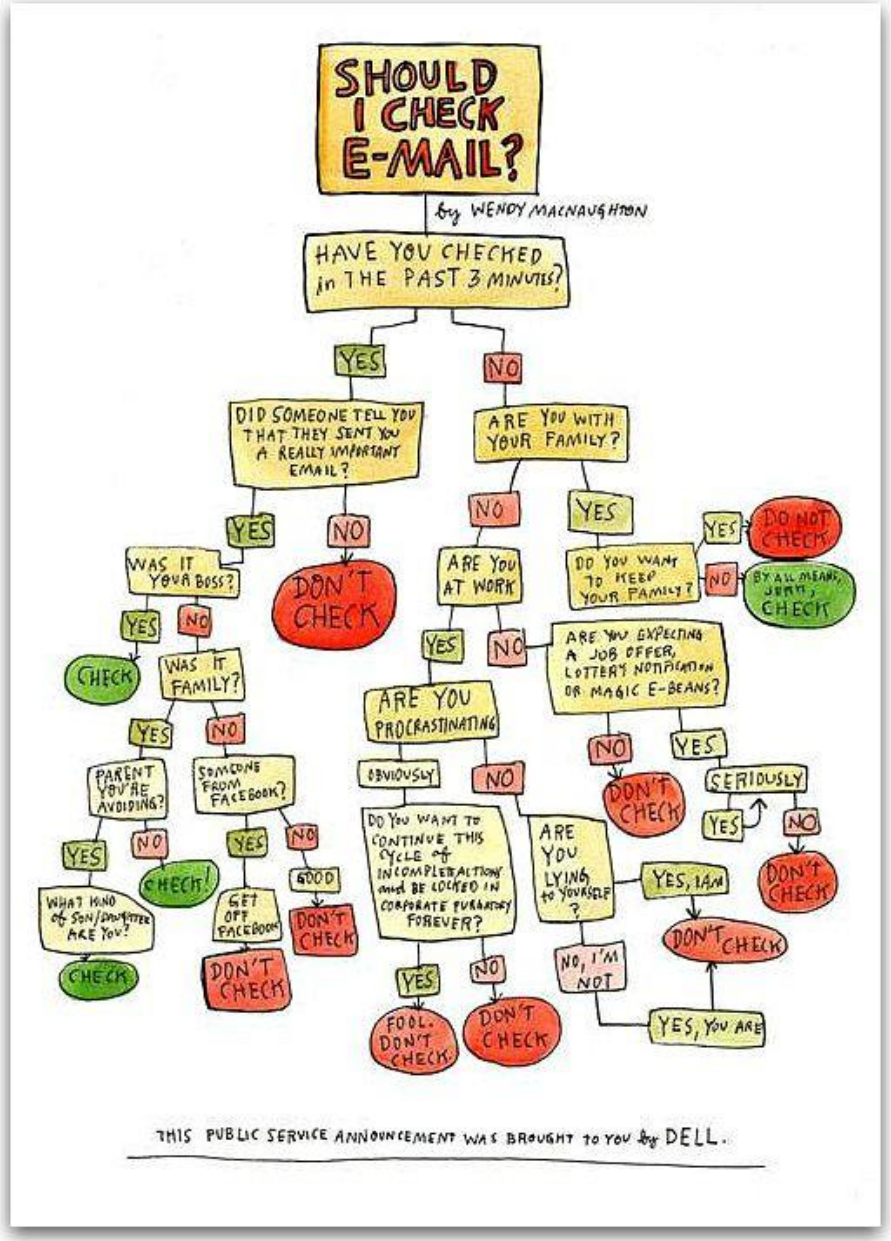
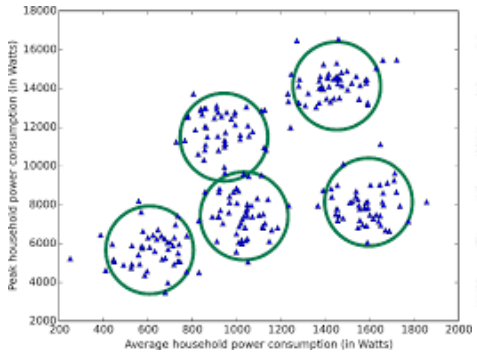
APRENDIZAJE AUTOMÁTICO

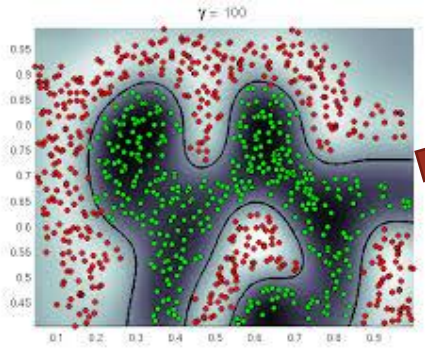


AGENDA

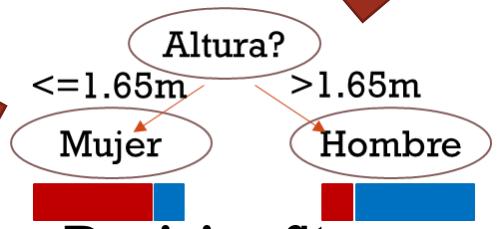
Aprendizaje automático



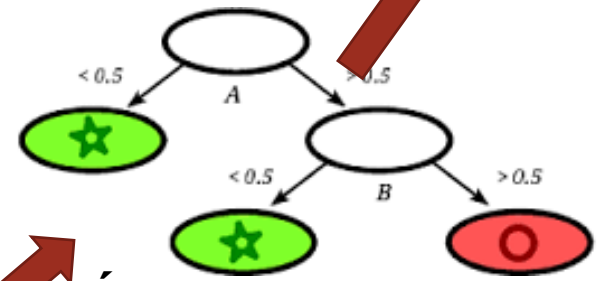
Aprendizaje no supervisado



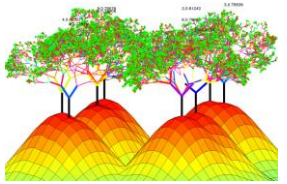
Aprendizaje supervisado



Decision Stump



Árboles de decisión



Random forest

Poda

TALLER: CLASIFICACIÓN CHURN

Vamos a analizar el dataset de churn de clientes de la empresa de telefonía móvil. La idea es poder identificar los clientes propensos a abandonar la compañía.

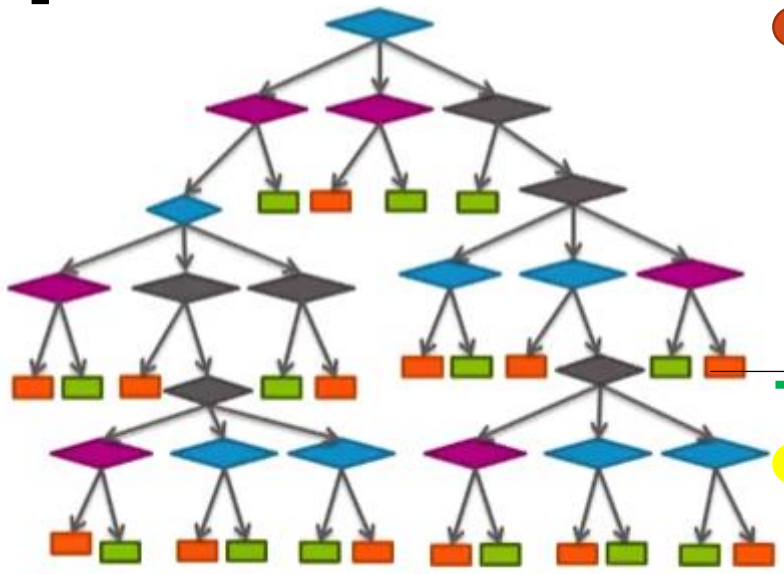
- Descargar el dataset "02_churn.csv" y el Jupyter Notebook correspondiente
- Responder a las partes 1 y 2 del taller.

PODA

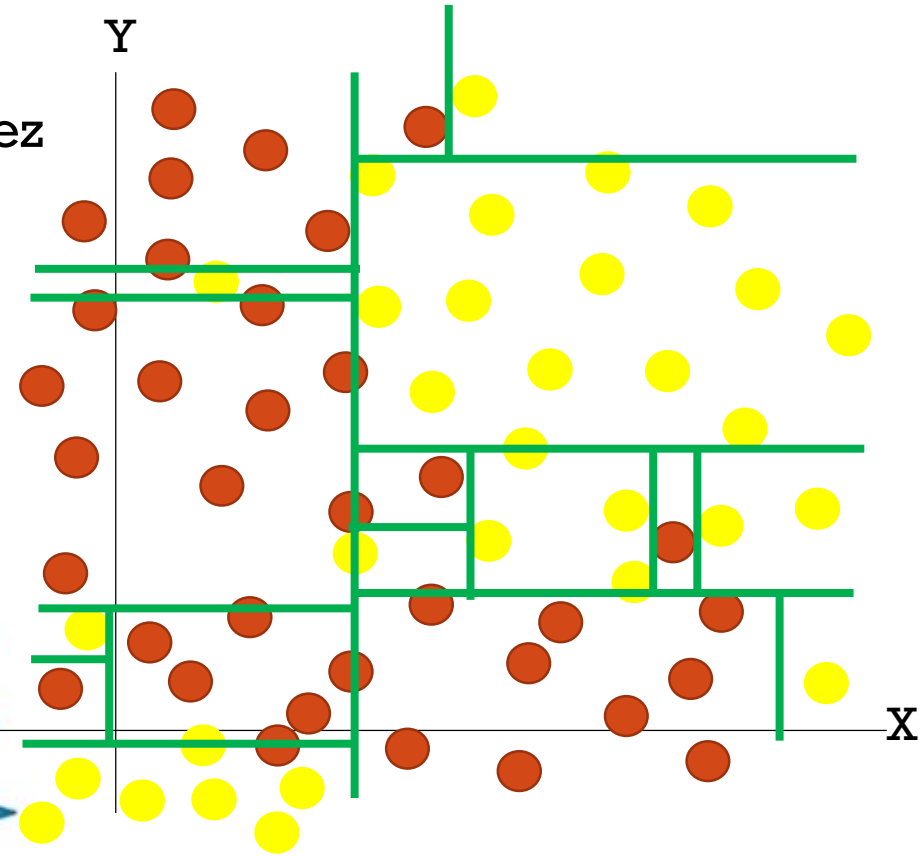
ÁRBOLES DE DECISIÓN: PODA

Los árboles de decisión tienden a llegar a la perfección absoluta por medio de reglas cada vez más complejas → **overfitting**

Compromiso a encontrar entre poder de clasificación y **simplicidad** de los árboles



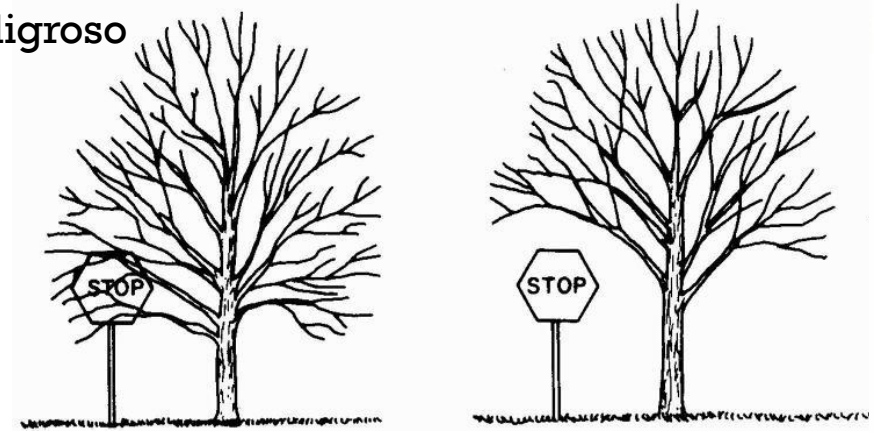
Carlos Guestrin, 2015

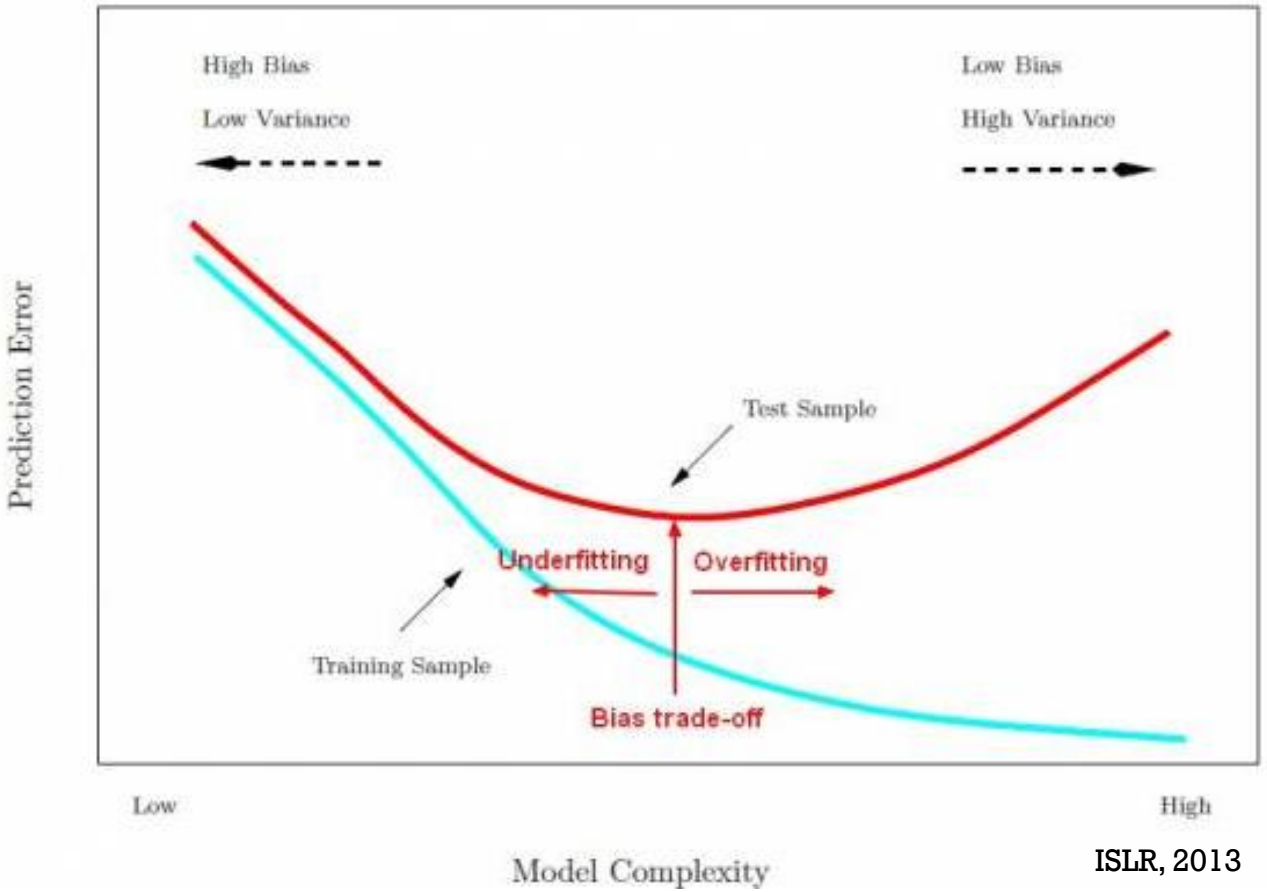


ÁRBOLES DE DECISIÓN: PODA

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

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





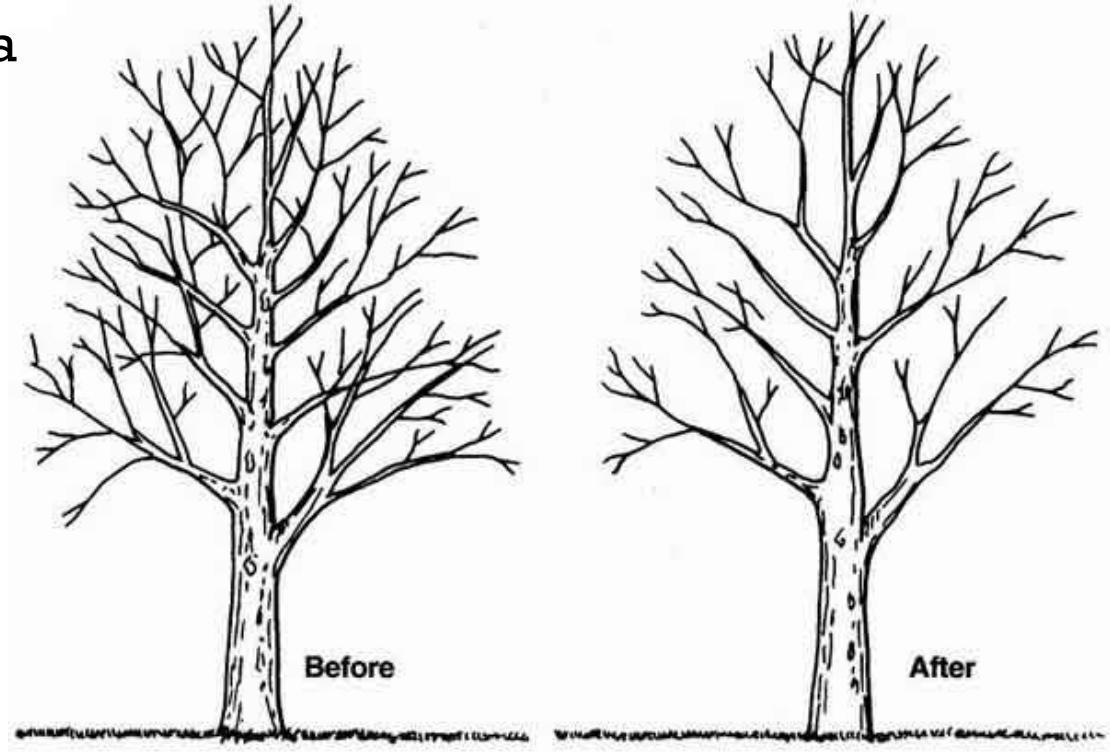
ISLR, 2013

ÁRBOLES DE DECISIÓN: PODA

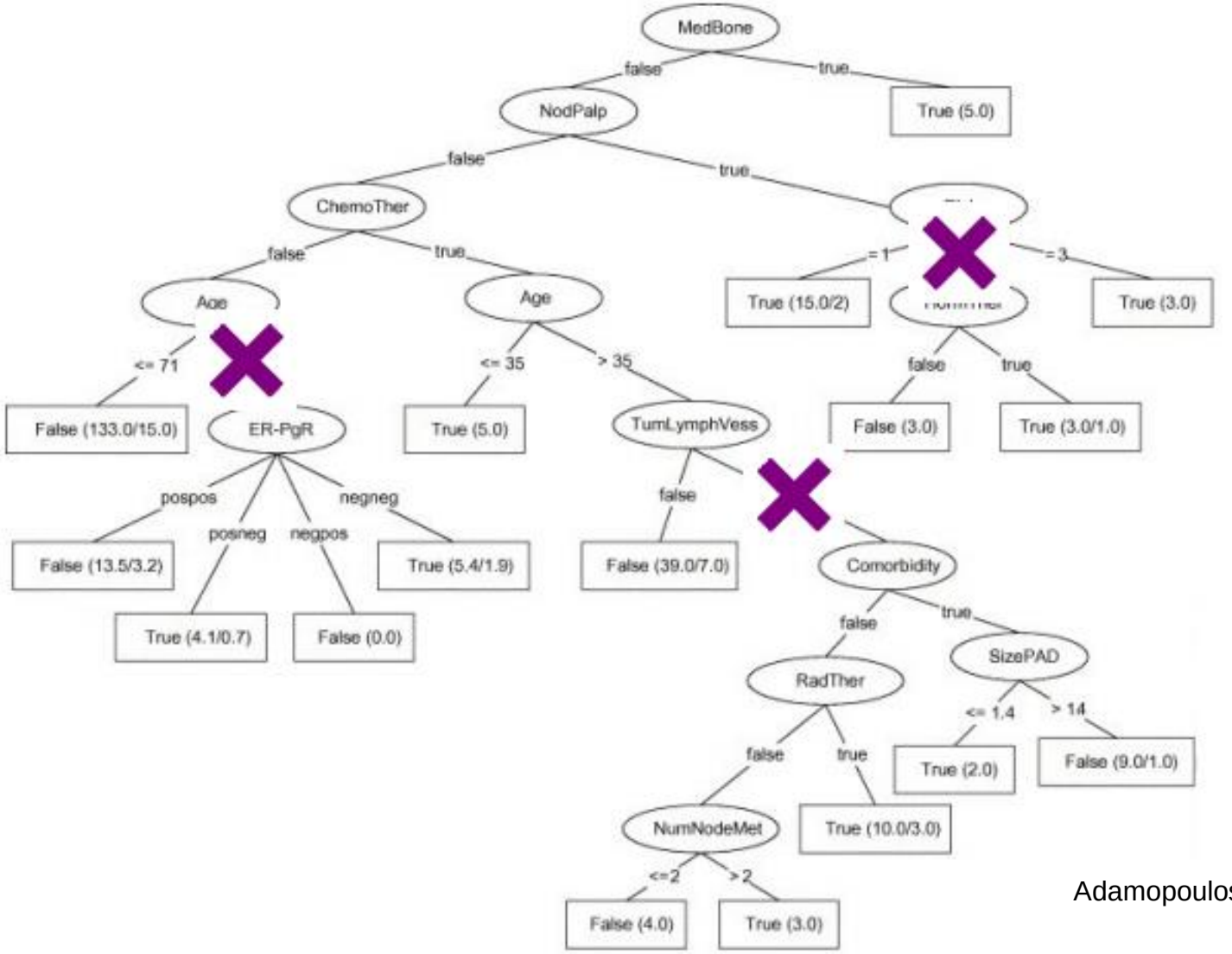
Post-poda: simplificar el árbol una vez se haya terminado de aprender:

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

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



ÁRBOLES DE DECISIÓN: PODA

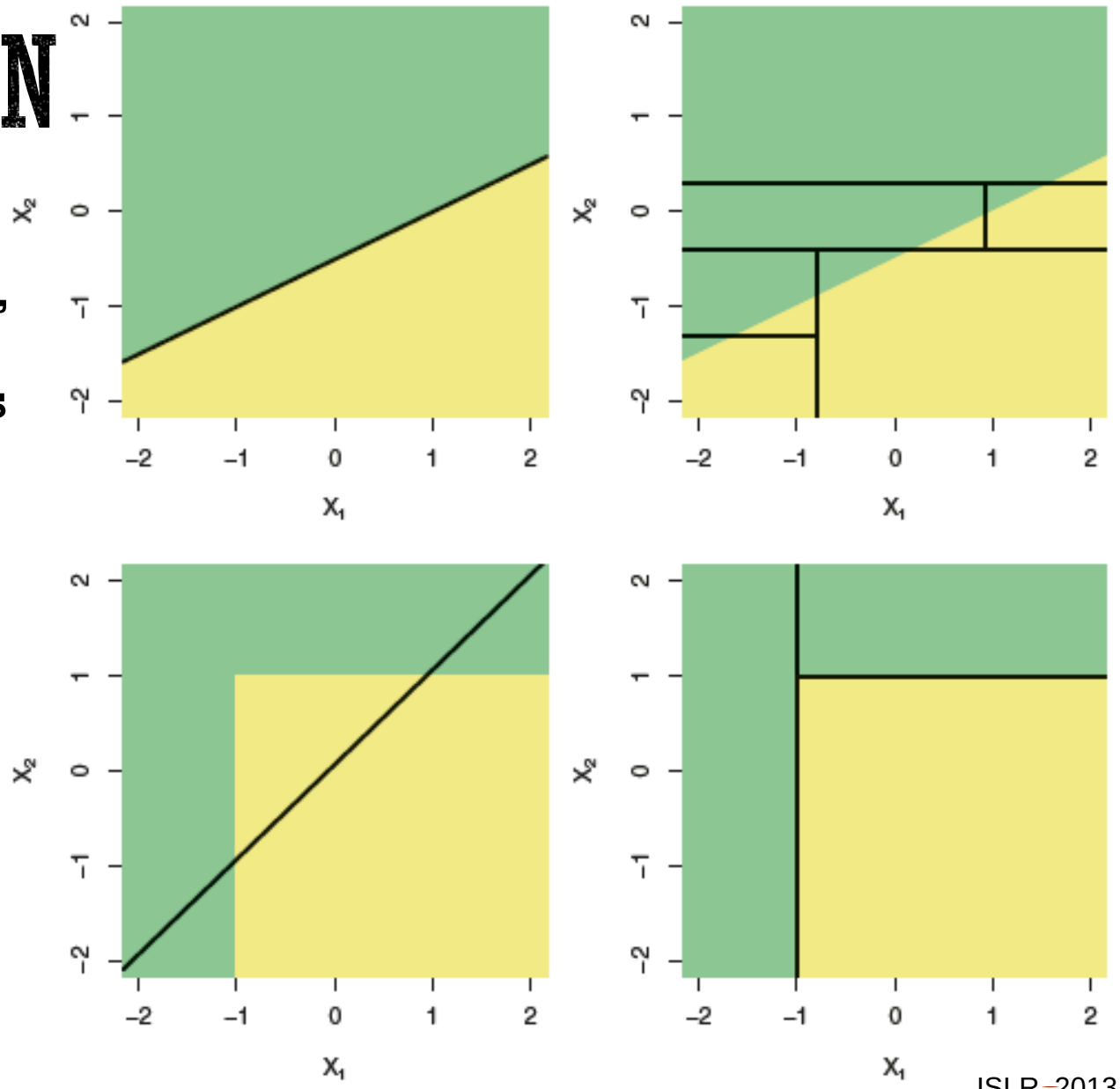


Adamopoulos, NYU, 2014

ÁRBOLES DE DECISIÓN

Consideraciones:

- Su aplicación no siempre es la ideal, dependiendo de los datos: escogencia entre árboles y modelos lineales
- No son los algoritmos más competitivos, pero producen un modelo simple, altamente interpretable, que semeja el razonamiento humano
- Tienden hacia el overfitting, por lo que hay que utilizar estrategias de poda



ISLR, 2013

ÁRBOLES DE DECISIÓN

Consideraciones:

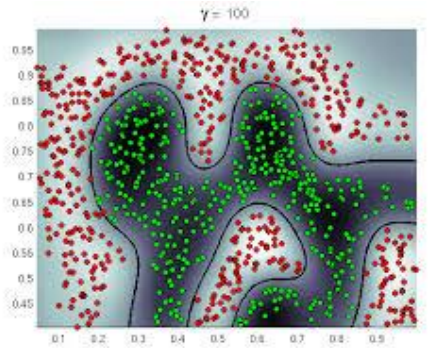
- Son indiferentes de cuestiones de escala y forma de la distribución de los datos
- Son robustos a la presencia de missing values
- Permiten la consideración de atributos cualitativos sin necesidad de crear variables adicionales
- Son vulnerables a casos de incoherencia de las instancias de aprendizaje
- Son sensibles a bases de datos desbalanceadas y a pequeños cambios en los datos
- Son sesgados hacia los atributos con mayor cantidad de valores
- Son la base de modelos de conjuntos de clasificadores (Bagging, Boosting, Random forest), que ofrecen un excelente poder predictivo, mientras reducen el error de varianza (overfitting)

TALLER: CLASIFICACIÓN CHURN

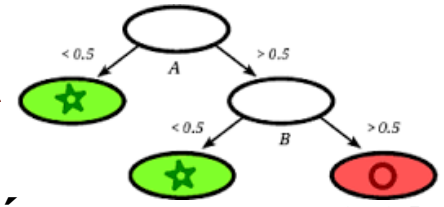
Vamos a analizar el dataset de churn de clientes de la empresa de telefonía móvil. La idea es poder identificar los clientes propensos a abandonar la compañía.

- Descargar el dataset "02_churn.csv" y el Jupyter Notebook correspondiente
- Responder a la parte 3 del taller.

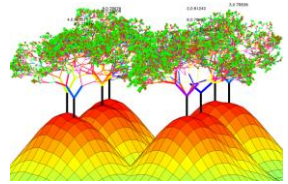
AGENDA



Aprendizaje supervisado



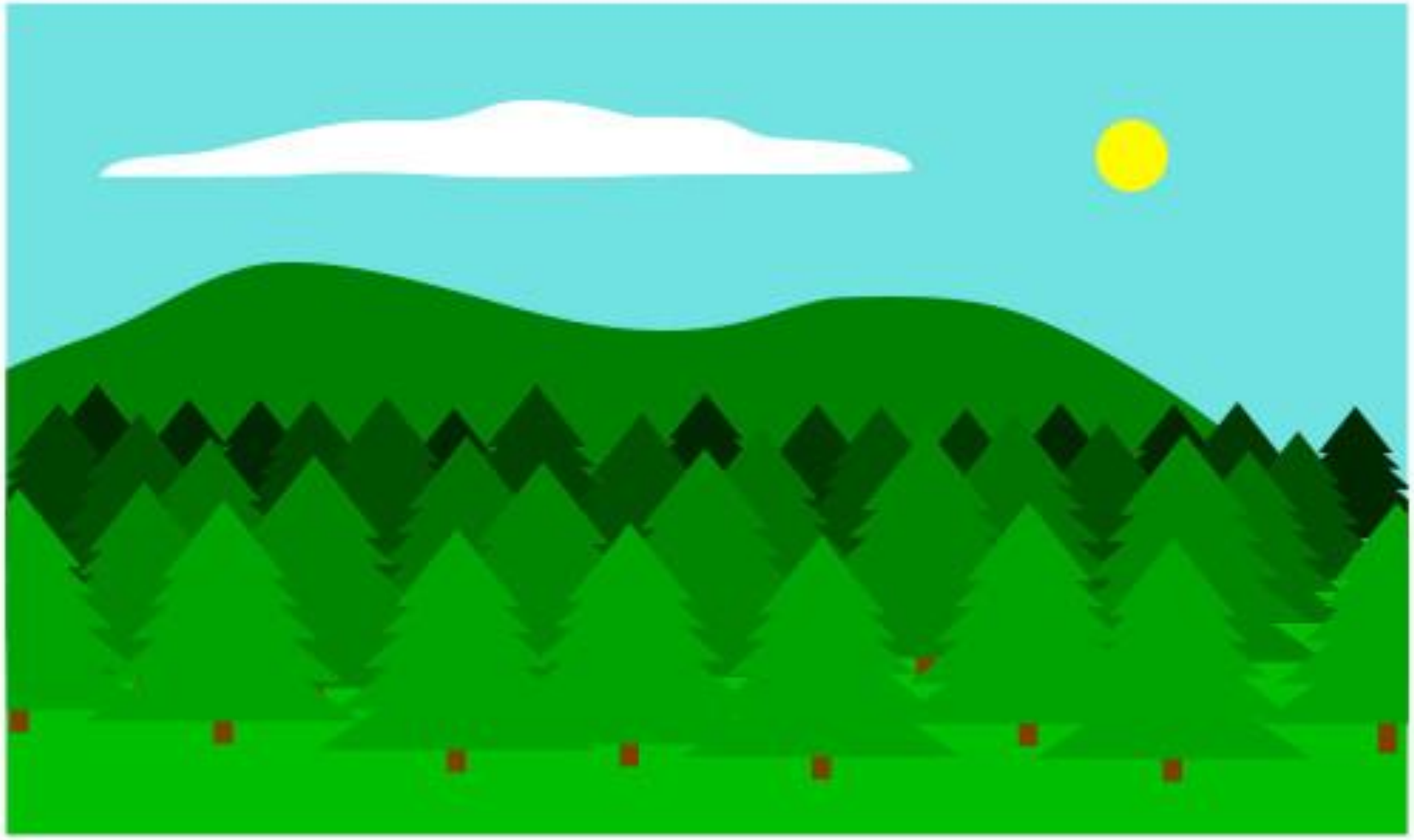
Árboles de decisión



Random forest

Poda

ENSEMBLE LEARNING



ENSEMBLE LEARNING

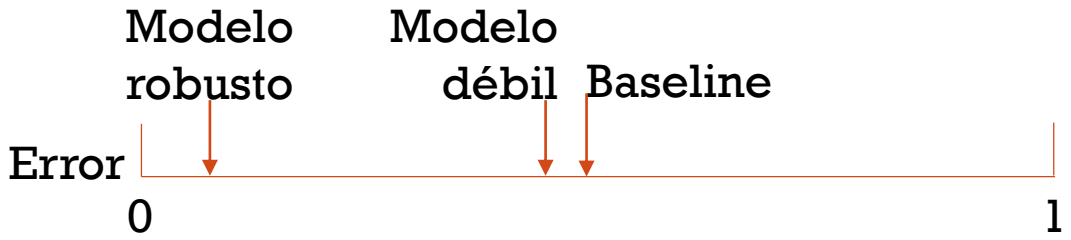
- Diversificación, sabiduría de las multitudes:
  - Cada modelo tiene un riesgo de mala predicción de datos futuros
  - ¿por qué limitar la decisión a un solo modelo si podemos construir y utilizar varios?

  → Combinar un grupo de clasificadores/regresores en un modelo de conjunto

  → Decisión basada en la agregación de varios modelos (max / promedio)
- Búsqueda de un mejor compromiso entre sesgo y varianza en el modelo de conjunto

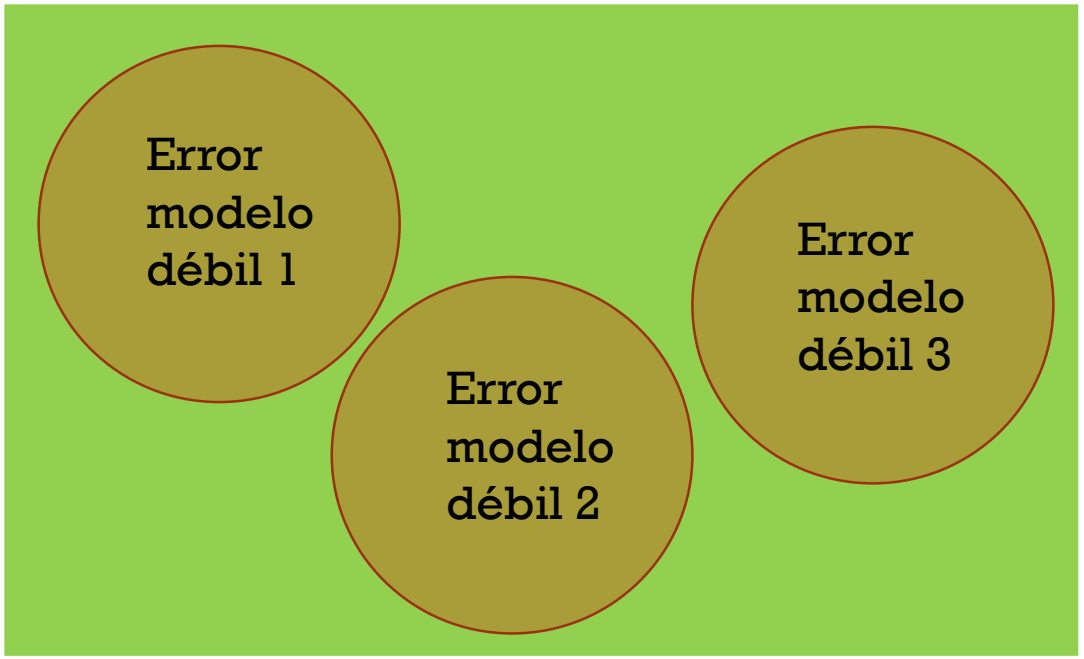
ENSEMBLE LEARNING

Vamos a distinguir entre modelos robustos y débiles



- Los modelos robustos son complejos y proporcionan modelos mucho mejores a los del baseline. Por ejemplo: árboles de decisión profundos.
- Los modelos débiles son sencillos y solo proporcionan resultados que son solo un poco mejores a los del baseline. Por ejemplo: árboles de decisión de poca profundidad.

BAGGING



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

BAGGING

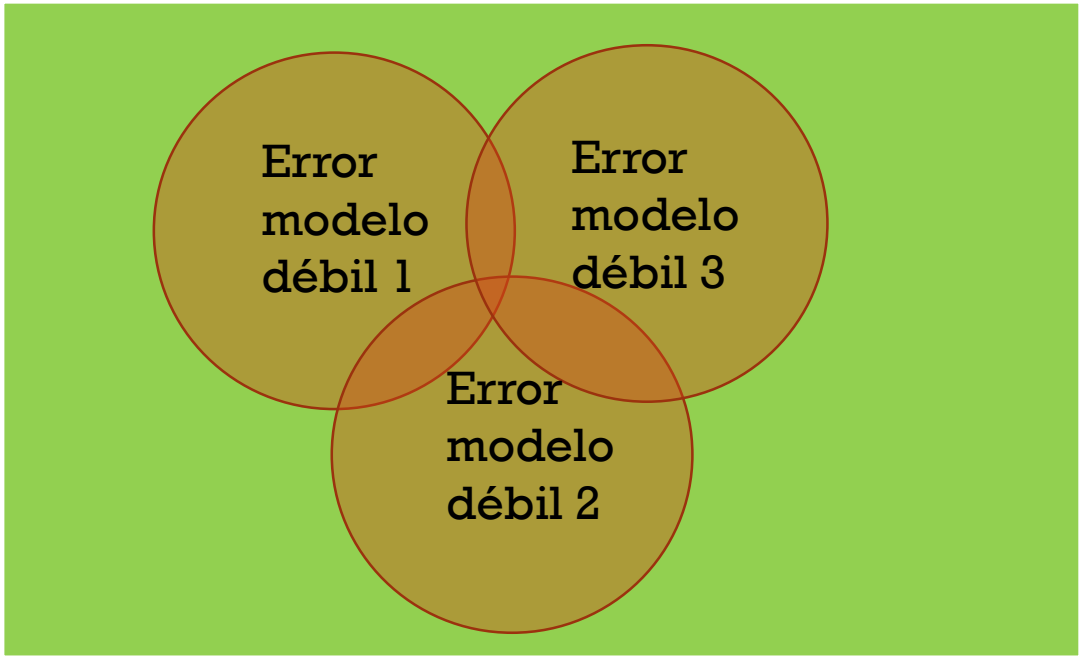
- Bagging = Bootstrap aggregating
- Bootstrap: técnica de evaluación basada en el remuestreo
- Algoritmo:
  1. Escoger múltiples subconjuntos de instancias con repetición (remuestreo)
  2. Entrenar varias instancias de un mismo modelo de aprendizaje **robusto** basado en cada subconjunto
  3. Agregar las decisiones de cada clasificador en una sola decisión global (votación, promedio)
  4. Muy útil cuando se trata de modelos no lineales (e.g. árboles de decisión).
- Resultado: sesgo similar, reducción de varianza

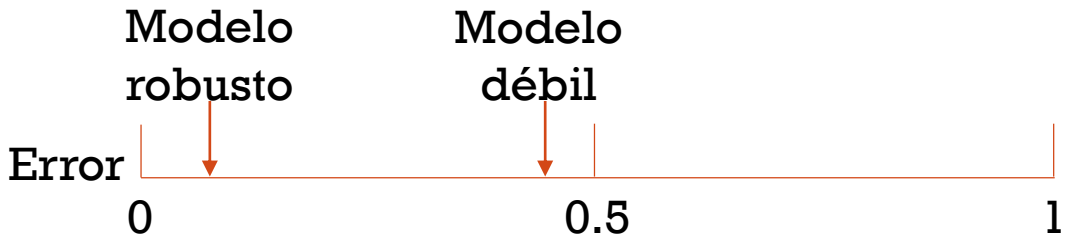
BAGGING

- Se aplica tanto a clasificadores como a regresores
- Todos los modelos son independientes entre ellos y se pueden crear en paralelo
- Dependiendo de como se escogen los subconjuntos de datos podemos distinguir distintas variantes:
  - **Pasting**: subconjuntos de registros aleatorios sin reemplazo
  - **Bagging**: subconjuntos de registros aleatorios con reemplazo
  - **Random subspaces**: subconjuntos de las variables independientes
  - **Random patches**: subconjuntos tanto de los registros como de las variables independientes

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?

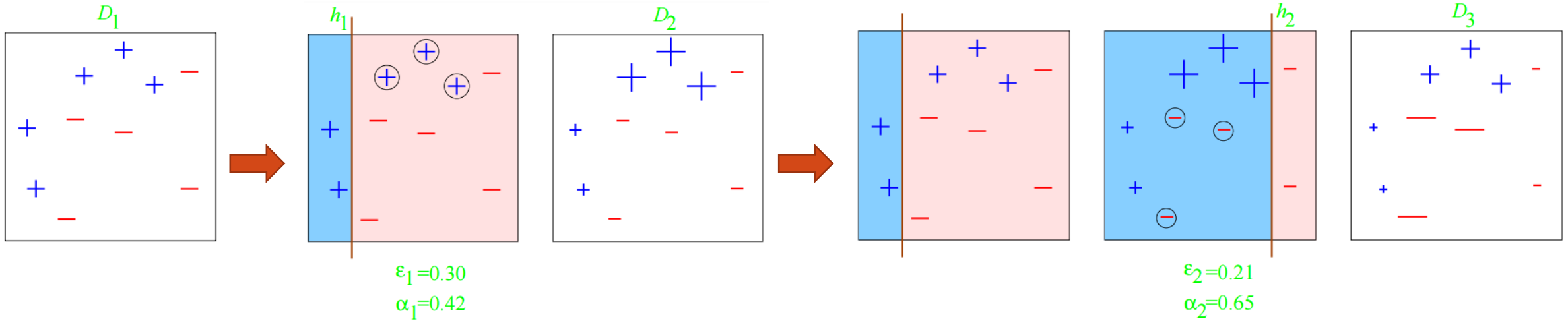
BOOSTING

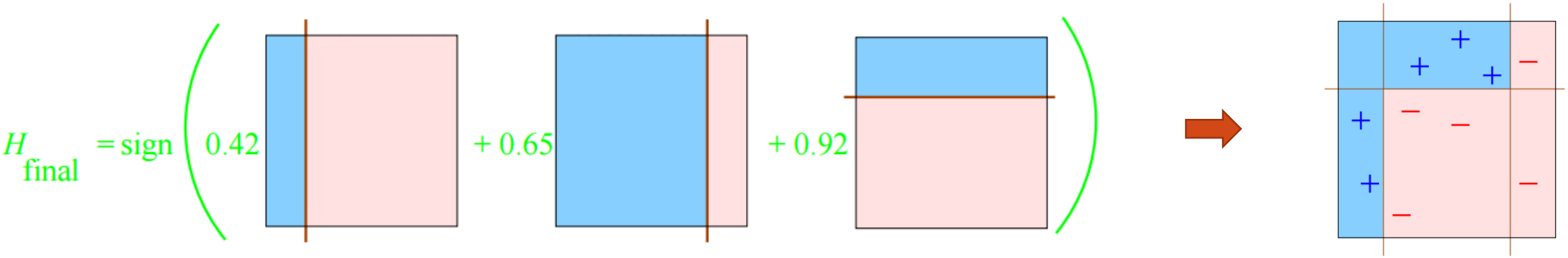
- La importancia de cada registro se modifica según los resultados del último modelo entrenado → Los modelos son **dependientes** entre ellos y se interesan por mejorar los erros de los modelos anteriores
- Algoritmo
  1. Crear un modelo predictor inicial sobre el dataset
  2. Asignar un peso a la decisión 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. Iterar y crear un nuevo predictor
  5. Agregar una decisión ponderada de todos los predictores creados

BOOSTING

- Se aplica tanto a clasificadores como a regresores
- Agregación de predictores **débiles**; por lo general se utilizan árboles de decisión muy simples (decision stump) → aumenta el error del sesgo pero se sobre compensa con la mejora del error de varianza

BOOSTING - ADABOOST





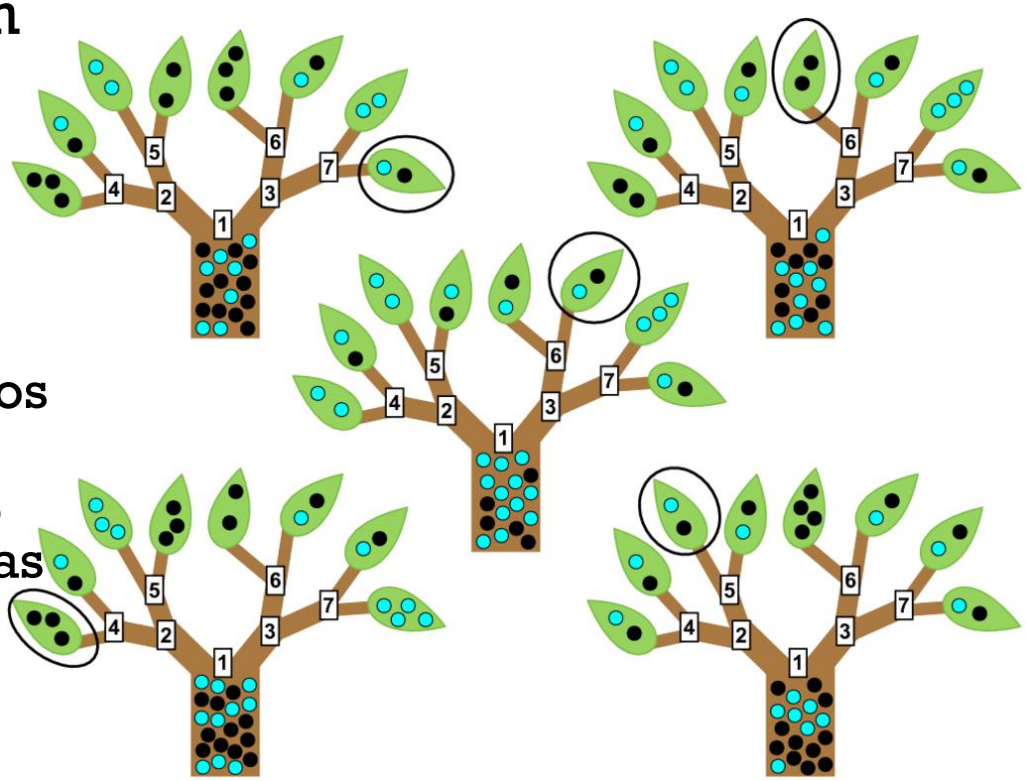
www.cs.princeton.edu/~schapire

BOOSTING

- Pesos de las n instancias normalizados (suma da 1), inicialmente son iguales a 1/n
- A medida que se itera se actualizan los pesos de cada instancia, de tal manera que:
  - La suma de todas las instancias correctas debe dar ½
  - La suma de todas las instancias incorrectas debe dar ½
- El peso de cada modelo está dado por $\alpha = \frac{1}{2} ln \frac{1-error}{error}$
- No presenta **overfitting** a primera vista, aunque todavía no se ha podido demostrar matemáticamente

RANDOM FOREST

- Junto a boosting, uno de los algoritmos con mejor performance
- Extensión de bagging basado en árboles CART
- Algoritmo
  1. Crear múltiples árboles de decisión, construidos sobre muestras aleatorias del dataset
  2. En cada punto de partición, hacer un bootstrap de las variables consideradas (solo unas cuantas en cada nodo)
  3. Agregar los resultados de todos los árboles creados
- Buen poder predictivo, pero lento (gran número de árboles) y difícil de interpretar



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

TALLER: CLASIFICACIÓN CHURN

Vamos a analizar el dataset de churn de clientes de la empresa de telefonía móvil. La idea es poder identificar los clientes propensos a abandonar la compañía.

- Descargar el dataset "02_churn.csv" y el Jupyter Notebook correspondiente
- Desarrollar la parte final del taller.

REFERENCIAS

- *Python Machine Learning*, Sebastian Raschka, Packt, 2015
- *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 decision trees*, Carlos Guestrin, University of Washington