TALLER DE EVALUACIÓN DE UN MODELO DE CLASIFICACIÓN (ÁRBOLES)

- 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.
- Encontrar particionamientos que permitan mejorar la tasa de correctitud del baseline

UNIVERSIDAD ICESI

TALLER DE DECISION STUMP EN PYTHON

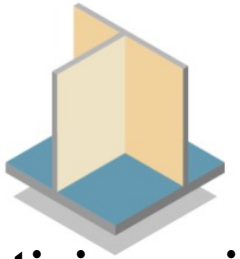
- DATASET: base de datos de 50000 clientes con 230 variables independientes.
- Calificar los clientes mas propensos a abandonar los servicios de una compañía (“**churn**”)

UNIVERSIDAD ICESI

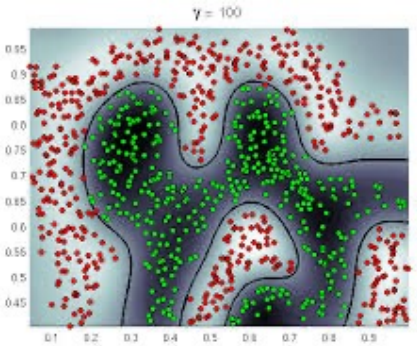
APRENDIZAJE SUPERVISADO

UNIVERSIDAD ICESI

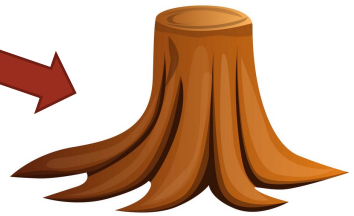
AGENDA



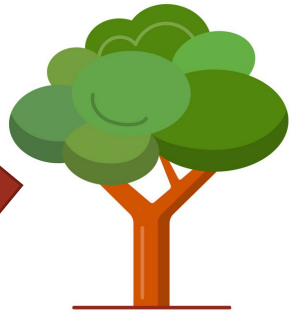
Particionamiento



Aprendizaje supervisado



Decision Stump



Árbol de decisión

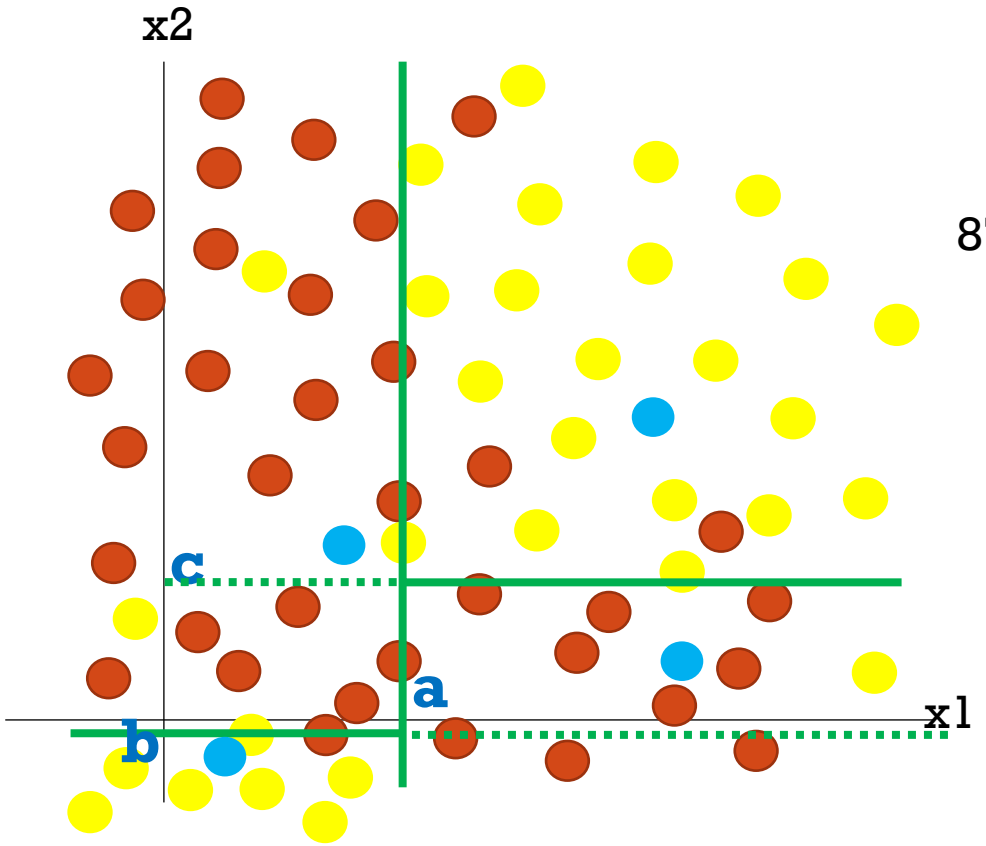
UNIVERSIDAD ICESI

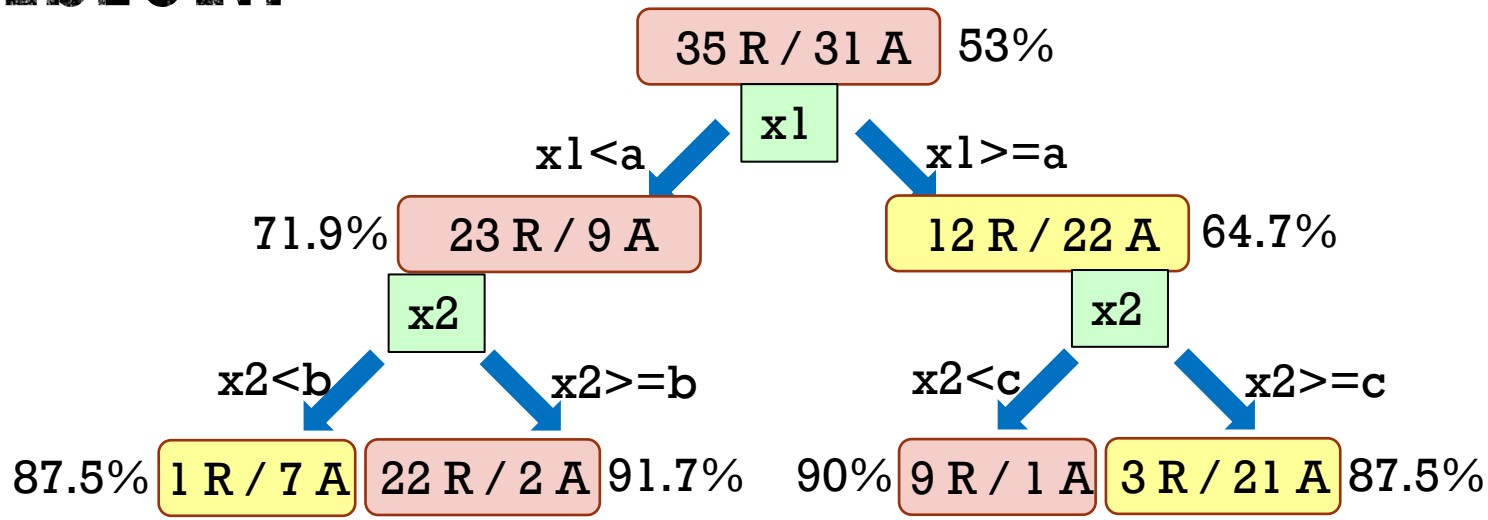
ÁRBOLES DE DECISIÓN: ALGORITMO

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

- **Seleccionar un factor** que separe **óptimamente** los valores objetivo del nodo actual, crear una rama por cada valor minimizando 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 existan más atributos para particionar
  - se llegue a un criterio de parada definido (pre-poda)

ÁRBOLES DE DECISIÓN: CLASIFICACIÓN



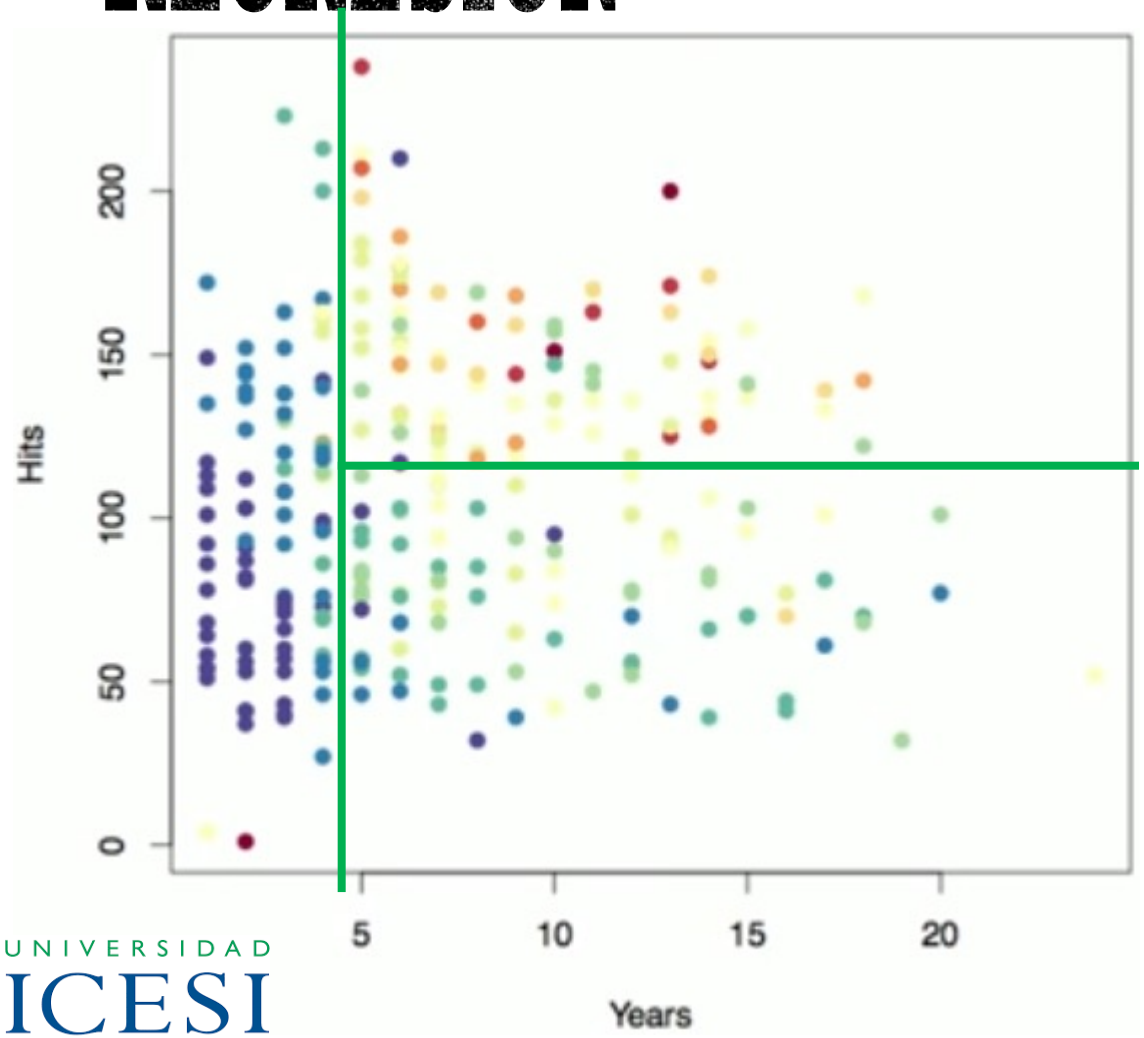


Paso	Accuracy
Raíz	35/66 = 53%
1era partición	45/66 = 68.2%
2a partición (rama izq.)	51/66 = 77.3%
3a partición (rama der.)	59/66 = 89.4%

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

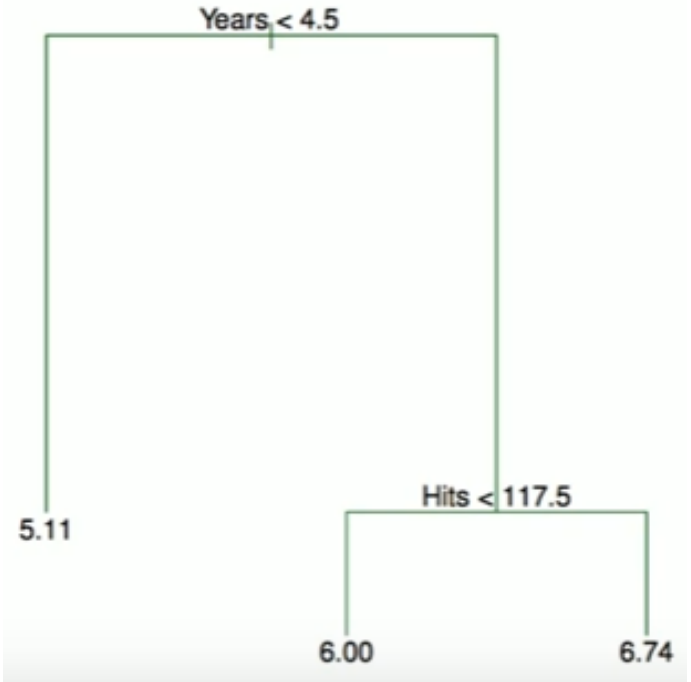
UNIVERSIDAD ICESI

ÁRBOLES DE DECISIÓN: REGRESIÓN



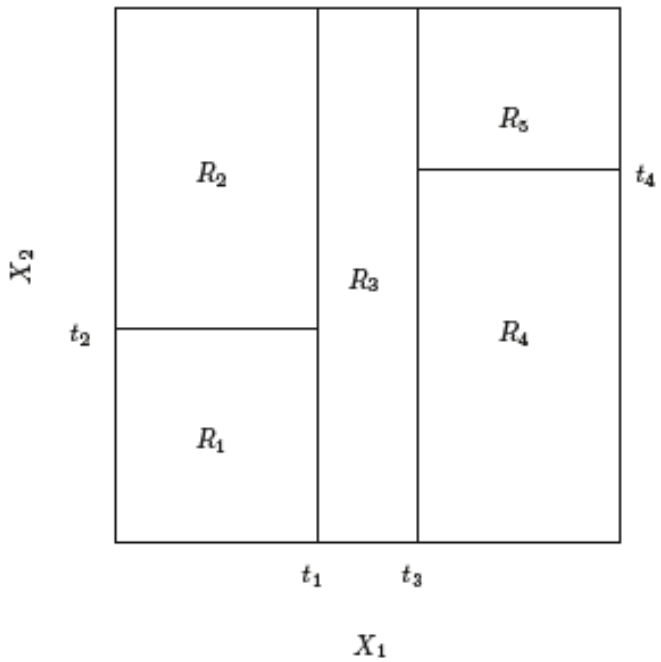
Evolución de lo salarios de beisbolistas (color) con respecto a años de experiencia (abscisa) y número de bateos exitosos (ordenada).

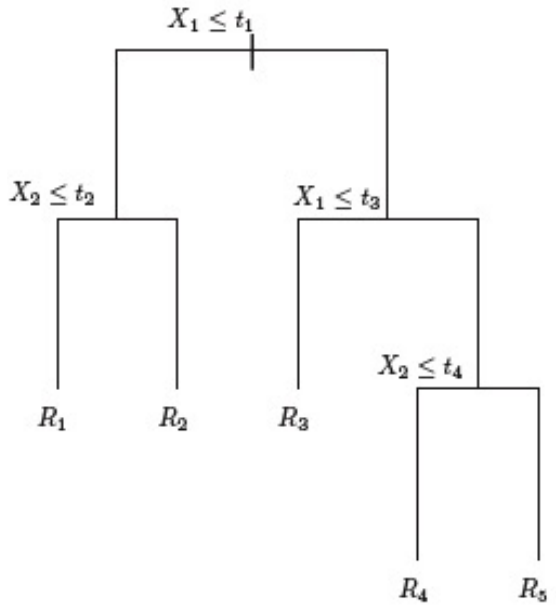
Se minimiza localmente $\sum_{j=1}^{J} \sum_{i \epsilon R_j} (y_i - \hat{y}_{R_j})^2$

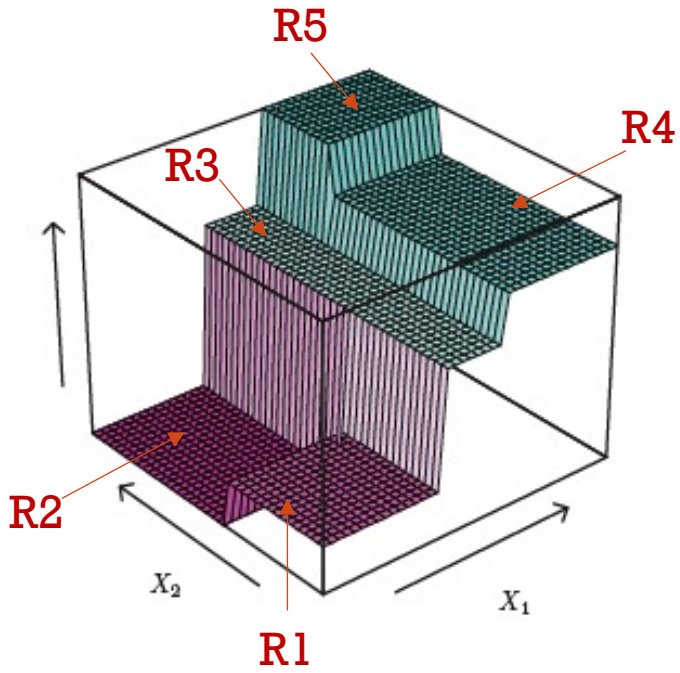


ISLR, 2013

ÁRBOLES DE DECISIÓN: REGRESIÓN







ISLR, 2013

UNIVERSIDAD ICESI

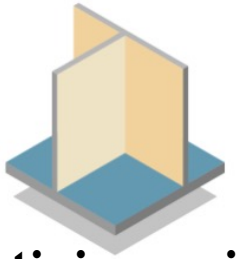
ÁRBOLES DE DECISIÓN

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

ÁRBOLES DE DECISIÓN

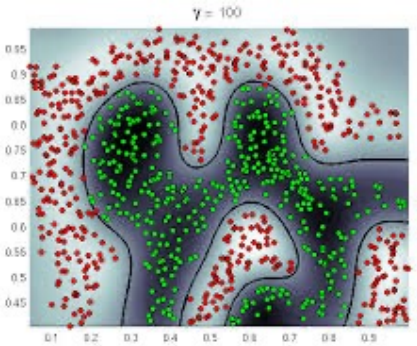
- Los datos deben ser **categóricos.** Las variables continuas deben ser **discretizadas**.
- Un árbol de decisión se puede representar como un conjunto de **reglas** booleanas
- Una nueva instancia se clasifica **siguiendo las ramas** del árbol
- Ideal para casos en los cuales un pequeño número de atributos provee gran cantidad de información
- Prueba diferentes atributos categóricos para aprender una clase, hace una selección automática de variables importantes.
- No se basa en ninguna noción de distancia, el modelo es **indiferente** a nociones de **normalización**

AGENDA

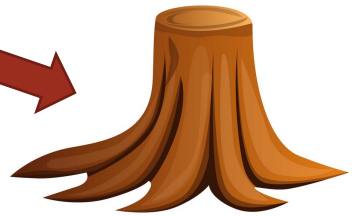


Particionamiento

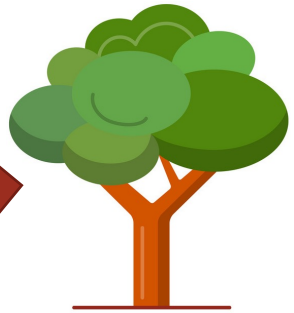
Criterios de particionamiento



Aprendizaje supervisado



Decision Stump



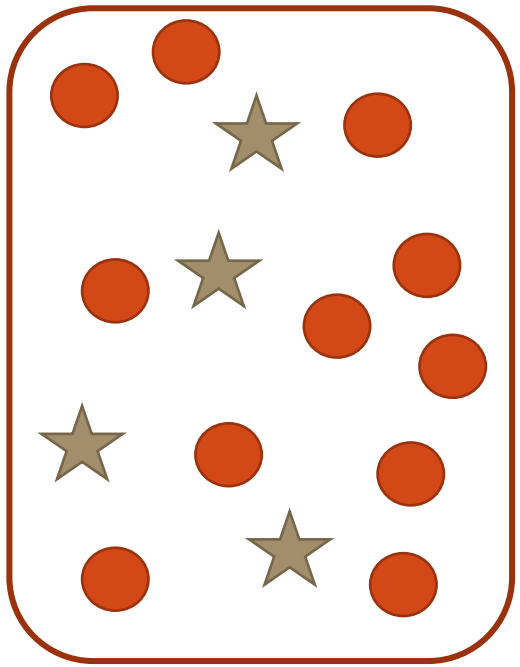
Árbol de decisión

UNIVERSIDAD ICESI

ÁRBOLES DE DECISIÓN

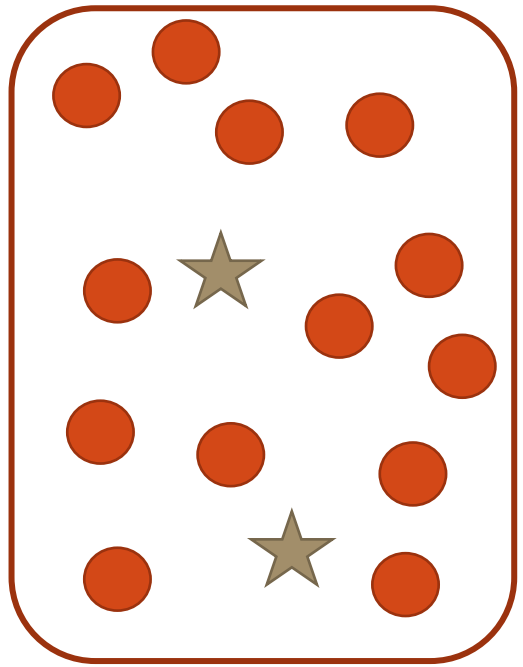
- **Existen diferentes criterios para determinar el mejor atributo de particionamiento en cada nodo** → Varios algoritmos:
  - **CART** (Classification And Regression Trees). Sólo particiones binarias, usando la métrica de impureza Gini para la clasificación y la reducción de varianza para la regresión
  - **ID3** (Iterative Dichotomizer). Basado en ganancia de información y entropía como criterio de división
  - **C4.5** (**C5.0** y **J48** en WEKA). Extensión de ID3, basado sobre razón de ganancia de información. 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



¿Cuál conjunto de datos presenta mayor desorden?

- →Entropia, como medida de desorden
- →Búsqueda de particiones cada vez mas puras
- →Reducción del desorden
- →Ganancia de información



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:

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

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

  - Ent.Cond.

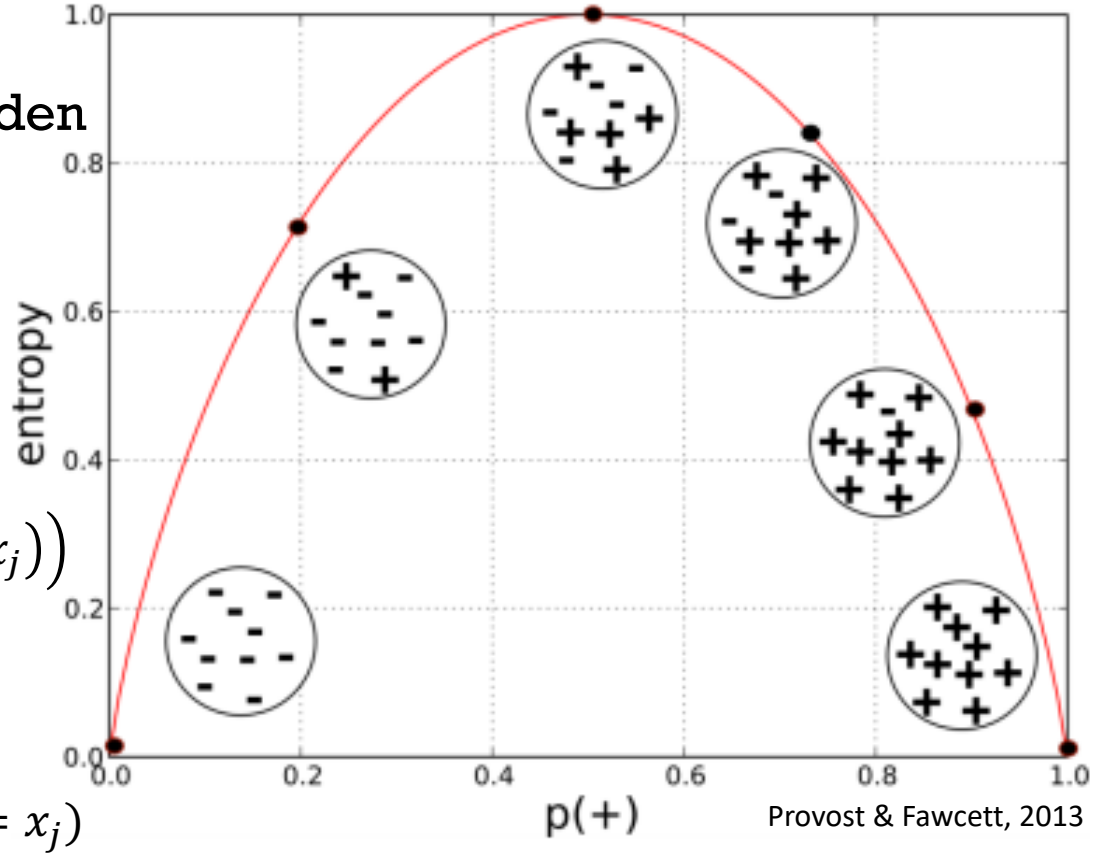
$$H\big(Y\big|X = x_j\big) = -\sum_i p\big(Y = y_i|X = x_j\big) * log_2\left(p\big(Y = y_i|X = x_j\big)\right)$$

  - 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

UNIVERSIDAD ICESI

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

¿Cuál es el mejor atributo para particionar el dataset?

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

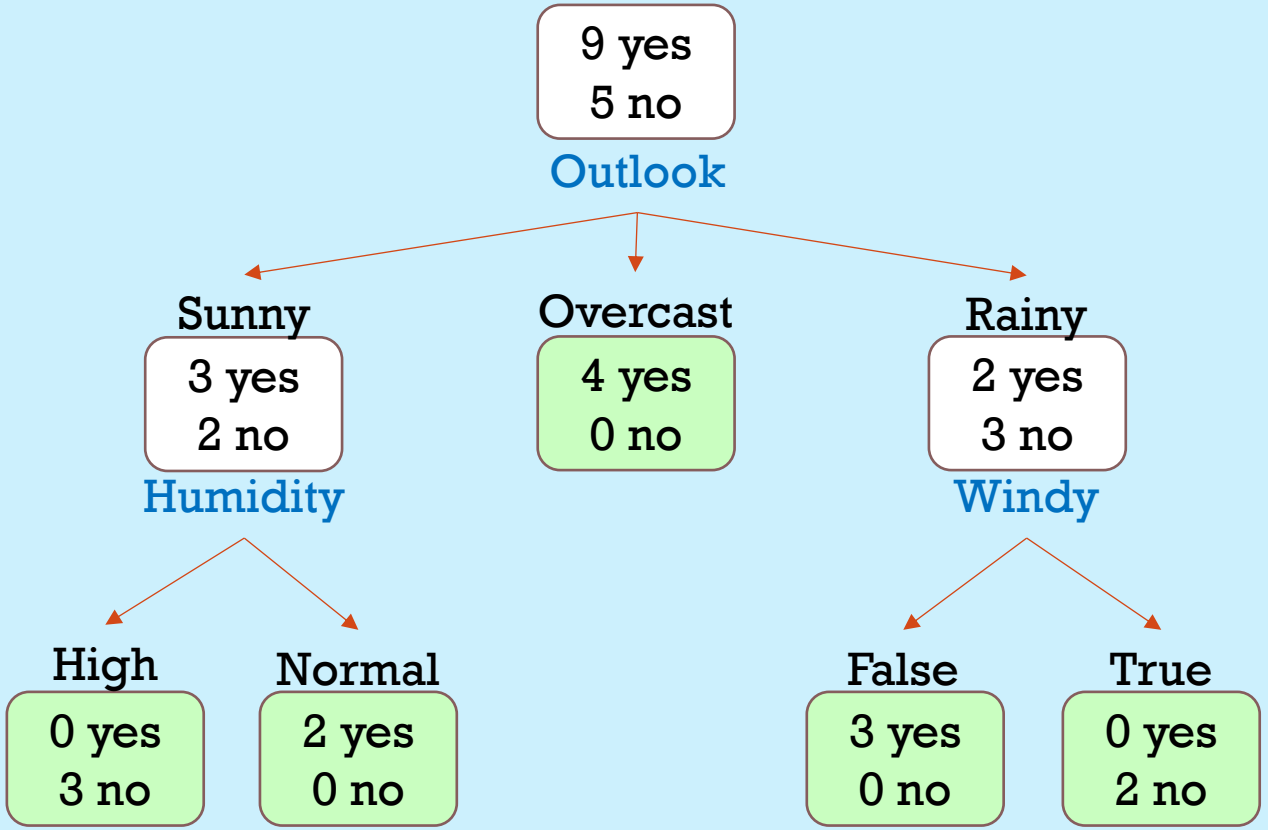
2. Calcular la entropía condicional para cada atributo y su ganancia de información

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		

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

UNIVERSIDAD ICESI

TALLER: ÁRBOLES DE DECISIÓN ID3



UNIVERSIDAD ICESI

ÁRBOLES DE DECISIÓN: CART (CLASIFICATION AND REGRESSION TREE)

- Solo árboles con particionamientos **binarios**
- Manejo de datos faltantes: Utilización de varias variables de particionamiento **sustitutas** (surrogate) para suplantar la variable cuyo valor es faltante
- **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

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

UNIVERSIDAD ICESI

ÁRBOLES DE DECISIÓN: CHAID (CHI-SQUARED AUTOMATIC INTERACTION DETECTION)

- 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

  Para cada atributo

  1. Calcular el valor esperado para cada combinación con la variable objetivo
  2. Calcular el Chi cuadrado para cada nodo hijo

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

  3. Escoger la variable con el mayor valor del chi cuadrado.

- Solo sirven para clasificación

TALLER: ÁRBOLES DE DECISIÓN CART + CHAID

Para esta clase (**ENTREGABLE**):

Identificar el atributo de particionamiento del dataset golf weather

- Para un árbol CART
  - Utilizando Gini
  - Considerar todas las particiones binarias posibles para las variables categóricas con mas de 2 valores. Por ejemplo: si $X \in \{A, B, C\}$, entonces se deben considerer 3 particionamientos: $\{A\}\ vs.\ \{B, C\}$, $\{B\}\ vs.\ \{A, C\}$, y $\{C\}\ vs.\ \{A, C\}$
- Para un árbol CHAID
  - Utilizando CHI2

	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

AYUDA: Sigue siendo el mismo (Outlook)

UNIVERSIDAD ICESI

PRIMER PARCIAL

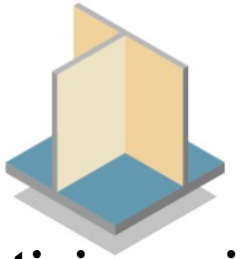
Revisión de las preguntas:

7 – 9, 12, 13

APRENDIZAJE SUPERVISADO

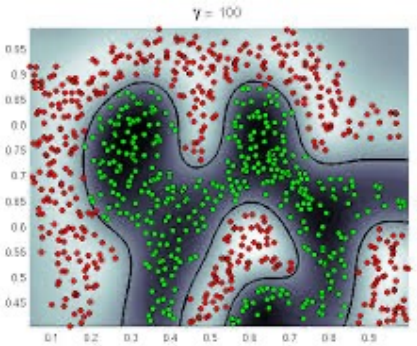
UNIVERSIDAD ICESI

Clase anterior

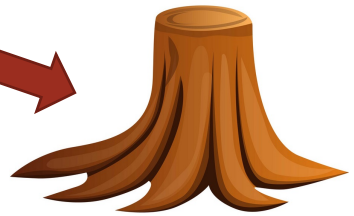


Particionamiento

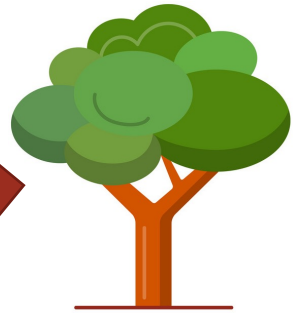
Criterios de particionamiento



Aprendizaje supervisado

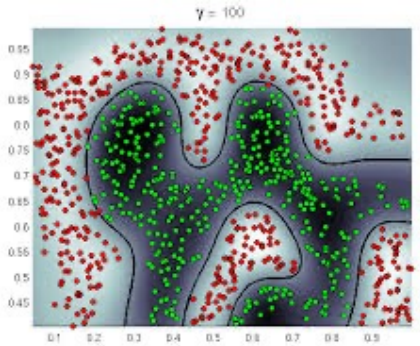


Decision Stump

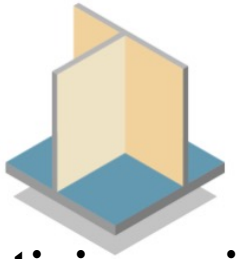


Árbol de decisión

UNIVERSIDAD ICESI



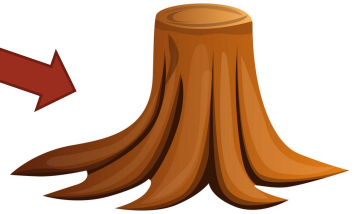
Aprendizaje supervisado



Particionamiento

Criterios de particionamiento

Modelos de ensamble



Decision Stump

Árbol de decisión

Poda

UNIVERSIDAD ICESI

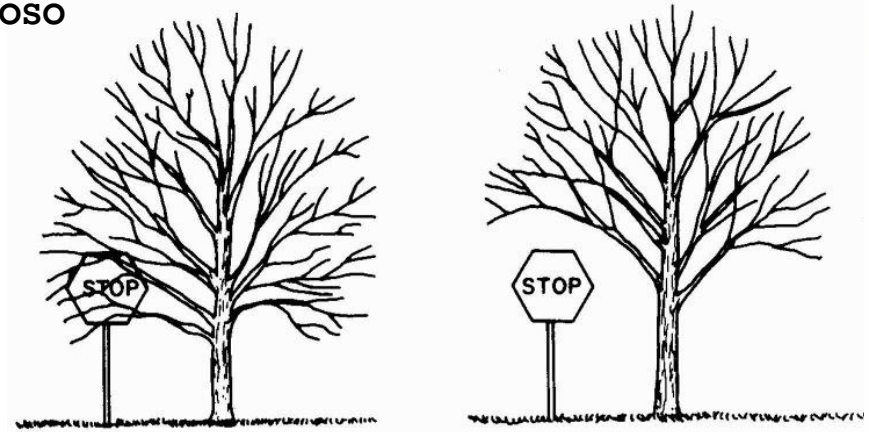
PODA

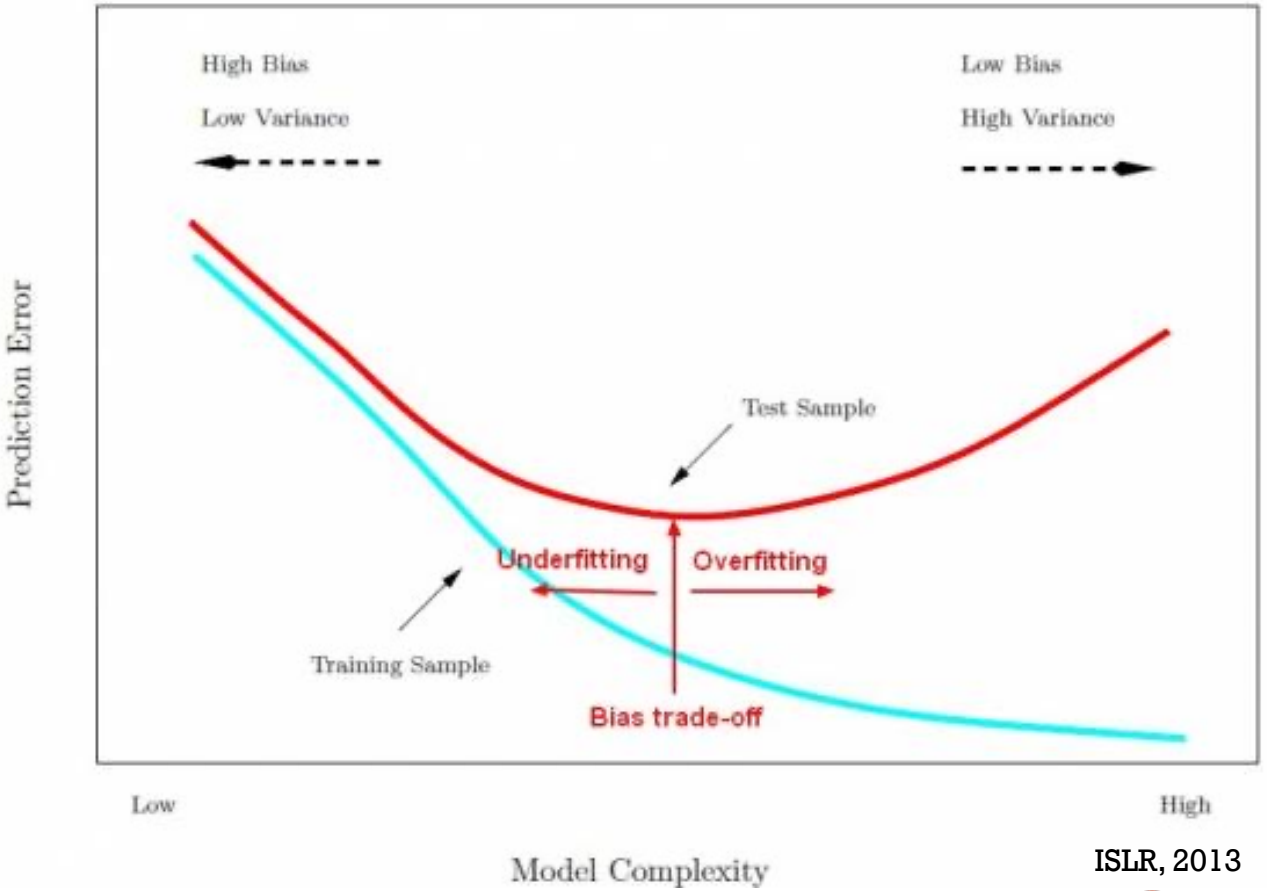
UNIVERSIDAD ICESI

ÁRBOLES DE DECISIÓN: PODA

Pre-Poda: Criterios de **parada temprana**

- Máxima **profundidad**: limita el crecimiento del árbol de manera global → usa CV
- Mínimo **número de instancias** para permitir particionamiento: limita localmente en cada nodo el crecimiento del árbol → usa CV
- No continuar si no se mejora suficiente el criterio de **particionamiento** o el **error** de entrenamiento → peligroso





ISLR, 2013

UNIVERSIDAD ICESI

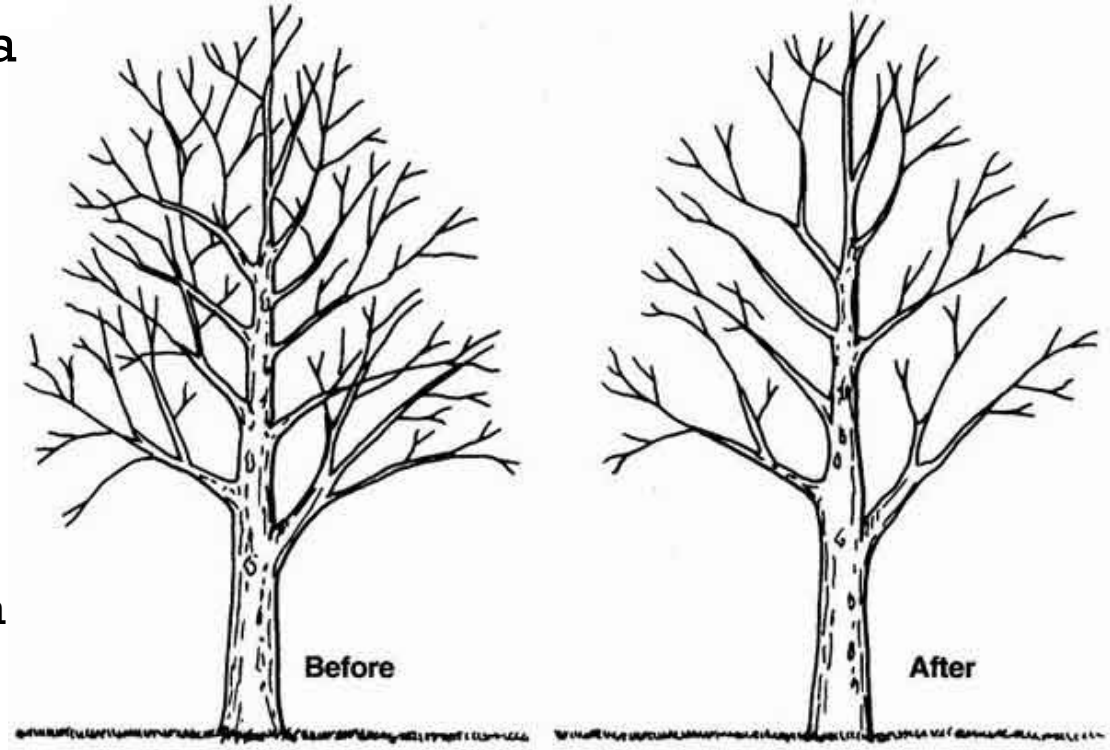
ÁRBOLES DE DECISIÓN: PODA

Post-poda: simplificar el árbol una vez se haya terminado de aprender:

- Preferible a la **pre-poda**
- Complejidad en términos de número de nodos hoja (terminales) L(T), no necesariamente de profundidad
- Podar bottom-up, teniendo en cuenta ahora también la complejidad del árbol:

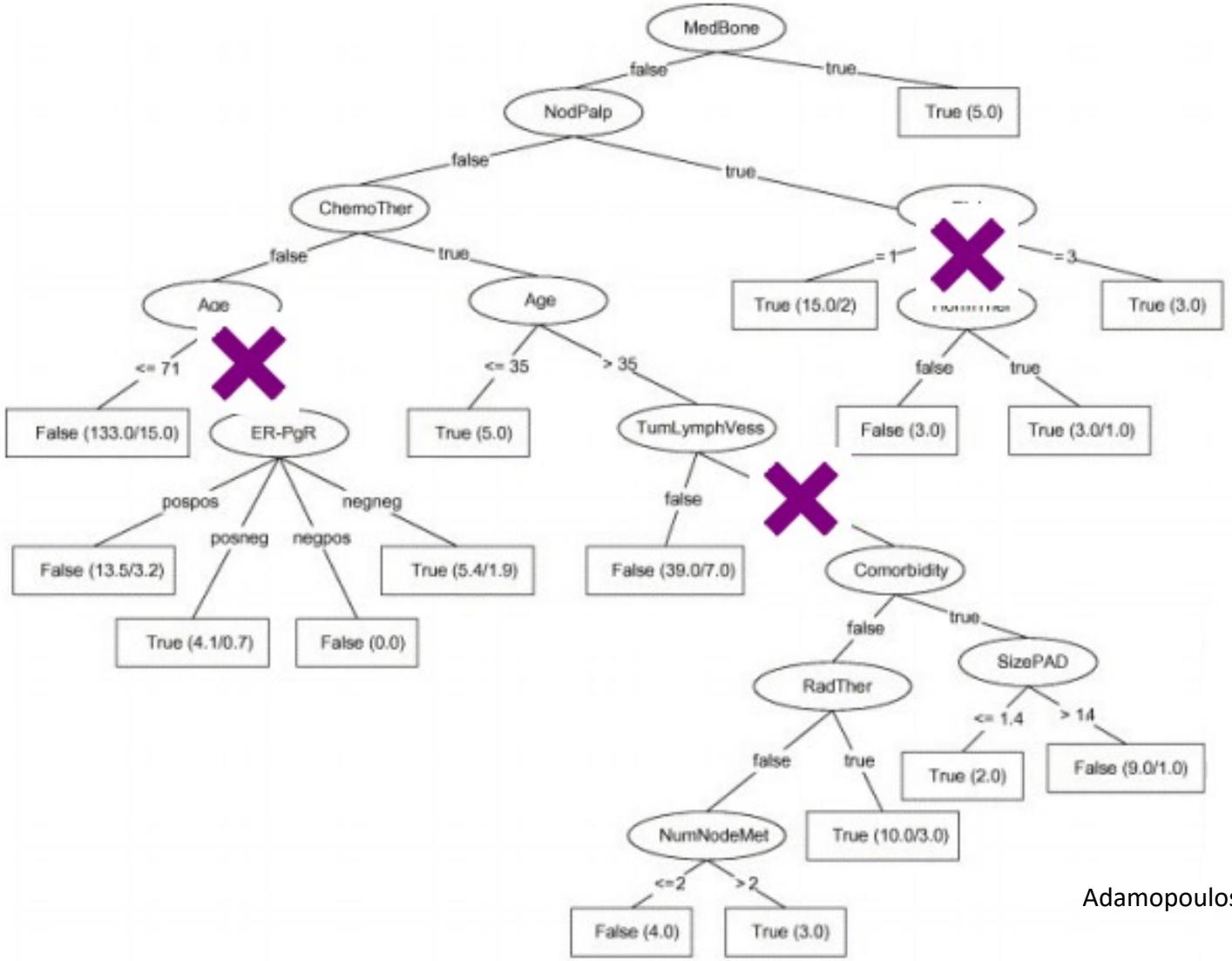
  Costo = Criterio(T)+ α*L(T), donde α controla la complejidad del modelo (α se puede estimar a través de CV).

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



UNIVERSIDAD ICESI

ÁRBOLES DE DECISIÓN: PODA

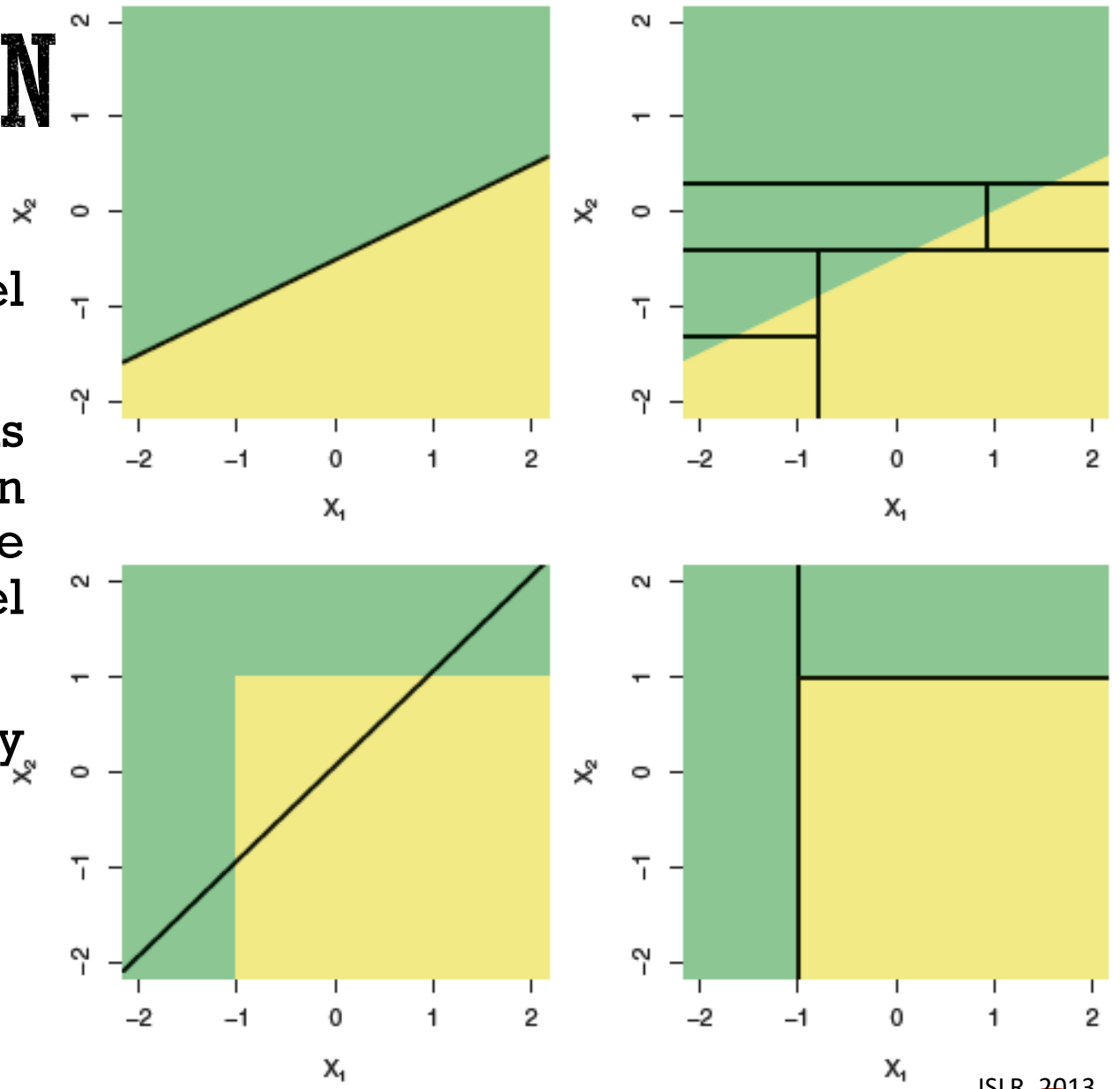


Adamopoulos, NYU, 2014

ÁRBOLES DE DECISIÓN

Consideraciones:

- Su rendimiento depende del comportamiento de los datos
- No son los algoritmos más competitivos, pero producen un modelo simple, altamente interpretable y similar el razonamiento humano
- Tienden al overfitting, por lo que hay que utilizar estrategias de poda



ISLR, 2013

UNIVERSIDAD ICESI

ÁRBOLES DE DECISIÓN

Consideraciones:

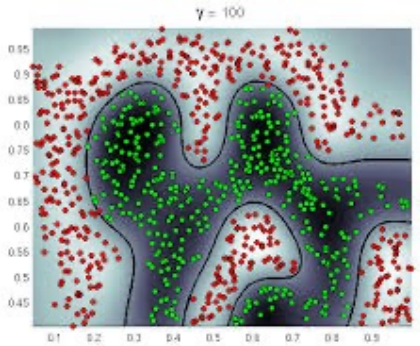
- Indiferentes a escalamiento, distribución de los datos y a datos faltantes
- Permiten la consideración de atributos cualitativos sin necesidad de crear variables adicionales (al menos en R, en Python obligan a variables numéricas)
- Crean un ranking de importancia de todas las variables predictivas
- Excluyen automáticamente variables no óptimas en el particionamiento
- Sensibles a bases de datos desbalanceadas y a pequeños cambios en los datos: sesgo hacia los atributos con mayor cantidad de valores
- Base de modelos de ensamble (Bagging, Boosting, Random Forest), que mejoran el poder predictivo, mientras reducen el error de varianza (overfitting)

TALLER: CLASIFICACIÓN CHURN

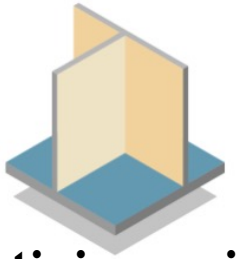
Vamos a continuar analizando el dataset de churn de clientes de la empresa de telefonía móvil. La idea es poder identificar los clientes propensos a abandonar la compañía utilizando árboles de decisión, interpretar sus resultados y compararlo con los encontrados previamente.

- Descargar el dataset 01_churn.csv y el cuaderno 08_Arboles_Churn-STUD.html correspondiente
- Desarrollar el cuaderno hasta llegar a la sección de ensambles

UNIVERSIDAD ICESI



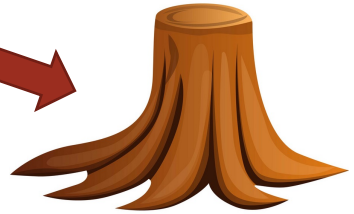
Aprendizaje supervisado



Particionamiento

Criterios de particionamiento

Modelos de ensamble



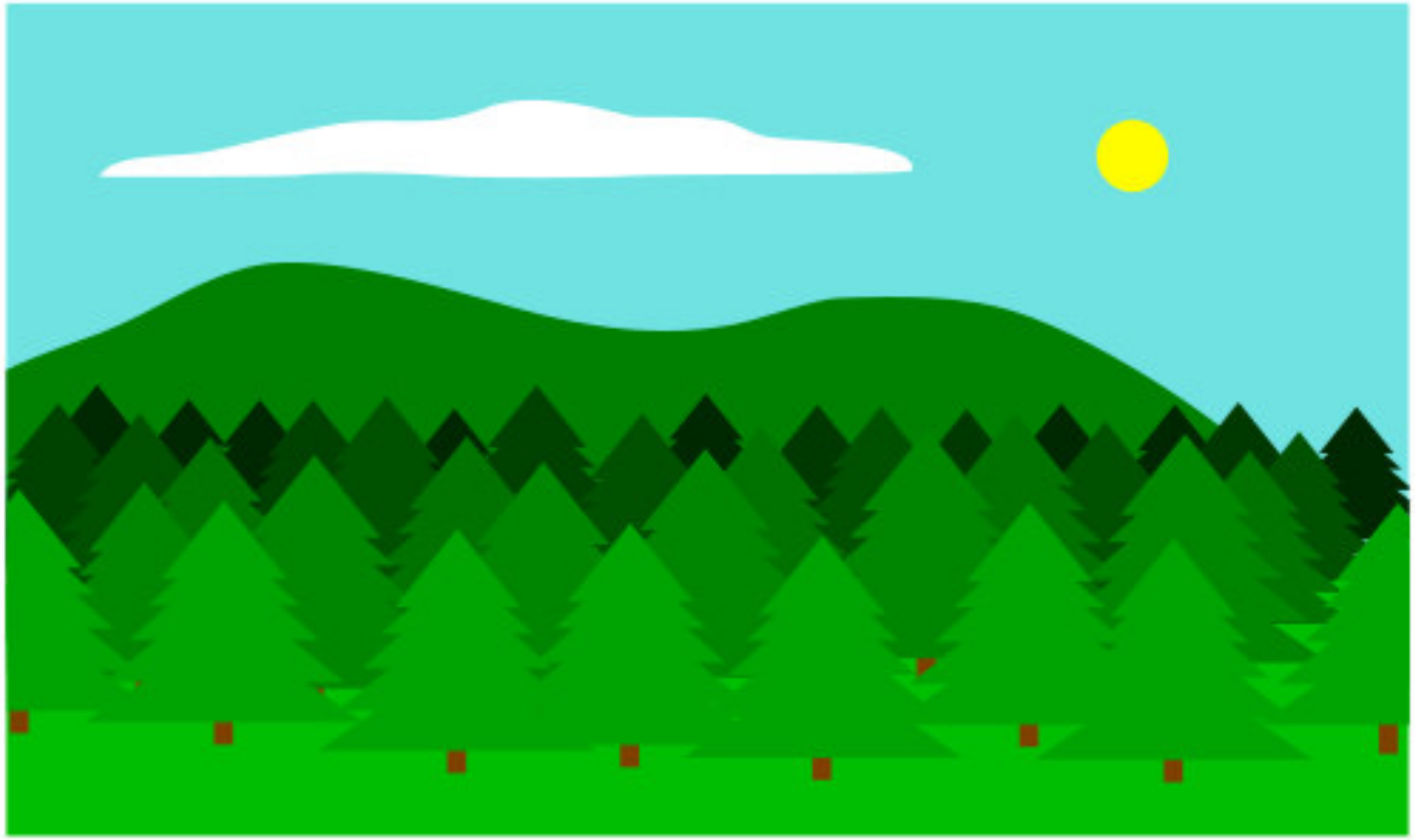
Decision Stump

Árbol de decisión

Poda

UNIVERSIDAD ICESI

ENSEMBLE LEARNING



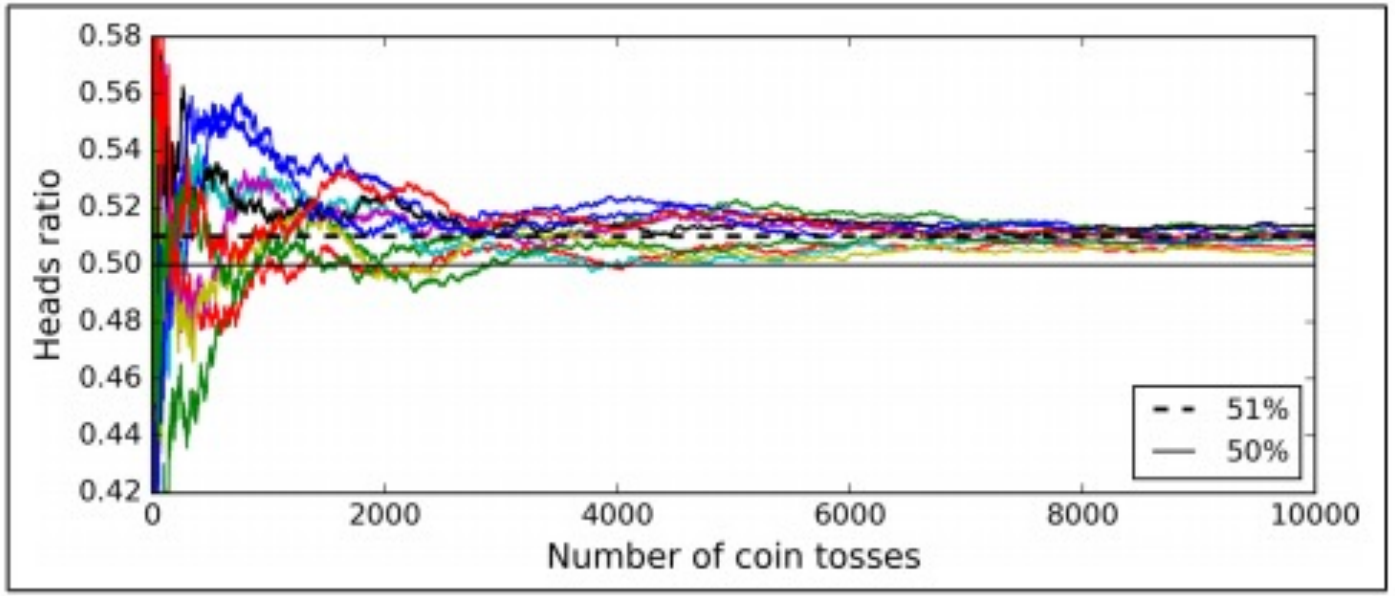
UNIVERSIDAD ICESI

ENSEMBLE LEARNING

Cómo verificar que una moneda no esté trucada?

→ Lanzar la moneda n veces

→ A mayor número de lanzamientos, mejor la estimación



Géron, 2017

UNIVERSIDAD ICESI

ENSEMBLE LEARNING

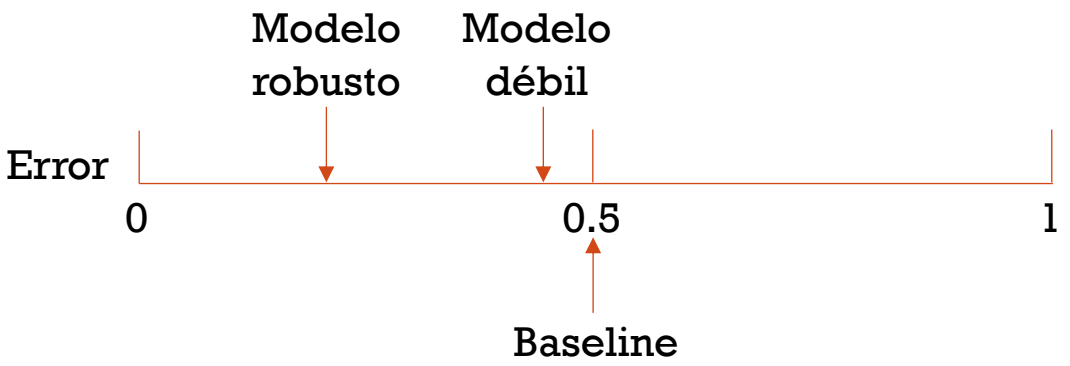
- Diversificación:
  - Cada modelo tiene un riesgo de mala predicción
  - ¿Por qué limitar la decisión a un solo modelo si podemos construir y utilizar varios?

    → Combinar un grupo de clasificadores/regresores

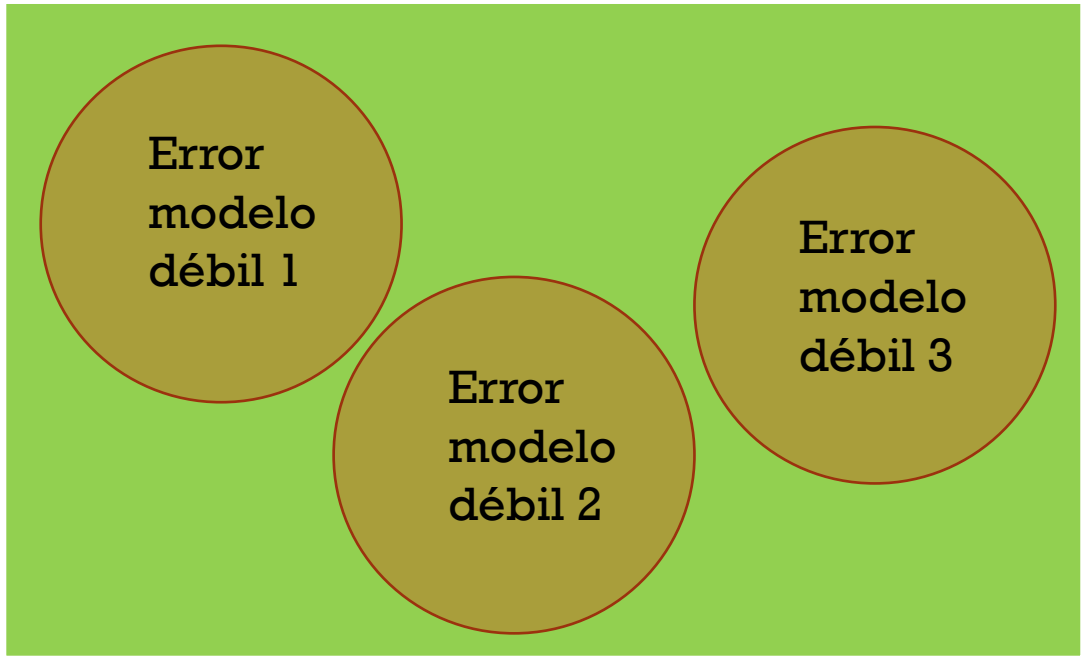
    → Decisión basada en la agregación de varios modelos (max / promedio)
- Reducir la varianza global al agregar un conjunto de modelos supervisados, y obtener mejor generalización
- Se aumenta la calidad de la predicción pero se pierde en interpretabilidad

ENSEMBLE LEARNING

- Modelos débiles vs. robustos:
  - Los modelos robustos tenderán a estar de acuerdo en la mayoría de los casos
  - A menudo los datos permiten construir modelos que son apenas mejores al baseline



BAGGING (BOOTSTRAP)

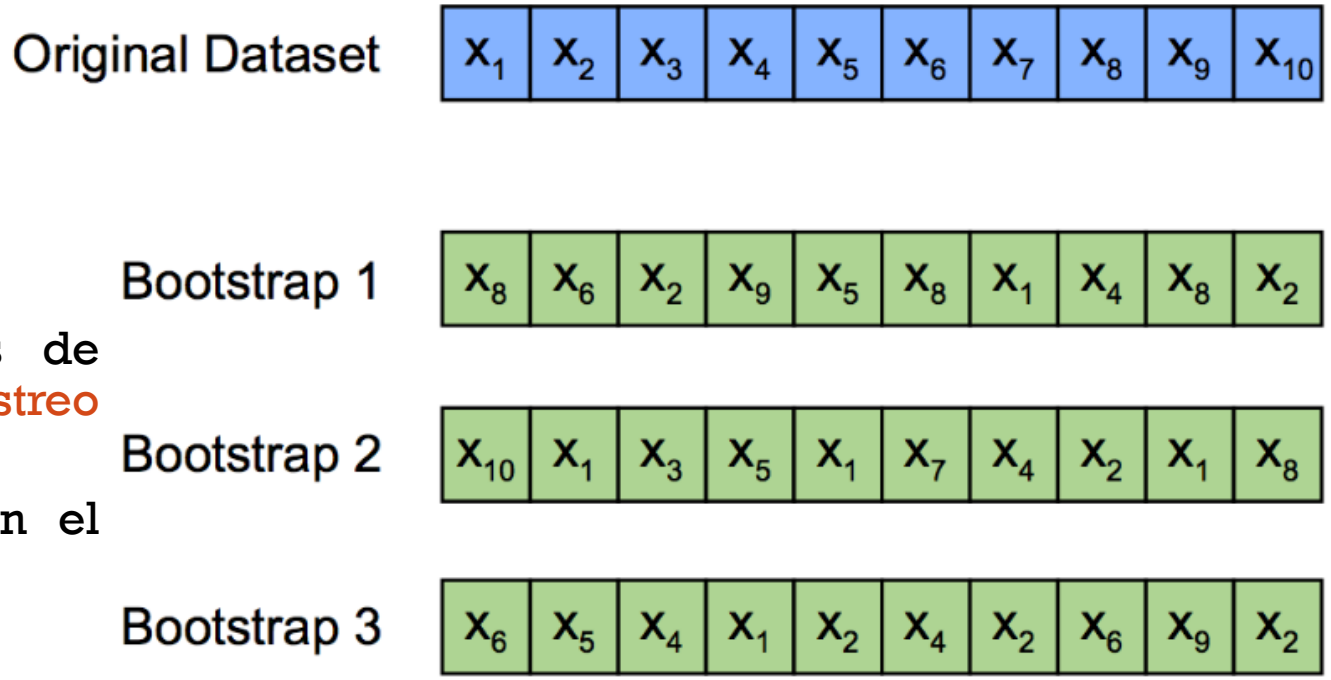


- ¿Cómo es nuestro modelo si ponemos a votar a los 3 modelos?

→ Entre más de-correlacionados estén los modelos individuales, mejor va a ser el meta-modelo agregado

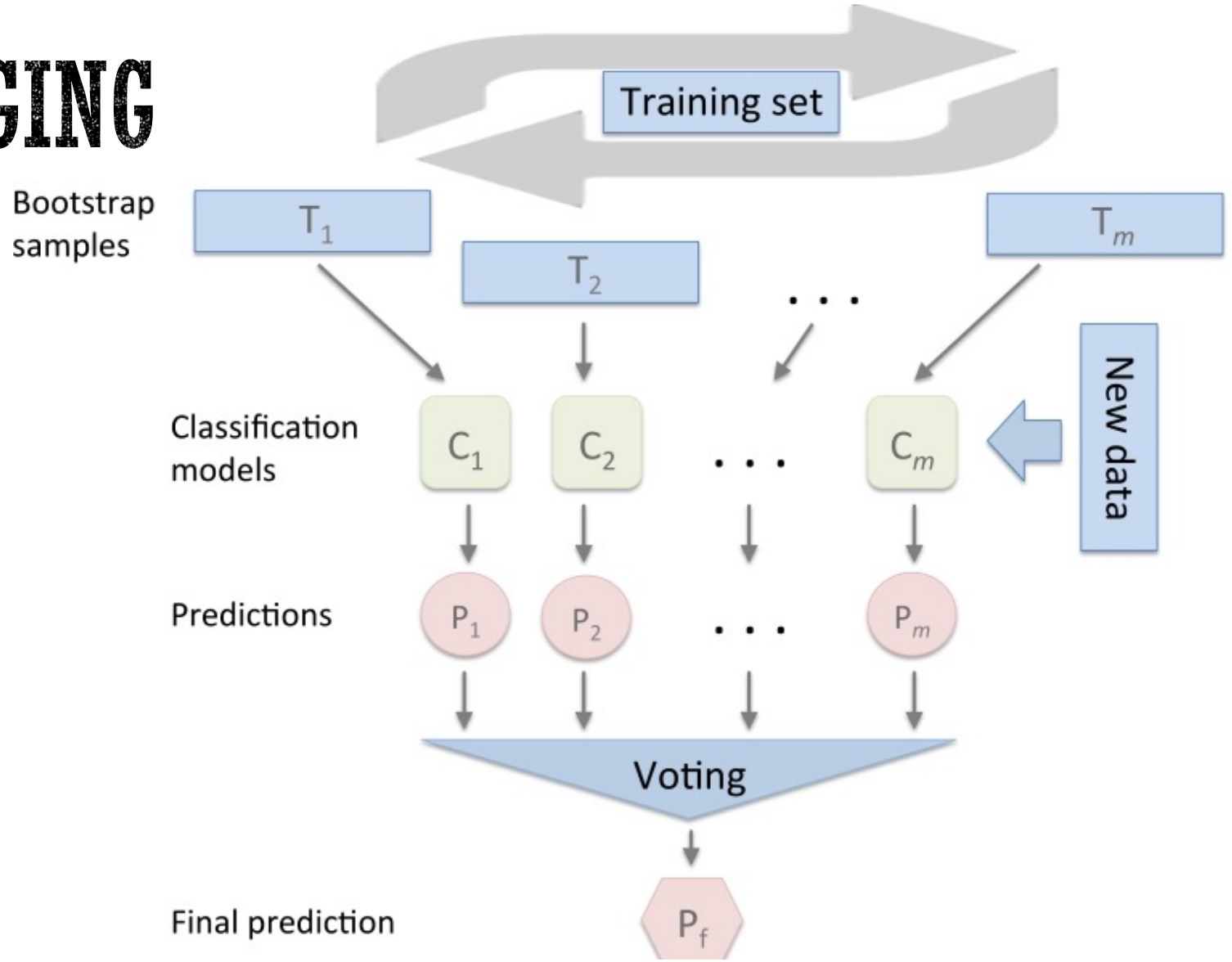
BAGGING

- Bagging = Bootstrap aggregating
- Bootstrap: técnica de muestreo
  - Consideración de varios conjuntos de entrenamiento/test utilizando muestreo con reemplazo
  - Por lo general los muestreos tienen el mismo tamaño del conjunto original
  - Reduce la varianza



Sebastian Raschka, 2015

BAGGING



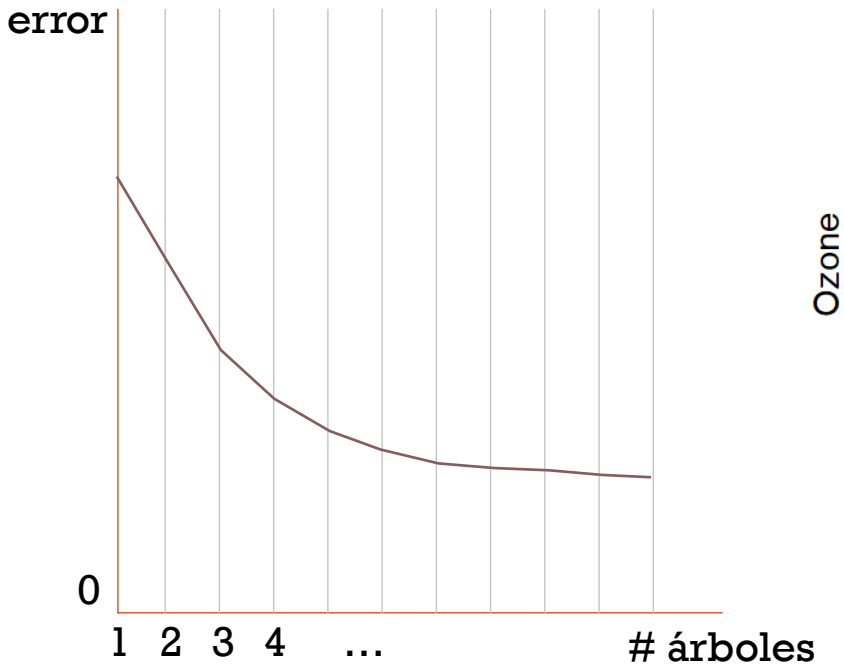
Raschka, 2017

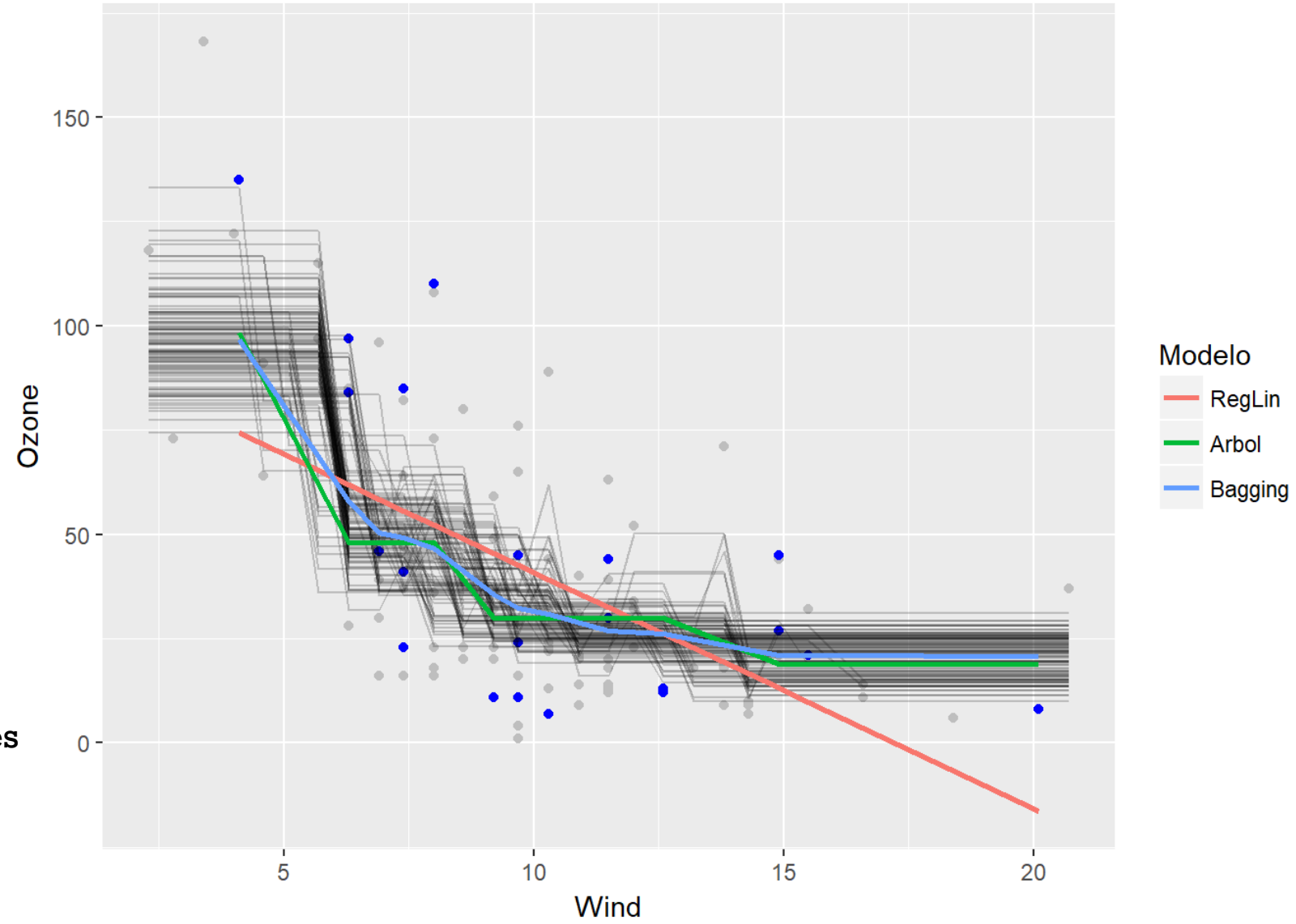
UNIVERSIDAD ICESI

BAGGING

- Algoritmo (Breiman, 1994-1996):
  1. Escoger múltiples subconjuntos de instancias con repetición
  2. Entrenar varias instancias de un mismo tipo de modelo "**robusto**" de aprendizaje (bajo sesgo – alta varianza) basado en cada subconjunto
  3. Agregar las decisiones de cada modelo en una sola decisión global (votación, promedio)
- Muy útil cuando se trata de modelos no lineales (e.g. árboles de decisión)
- Resultado: sesgo similar, reducción de varianza (estabilización de las predicciones) → mejora el accuracy
- Disminuye la interpretabilidad

BAGGING



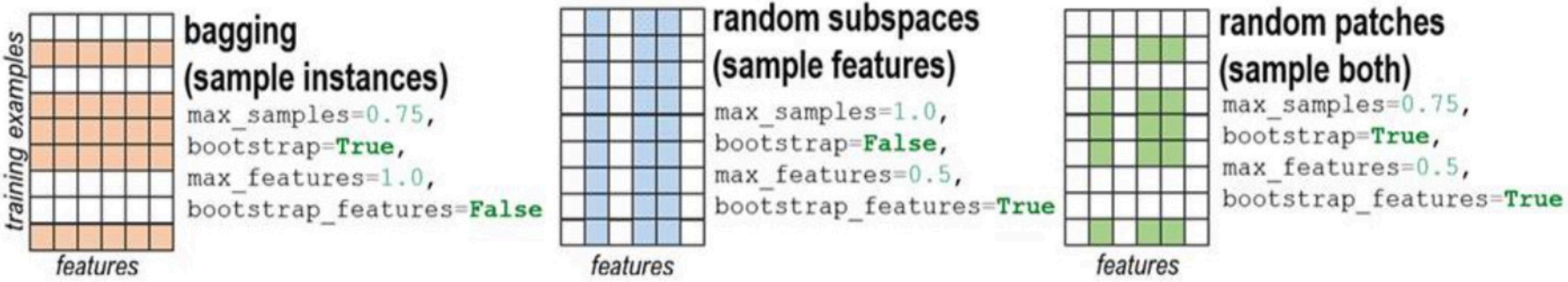


UNIVERSIDAD ICESI

BAGGING

- Se aplica tanto a clasificadores como a regresores

Figure 2.9 Bagging compared to random subspaces and random patches.



- **Random patches**: subconjuntos tanto de los registros como de las variables independientes

Random patching

UNIVERSIDAD ICESI

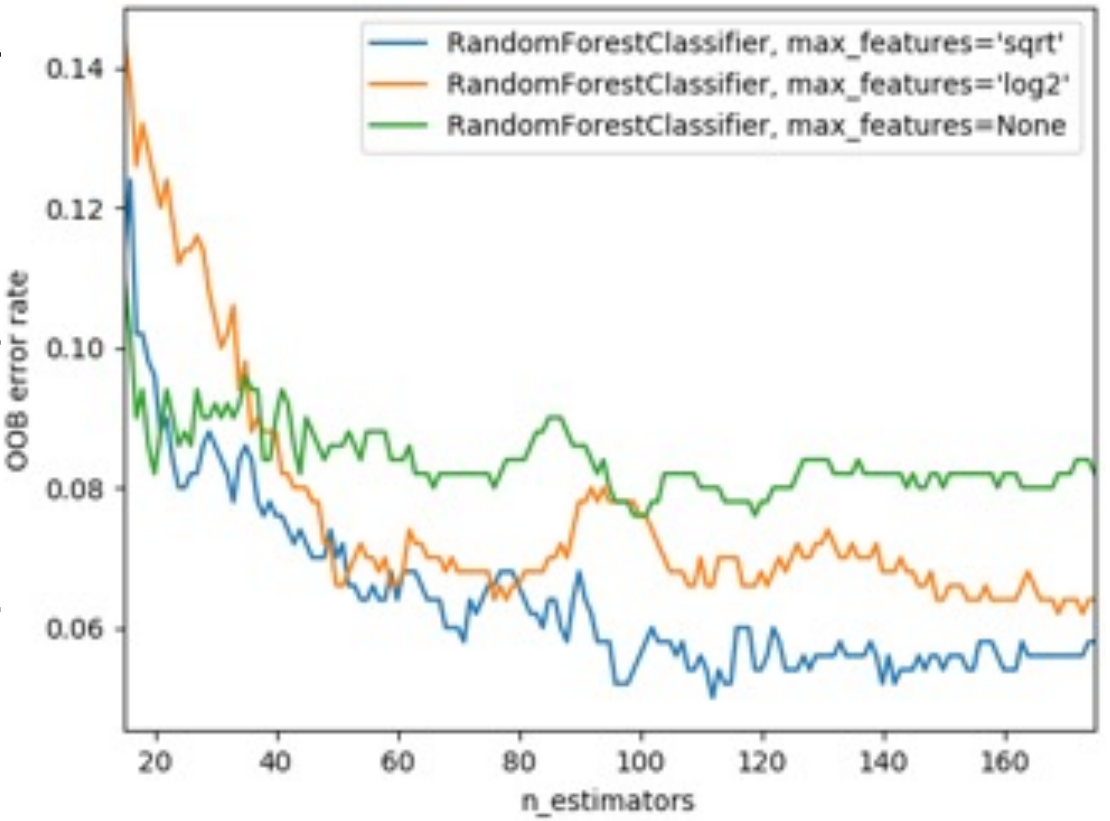
BAGGING

OOB (Out of bag error estimation):

- Un subconjunto bootstrap contiene, en promedio, aproximadamente 2/3 de los datos del conjunto original
- OOB usa el 1/3 restante de datos para calcular métricas de evaluación de los modelos individuales, que luego deben agregarse
- Con una cantidad elevada de árboles, tiende hacia el error LOOCV :
  - Evita tener que dejar un test set independiente
  - Evita el costoso proceso de K-Fold CV

RANDOM FOREST

- Junto a boosting, uno de los algoritmos con mejor performance en casos de datos **estructurados**
- Extensión de bagging basado en árboles CART
- Cada árbol es **independiente** y tiene la **misma importancia** en la decisión final
- La idea es minimizar la **correlación** entre los modelos, forzando la diferencia más alla del bootstrap de los datos de entrenamiento
- Limita el **número de atributos** a considerar en cada nodo de particionamiento.
- Valores por defecto:
  - En caso de regresión: m/3
  - En caso de clasificación: $\sqrt{m}$

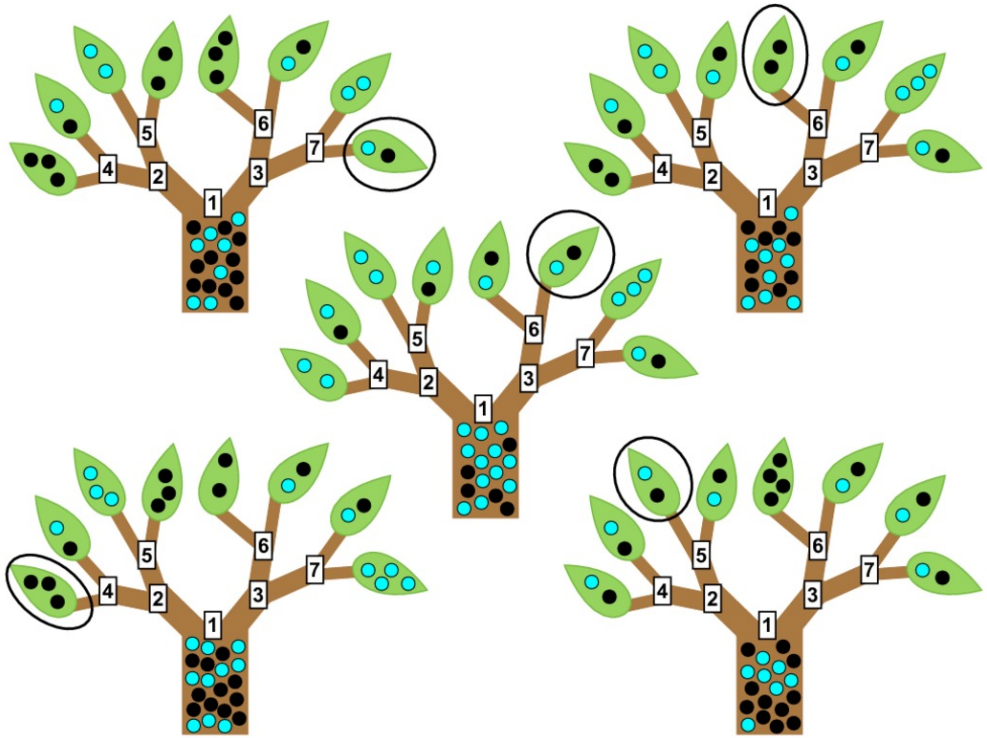


Scikit-learn.org

UNIVERSIDAD ICESI

RANDOM FOREST

- Algoritmo
  1. Crear múltiples árboles de decisión, construidos sobre muestras con reemplazo (bootstrapping) del dataset
  2. En cada punto de partición, considerar solo una muestra aleatoria de todas las variables disponibles
  3. Crecer el árbol
  4. Agregar los resultados de todos los árboles creados



http://inspirehep.net/record/1335130/plots

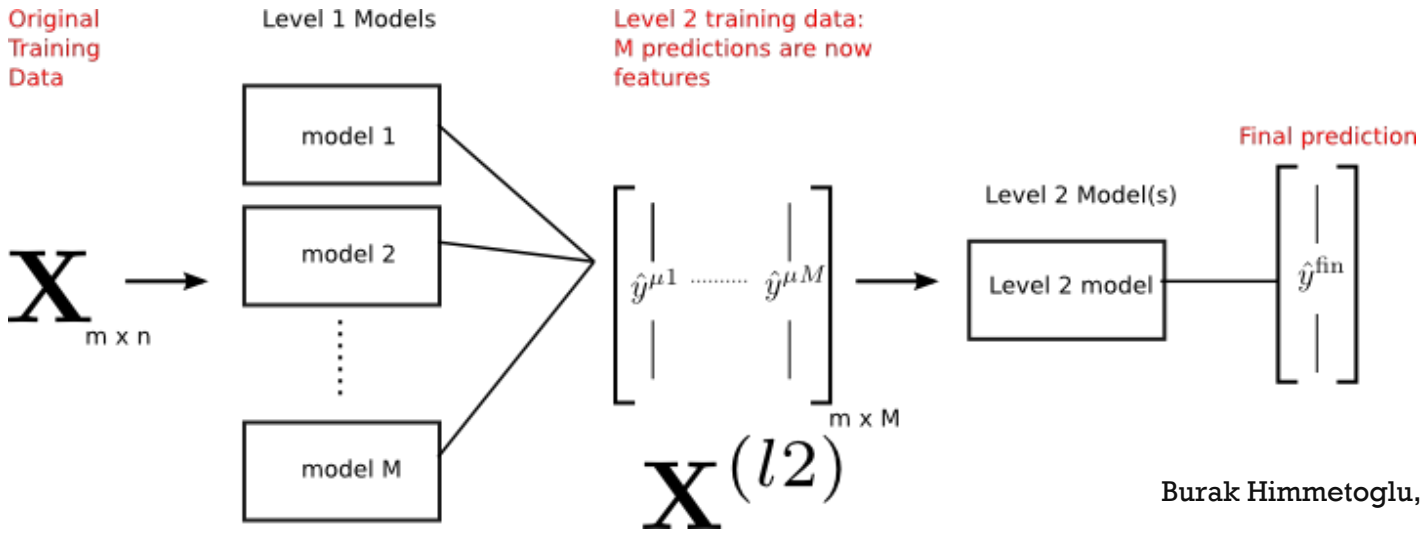
UNIVERSIDAD ICESI

RANDOM FOREST

- Consideraciones
  - Al igual que el bagging:
    - **Heredan** cosas buenas de los árboles de decisión (importancia de las variables), mejoran las malas (varianza)
    - **Lento** (gran número de árboles a entrenar) y toman mucho espacio en memoria, pero **escalable** (independencia)
    - No genera **overfitting** con más árboles, pero los árboles individuales pueden tenerlo (limitar su profundidad)
    - Muy difíciles de interpretar
  - Requieren de un buen afinamiento de parámetros (bagging es mas simple)

STACKING

- 2 capas de modelos:
  - Una primera capa incluye modelos diferentes, no necesariamente débiles, no necesariamente de la misma familia, que se entrenan con el dataset original
  - Una segunda capa con un solo modelo, típicamente una regresión logística (en clasificación) o lineal (en regresión), que se entrena con las salidas de los modelos de la primera capa como entradas
  - Los modelos de las 2 capas comparten la misma variable objetivo

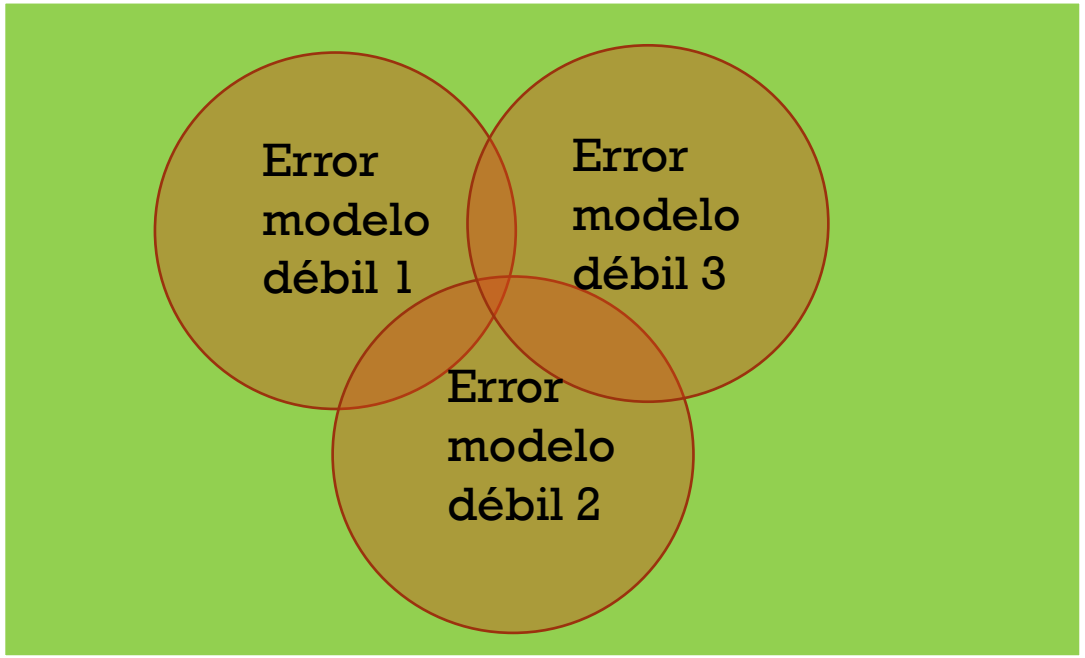


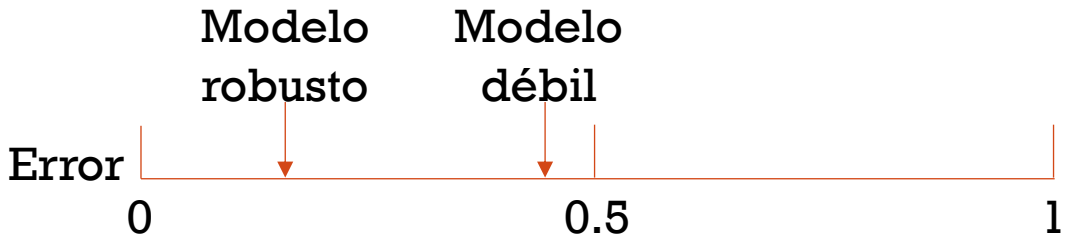
Burak Himmetoglu, UC Santa Barbara.

UNIVERSIDAD ICESI

BOOSTING

- Modelos débiles vs. robustos





- ¿Cómo es nuestro modelo si ponemos a votar a los 3 modelos?
- ¿Hay ciertas regiones mas importantes en cuanto al error que otras?
- ¿Cómo mejoro en las regiones más susceptibles al error donde ningún modelo logra buenos resultados?

UNIVERSIDAD ICESI

BOOSTING

- Agregación de predictores **débiles**, aprendidos **secuencialmente**
- Aprendizaje lento progresivo: reduce el error de sesgo al considerar los errores previos en los modelos subsiguientes.
- La **importancia de cada registro** se modifica según los resultados del último modelo entrenado → Cada modelo es **dependiente** de los modelos anteriores
- Cada modelo tiene **diferente influencia** en la decisión final; este depende de su poder predictivo
- La idea es reducir el error de sesgo al considerar los errores previos en los modelos subsiguientes

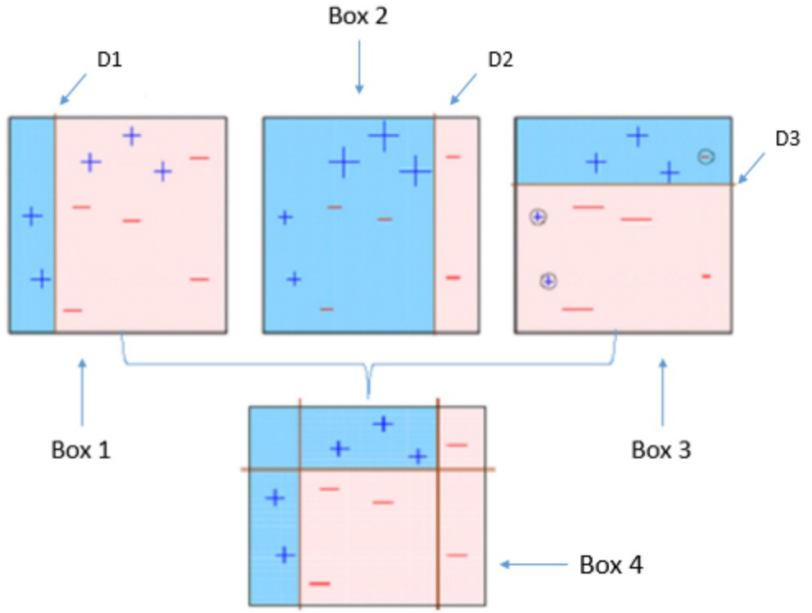
- Puede caer en **overfitting** si el número de árboles (modelos) es grande

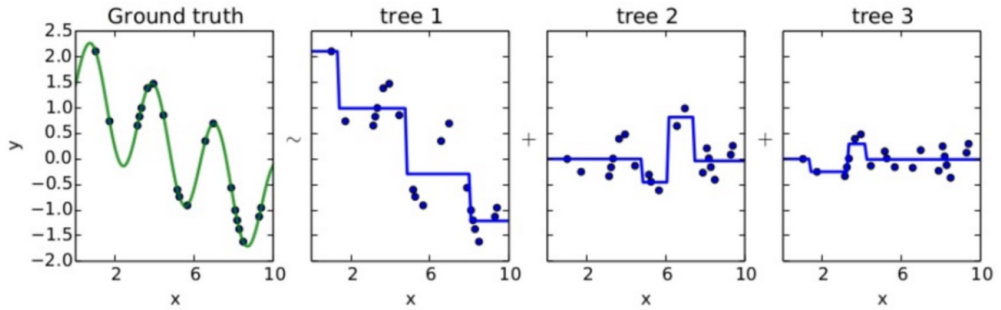
BOOSTING

- Algoritmo
  1. Crear un modelo predictor **débil** inicial sobre el dataset
  2. Asignar un **peso** a la decisión del modelo dado su **error** de entrenamiento
  3. Actualización de los pesos de cada instancia del dataset: se aumenta en las instancias que fueron mal predichas, se reduce en las instancias correctas
  4. Crear un nuevo predictor, influenciado por los errores cometidos por los modelos anteriores
  5. Agregar una decisión ponderada de todos los predictores creados
- Tres parámetros importantes:
  - Número de árboles a desarrollar
  - Learning rate o shrinking parameter: controla la velocidad del aprendizaje, para limitar el overfitting (valores típicos 0.01 - 0.001)
  - Número de particiones. Controla la complejidad de los árboles. Se conoce como interacción de profundidad máxima (1 funciona bien)

VARIANTES DE BOOSTING

- ADA BOOST
- GRADIENT BOOSTING
- EXTREME GRADIENT BOOSTING (XGBOOST)
  - Hands-On Machine Learning with Scikit-Learn and TensorFlow. Aurélien Géron. 2017. O'Reilly Medi



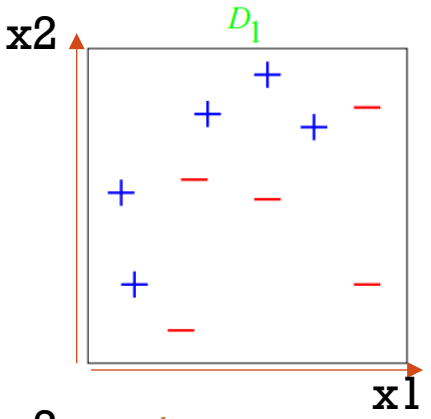


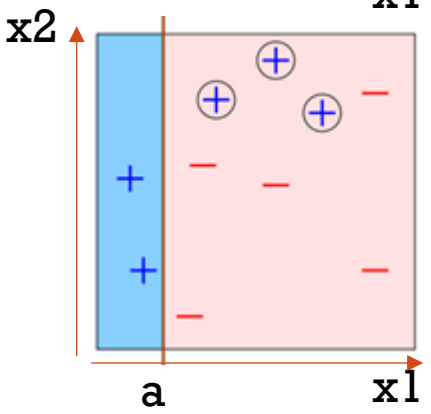
Source: Quora

UNIVERSIDAD ICESI

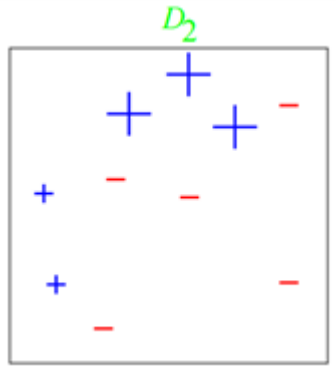
ADA BOOST

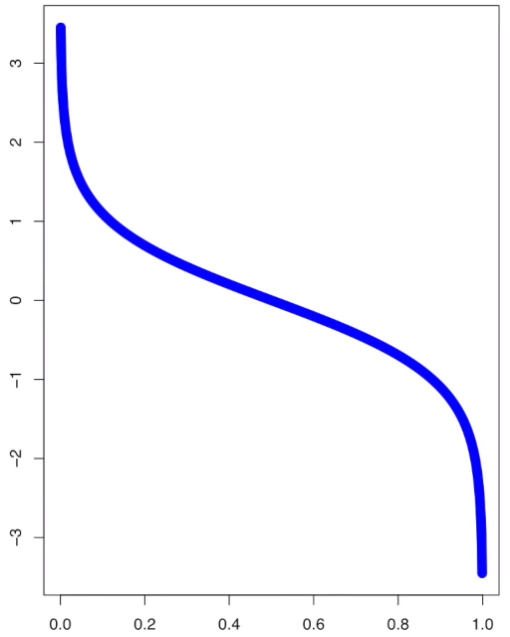
- Combinación de varios **decision stumps**, con **importancias diferentes** en la agregación final
- Inicialmente toda las instancias tienen el mismo peso: 1/n
- Se encuentra el mejor decisión stump,
- Se evalúan los errores y se calcula la importancia del stump, que se utilizará en el lugar del **learning rate**: $\alpha = \frac{1}{2} ln \frac{1-error}{error}$
- Actualizar los pesos de las instancias
  - Incorrectas: $peso_t = peso_{t-1} * e^{\alpha}$
  - Correctas: $peso_t = peso_{t-1} * e^{-\alpha}$



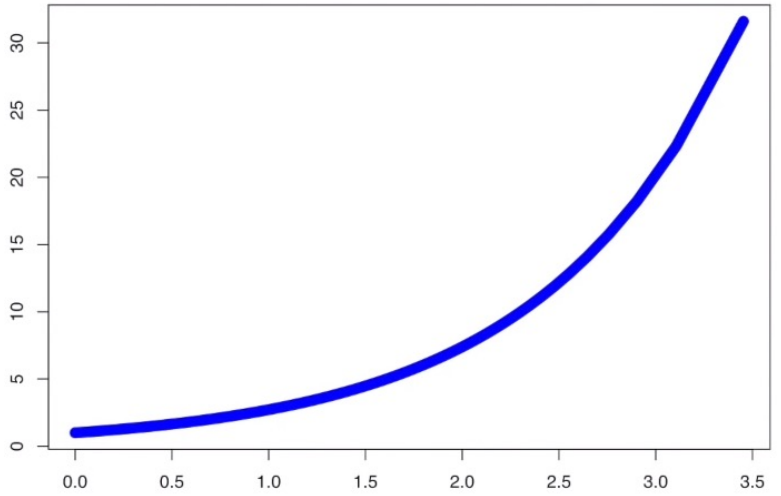


Árbol	Error	α
x1<a	3/10	0.424



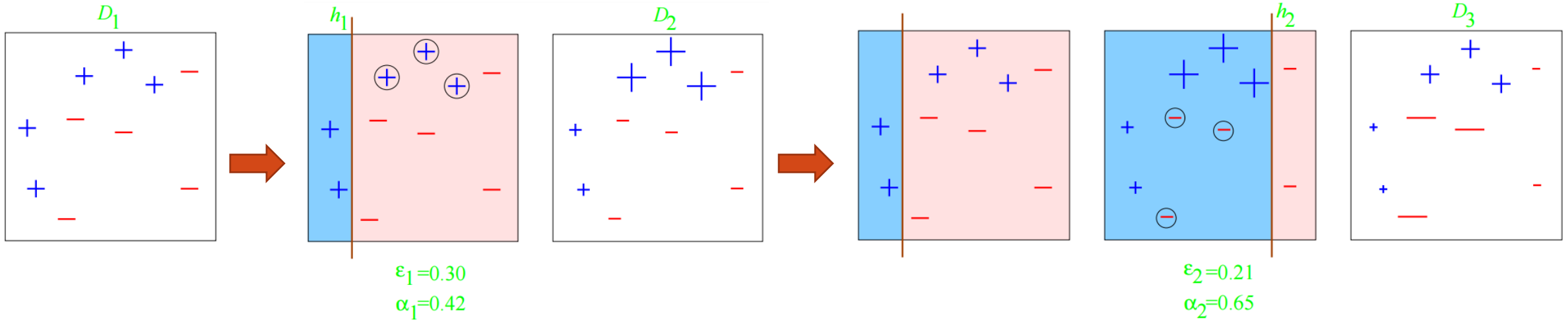


α según el error

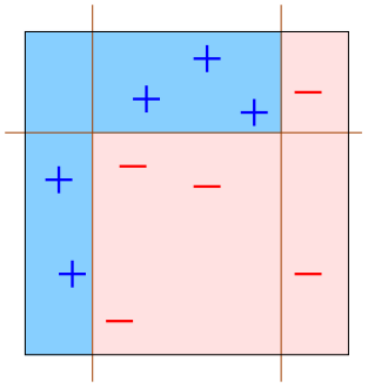


Nuevo peso dado α

ADA BOOST



$$H_{\text{final}} = \text{sign}\left(0.42 + 0.65 + 0.92\right)$$



TALLER: BAGGING, RANDOM FOREST, BOOSTING

- Vamos a analizar diferentes configuraciones de bagging y random forest para crear un clasificador del conjunto de datos 01_churn.csv que consideren diferentes valores de los parámetros #árboles y depth
- Continuar con el notebook 08_Arboles_Churn-STUD.html

REFERENCIAS

- *Introduction to Statistical Learning with Applications in R (ISLR), G. James, D. Witten, T. Hastie & R. Tibshirani, 2014*
- *Practical Data Science with R (2nd Edition)*, Nina Zoumel & John Mount, Manning, 2019
- *R in Action*, Robert I. Kabacoff, Manning, 2015