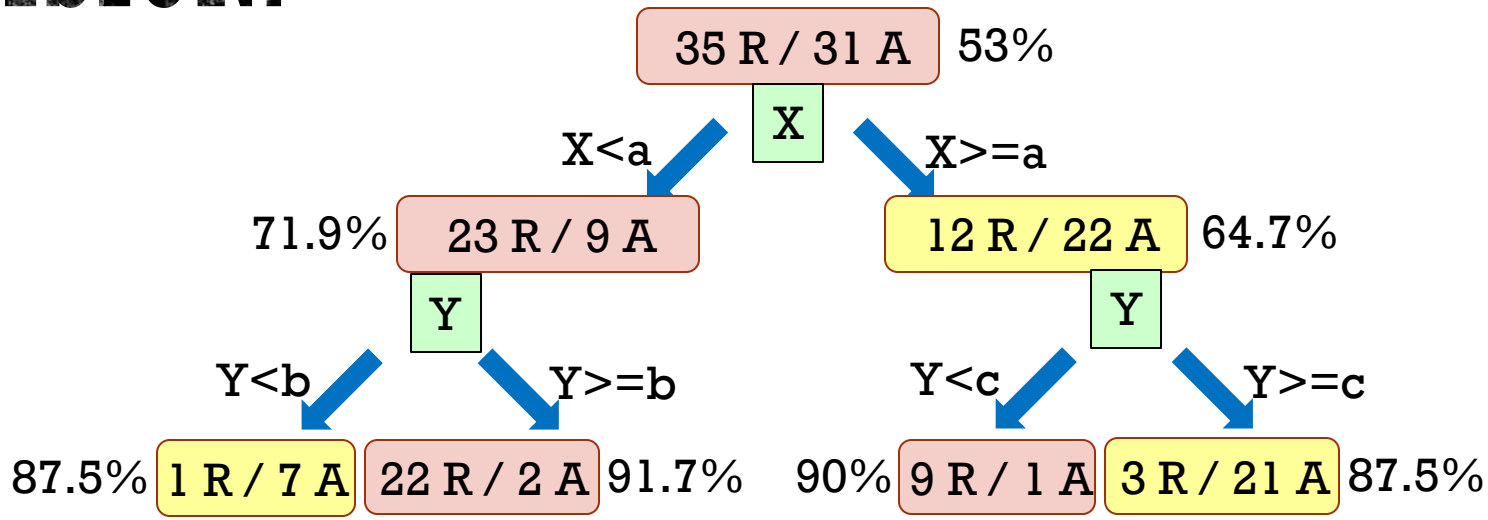
ÁRBOLES DE DECISIÓN: ALGORITMO

Dividir & conquistar: se divide de manera incremental el espacio en regiones no sobrelapadas, que constituyen los nodos del árbol:

- **Seleccionar factor** que mejor separa los valores objetivo del nodo actual, crear una rama por cada valor, que minimiza una función de impureza del nodo en cuestión
- **Dividir** el conjunto de datos del nodo con respecto a los valores del factor seleccionado y crear los nodos correspondientes
- **Repetir recursivamente** hasta que
  - todas las instancias de los nodos hoja sean de la misma clase
  - no hayan mas atributos por los cuales particionar
  - se llegue a un criterio de parada definido (pre-poda)

ÁRBOLES DE DECISIÓN: CLASIFICACIÓN

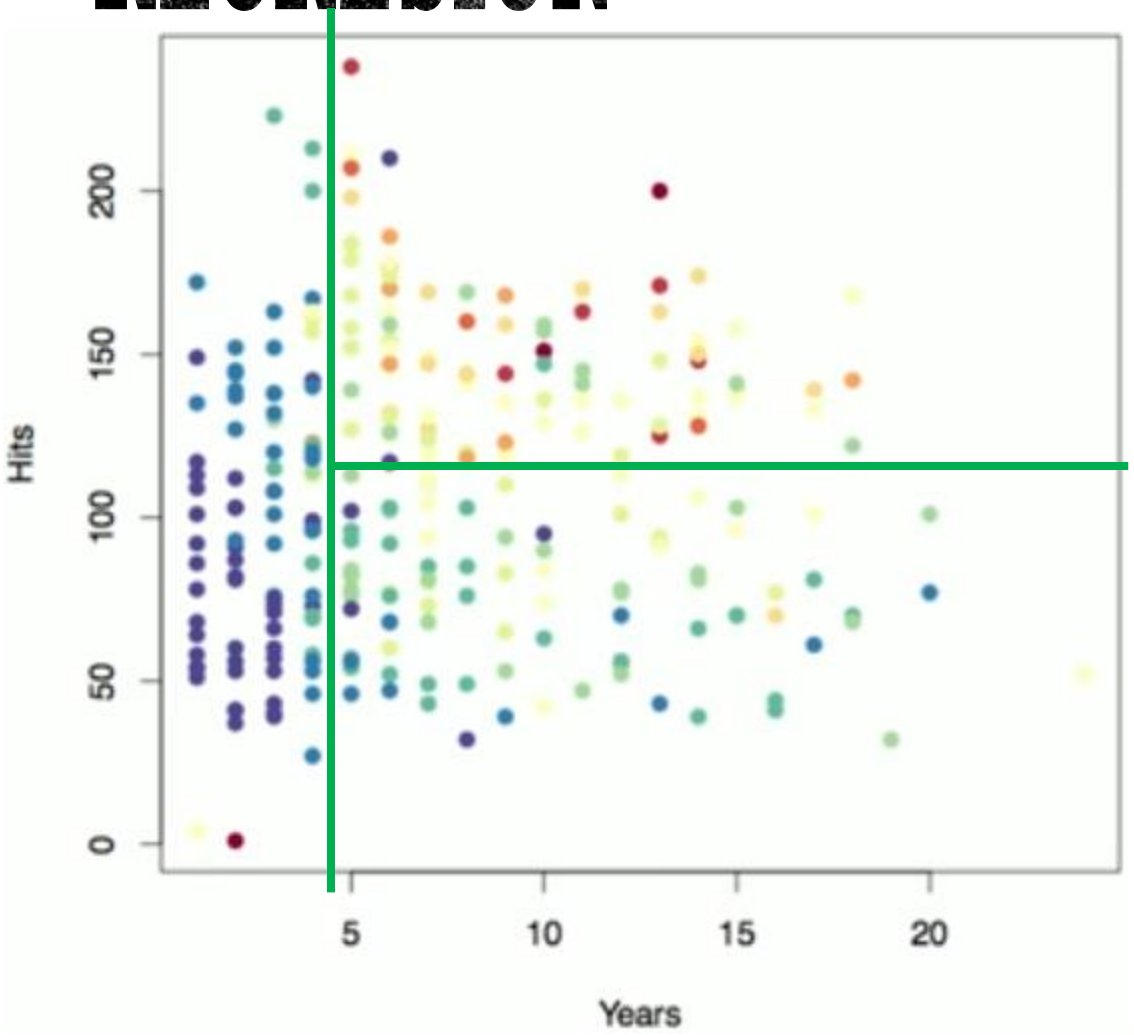




Paso	Accuracy
Raíz	35/66 = 53%
1era partición	45/66 = 68.2%
2a partición (rama izq.)	51/66 = 77.3%
3a partición (rama der.)	59/66 = 89.4%

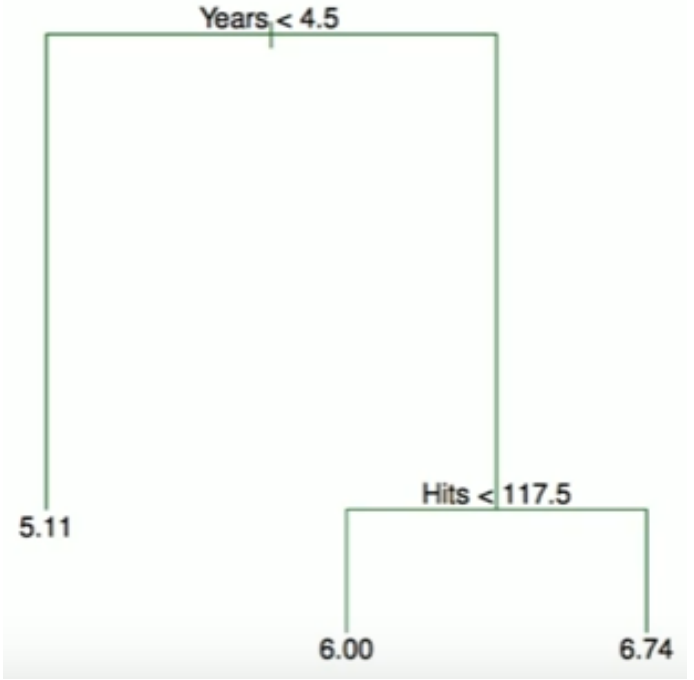
Se minimiza localmente una función de costo que considera la **impureza** de los nodos terminales del árbol

ÁRBOLES DE DECISIÓN: REGRESIÓN



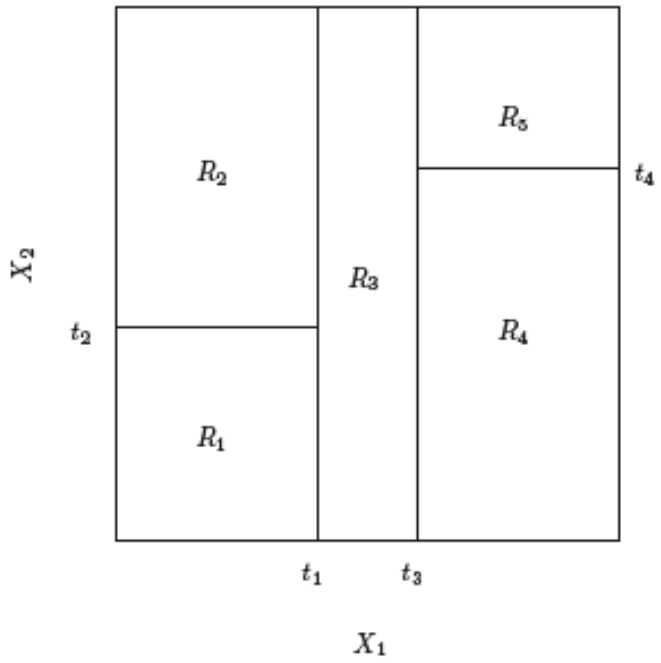
Evolución de lo salarios de beisbolistas (color) con respecto a años de experiencia (abscisa) y número de bateos exitosos (ordenada).

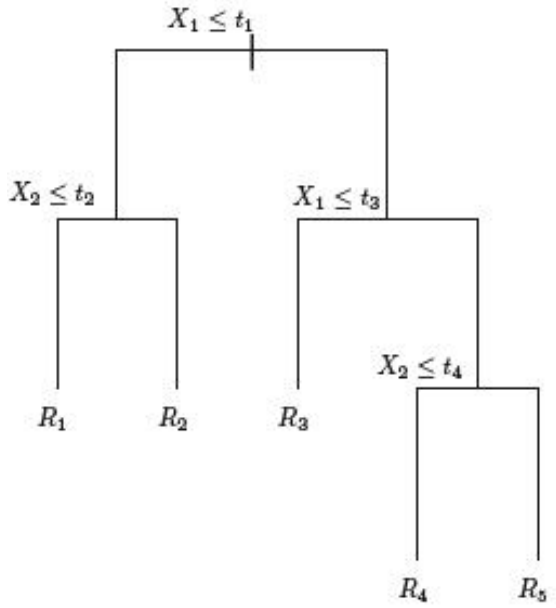
Se minimiza localmente $\sum_{j=1}^{J} \sum_{i \epsilon R_j} (y_i - \hat{y}_{R_j})^2$

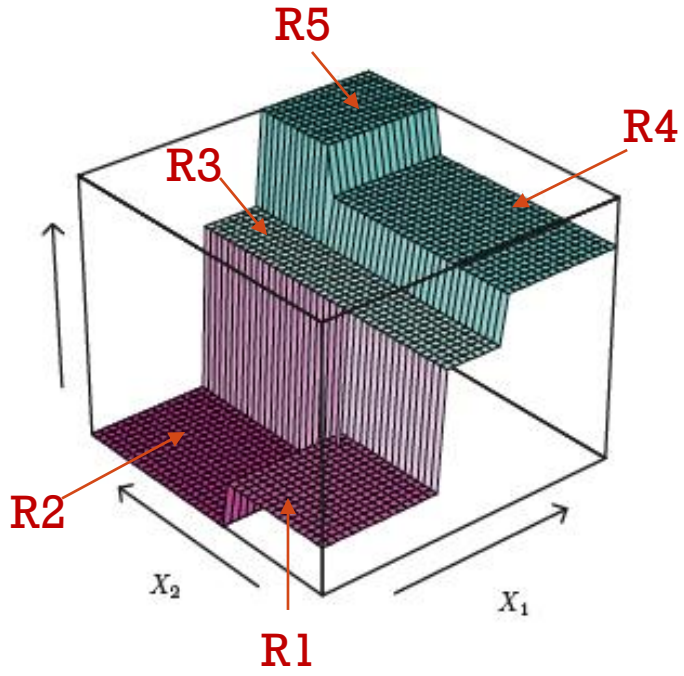


ISLR, 2013

ÁRBOLES DE DECISIÓN: REGRESIÓN







ISLR, 2013

ÁRBOLES DE DECISIÓN

- **Aprendizaje inductivo**: generalización
- Algoritmo **greedy**: busca óptimos locales a cada etapa, que no son necesariamente los óptimos globales
- **Simple** de comprender, implementar y explotar
- Puede ser usado para **clasificación** y **regresión**
- Clasificador **no lineal** (considera interacciones entre los factores)
- Mejor **performance** en contextos no lineales
- Tamaño variable, **escalable** (BIG DATA)

ÁRBOLES DE DECISIÓN

- Los datos deben ser **categóricos**, no se puede definir una distancia de manera natural
- Un árbol de decisión se puede representar como un conjunto de **reglas** booleanas
- Una nueva instancia puede ser clasificada **siguiendo las ramas** del árbol
- Ideal para los casos en que un pequeño número de atributos provee una gran cantidad de la información
- Prueba diferentes atributos categóricos para aprender una clase. Variables continuas deben ser **discretizadas**.
- No se basa en ninguna noción de distancia, el modelo es **indiferente** a nociones de **normalización**

ÁRBOLES DE DECISIÓN

- **Existen diferentes criterios para determinar el mejor atributo en cada nodo:** Varios algoritmos, con diferentes criterios de división
  - **CART**: Classification and regression trees. Sólo particiones binarias, usando la métrica de impureza Gini para la clasificación y la reducción de varianza para la regresión
  - **ID3**, basado en ganancia de información y entropía como criterio de división
  - **C4.5** (**C5.0** y **J48** en WEKA), extensión de ID3. Considera atributos continuos y discretos, información faltante, diferentes costos de clasificación y poda
  - **CHAID**: Chi-squared Automatic Interaction Detector. Utiliza la métrica Chi cuadrado para la clasificación y pruebas F para la regresión
  - …

ÁRBOLES DE DECISIÓN: ID3

Utiliza métricas de la **teoría de información**

- Seleccionar el atributo que más reduce el desorden en la variable objetivo del dataset
  - Entropía:

$$H(Y) = -\sum_i p(Y = y_i) * log_2\big(p(Y = y_i)\big)$$

$$H(Y) = 0, si\ no\ hay\ erroers\ de\ clasificación$$

  - Ent.Cond.

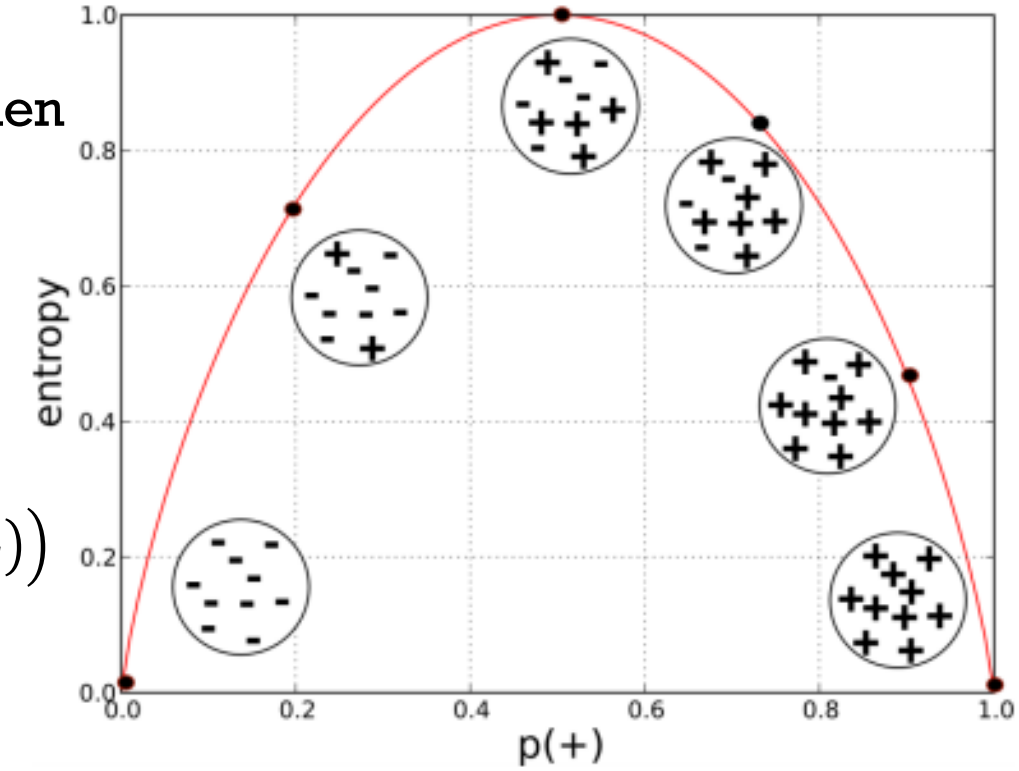
$$H\big(Y|X = x_j\big) = -\sum_i p\big(Y = y_i|X = x_j\big) * log_2\Big(p\big(Y = y_i|X = x_j\big)\Big)$$

  - Ent.Cond.Prom.

$$H(Y|X) = \sum_j p\big(X = x_j\big) * H(Y|X = x_j)$$

  - Ganancia de información

$$Gain\big(Y,\ X = x_j\big) = H(Y) - \sum_j p\big(X = x_j\big) * H(Y|X = x_j)$$



Provost & Fawcett, 2013

TALLER: ÁRBOLES DE DECISIÓN: ID3

	outlook	temperature	humidity	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Dataset de clima: 14 instancias, 4 variables independientes para predecir una clase con 2 categorías posibles

¿cómo escojo el mejor atributo para particionar?

1. Calcular la entropía de la clase (“play”)

Play (Y)				H
p(Y=no)	35.7%	-p(Y=no) log p(Y=no)	0.53	
p(Y=yes)	64.3%	-p(Y=yes) log p(Y=yes)	0.41	0.940

2. Calcular la entropía condicional para cada atributo y su ganancia de información

Outlook								GAIN
p(sunny)	35.7%	p(yes|sunny)	40.0%	p(no|sunny)	60.0%	0.971		
p(overcast)	28.6%	p(yes|overcast)	100.0%	p(no|overcast)	0.0%	0.000	0.694	0.247
p(rainy)	35.7%	p(yes|rainy)	60.0%	p(no|rainy)	40.0%	0.971		

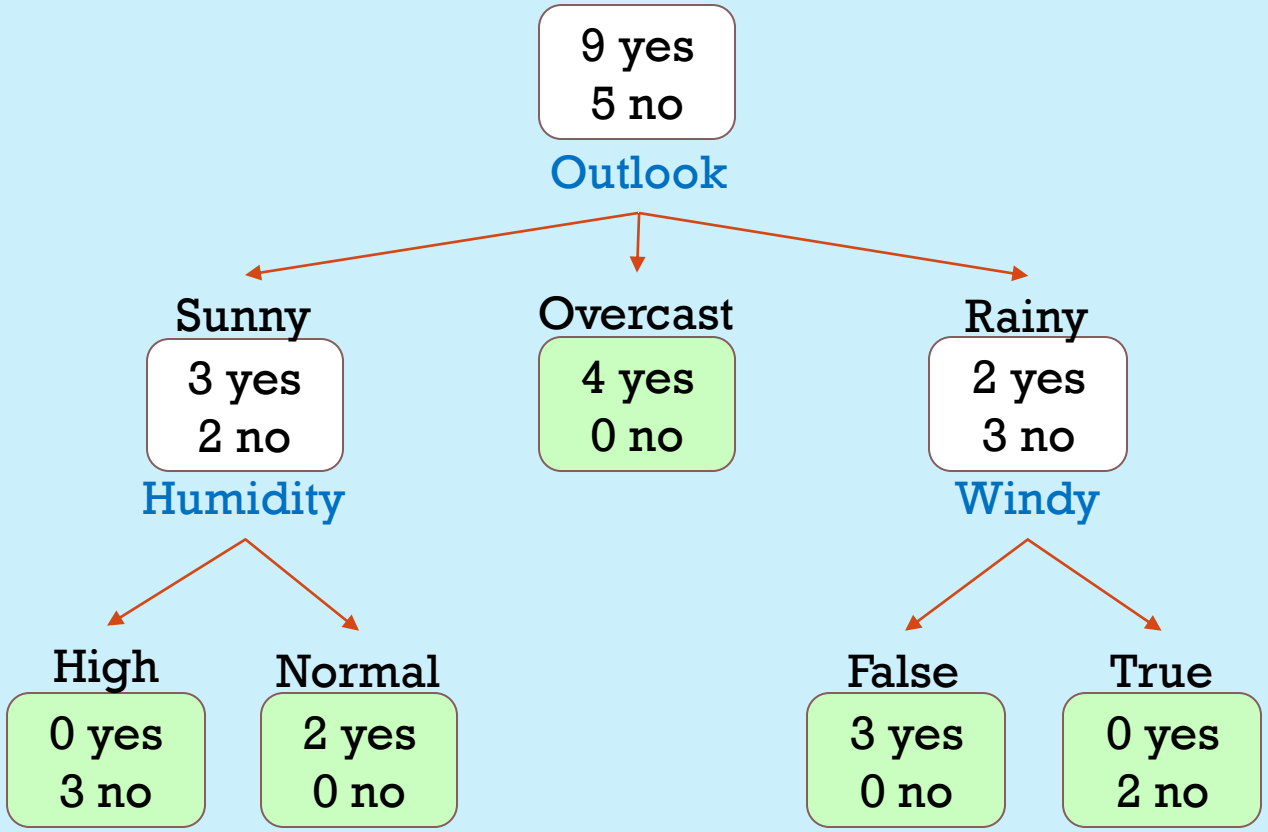
Temperature								GAIN
p(hot)	28.6%	p(yes|hot)	50.0%	p(no|hot)	50.0%	1.000		
p(mild)	42.9%	p(yes|mild)	66.7%	p(no|mild)	33.3%	0.918	0.911	0.029
p(cool)	28.6%	p(yes|cool)	75.0%	p(no|cool)	25.0%	0.811		

Humidity								GAIN
p(normal)	50.0%	p(yes|normal)	85.7%	p(no|normal)	14.3%	0.592	0.788	0.152
p(high)	50.0%	p(yes|high)	42.9%	p(no|high)	57.1%	0.985		

Windy								GAIN
p(FALSE)	57.1%	p(yes|W=FALSE)	75.0%	p(no|W=FALSE)	25.0%	0.811	0.892	0.048
p(TRUE)	42.9%	p(yes|W=TRUE)	50.0%	p(no|W=TRUE)	50.0%	1.000		

3. Particionar según el atributo con mayor ganancia de información
4. Parar si todas las hojas son puras o ya no hay mas atributos

TALLER: ÁRBOLES DE DECISIÓN: ID3



TALLER: ÁRBOLES DE DECISIÓN: ID3

ID	outlook	temperature	humidity	windy	play (X)	p1	p2
5	rainy	cool	54	FALSE	yes	a	a
6	rainy	cool	58	TRUE	no	a	a
10	rainy	mild	59	FALSE	yes	a	a
7	overcast	cool	60	TRUE	yes	a	a
9	sunny	cool	60	FALSE	yes	a	a
11	sunny	mild	62	TRUE	yes	a	a
13	overcast	hot	63	FALSE	yes	b	a
3	overcast	hot	80	FALSE	yes	b	a
12	overcast	mild	81	TRUE	yes	b	a
2	sunny	hot	89	TRUE	no	b	b
14	rainy	mild	90	TRUE	no	b	b
1	sunny	hot	90	FALSE	no	b	b
8	sunny	mild	90	FALSE	no	b	b
4	rainy	mild	92	FALSE	yes	b	b

Ahora el atributo "humidity" es numérico.

¿Cómo encuentro el mejor particionamiento para ?

1. Consider every possible binary partition and choose the one with the highest information gain (here we only try out 2)
2. Calculate the class entropy

Play (Y)				H
p(Y=no)	35.7%	Y=no) log p(Y=r	0.53	
p(Y=yes)	64.3%	Y=yes) log p(Y=y	0.41	0.940

3. Calculate the conditional entropy for each partition and their information gain

P1								GAIN
p(a)	42.9%	p(yes|a)	83.3%	p(no|a)	16.7%	0.650	0.850	0.090
p(b)	57.1%	p(yes|b)	50.0%	p(no|b)	50.0%	1.000		

P2								GAIN
p(a)	64.3%	p(yes|a)	88.9%	p(no|a)	11.1%	0.503	0.581	0.359
p(b)	35.7%	p(yes|b)	20.0%	p(no|b)	80.0%	0.722		

4. Select the one with the highest information gain

ÁRBOLES DE DECISIÓN: CART

- Solo árboles con particionamientos binarios
- Gini como criterio de impureza para el particionamiento:
  - 0 pureza perfecto: todas las instancias de la misma clase
  - 0.5 impureza: distribución equitativa de las instancias entre ambas clases
- Algoritmo
  - Para cada atributo
    - Para cada posible split binario del atributo
      - Calcular el Gini para ambos subnodos

        $gini = \sum p * (1 - p) = 1 - \sum p^2,$ donde p es la probabilidad de cada clase.
      - Calcular el promedio ponderado del Gini de las particiones
    - Seleccionar el split binario con el mayor promedio de Gini
  - Seleccionar el atributo con el mayor promedio de Gini

ÁRBOLES DE DECISIÓN: CART

	outlook	temperature	humidity	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

n	yes	p	w	Split	Gini	Avg.Gini
5	2	40.0%	35.7%	sunny	0.480	0.394
9	7	77.8%	64.3%	(overcast | rainy)	0.346	

n	yes	p	w	Split	Gini	Avg.Gini
4	4	100.0%	28.6%	overcast	0.000	0.357
10	5	50.0%	71.4%	(sunny | rainy)	0.500	

n	yes	p	w	Split	Gini	Avg.Gini
5	3	60.0%	35.7%	rainy	0.480	0.457
9	6	66.7%	64.3%	(sunny | overcast)	0.444	

n	yes	p	w	Split	Gini	Avg.Gini
4	2	50.0%	28.6%	hot	0.500	0.443
10	7	70.0%	71.4%	(mild | cool)	0.420	

n	yes	p	w	Split	Gini	Avg.Gini
6	4	66.7%	42.9%	mild	0.444	0.458
8	5	62.5%	57.1%	(hot | cool)	0.469	

n	yes	p	w	Split	Gini	Avg.Gini
4	3	75.0%	28.6%	cool	0.375	0.450
10	6	60.0%	71.4%	(hot | mild)	0.480	

n	yes	p	w	Split	Gini	Avg.Gini
7	6	85.7%	50.0%	normal	0.245	0.439
7	3	42.9%	50.0%	high	0.633	

n	yes	p	w	Split	Gini	Avg.Gini
8	6	75.0%	57.1%	FALSE	0.375	0.429
6	3	50.0%	42.9%	TRUE	0.500	

ÁRBOLES DE DECISIÓN: CHAID

- Particiones en 2 o más subconjuntos
- Chi cuadrado como criterio de particionamiento: significancia estadística de las diferencias entre los nodos hijos y el nodo padre
- Algoritmo
  1. Para cada atributo
     1. Calcular el Chi cuadrado para cada nodo hijo

$$\chi^2 = \frac{(Observado - Esperado)^2}{Esperado}$$

     2. Suma de los valores de Chi cuadrado, cálculo del valor-p correspondiente
  2. Seleccionar el atributo con el menor valor-p
  3. Si el valor-p es inferior a cierto umbral (i.e. 5%), particionar. Sino, parar

DECISION TREES: CHAID

	outlook	temperature	humidity	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Outlook	play		
Observed	yes	no	
sunny	2	3	5
overcast	4	0	4
rainy	3	2	5
	9	5	14
Expected			
sunny	3.2	1.8	5.0
overcast	2.6	1.4	4.0
rainy	3.2	1.8	5.0
	9.0	5.0	14.0
CHI2			
sunny	0.46	0.83	1.28
overcast	0.79	1.43	2.22
rainy	0.01	0.03	0.04
	1.27	2.28	3.55
			0.170

Temperature	play		
Observed	yes	no	
hot	2	2	4
mild	4	2	6
cool	3	1	4
	9	5	14
Expected			
hot	2.6	1.4	4.0
mild	3.9	2.1	6.0
cool	2.6	1.4	4.0
	9.0	5.0	14.0
CHI2			
hot	0.13	0.23	0.36
mild	0.01	0.01	0.01
cool	0.07	0.13	0.20
	0.20	0.37	0.57
			0.752

Humidity	play		
Observed	yes	no	
high	3	4	7
normal	6	1	7
	9	5	14
Expected			
hot	4.5	2.5	7.0
normal	4.5	2.5	7.0
	9.0	5.0	14.0
CHI2			
hot	0.50	0.90	1.40
normal	0.50	0.90	1.40
	1.00	1.80	2.80
			0.0943

Windy	play		
Observed	yes	no	
FALSE	6	2	8
TRUE	3	3	6
	9	5	14
Expected			
FALSE	5.1	2.9	8.0
TRUE	3.9	2.1	6.0
	9.0	5.0	14.0
CHI2			
hot	0.14	0.26	0.40
normal	0.19	0.34	0.53
	0.33	0.60	0.93
			0.334

REFERENCIAS

- *Introduction to Statistical Learning with Applications in R (ISLR), G. James, D. Witten, T. Hastie & R. Tibshirani, 2014*
- *Data Science for Business*, Foster Provost & Tom Fawcett, O'Reilly, 2013
- *Machine Learning,* Tom M. Mitchell, McGraw-Hill, 1997
- *Overfitting in decisión trees*, Carlos Guestrin, University of Washington
- *Python Machine Learning*, Sebastian Raschka, Packt, 2015