

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación



IIC2613 - Inteligencia Artificial

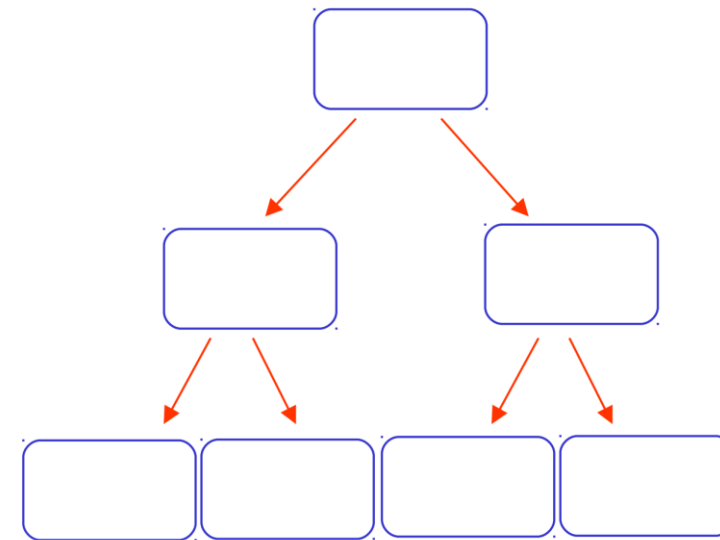
Random Forests

Hans Löbel

Dpto. Ingeniería de Transporte y Logística
Dpto. Ciencia de la Computación

Árboles de decisión son ampliamente utilizados en la práctica

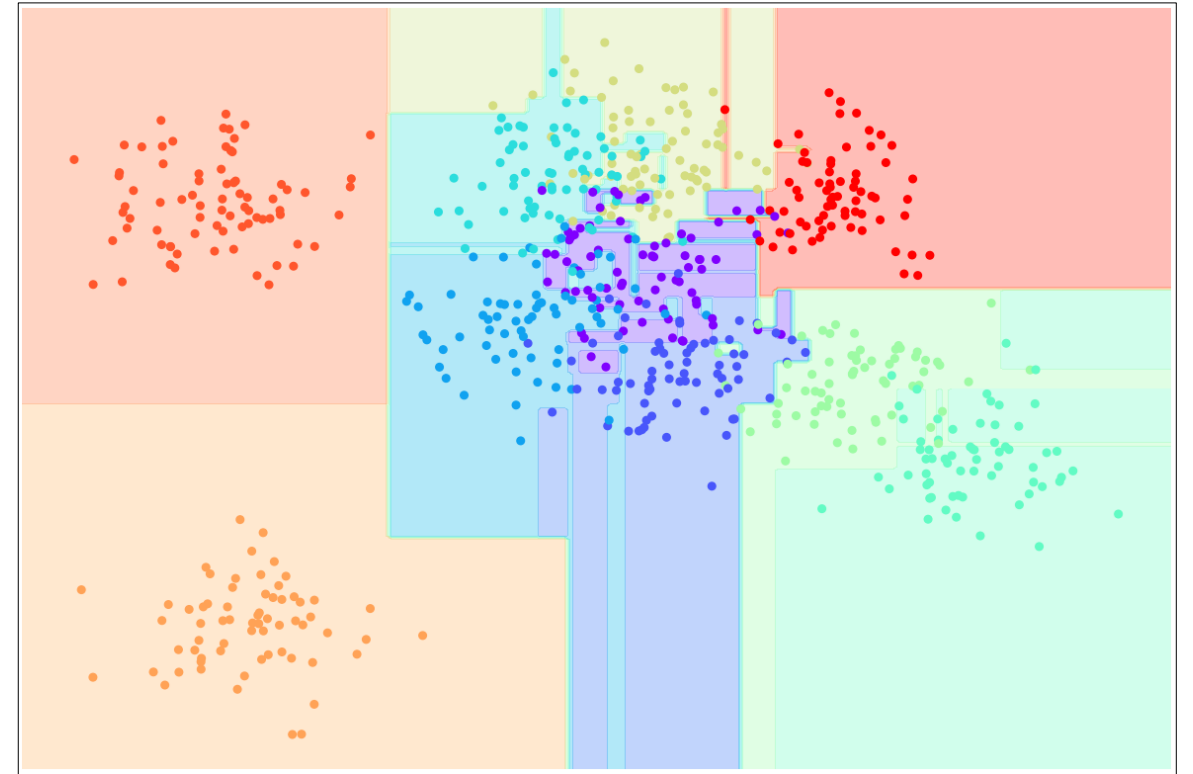
- Gran ventaja radica en la simplicidad y facilidad de interpretación.
- Pueden sufrir de serios problemas de sobreajuste.



¿Podemos hacer algo más para evitar el sobreajuste en los árboles?

¿De dónde proviene este problema?

- Muestra es progresivamente reducida al crear nuevos nodos en el entrenamiento (al bajar en el árbol).
- Si hay muchos atributos, existe alta probabilidad de elegir alguno bueno en los datos de entrenamiento, pero “inútil” en cuanto a generalización.



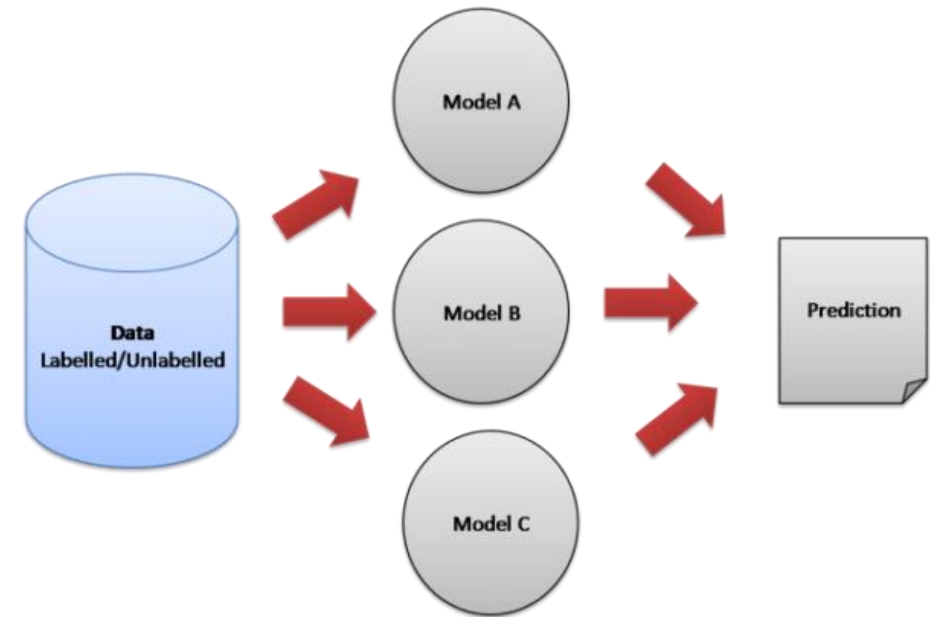
Busquemos un enfoque distinto para enfrentar el sobreajuste

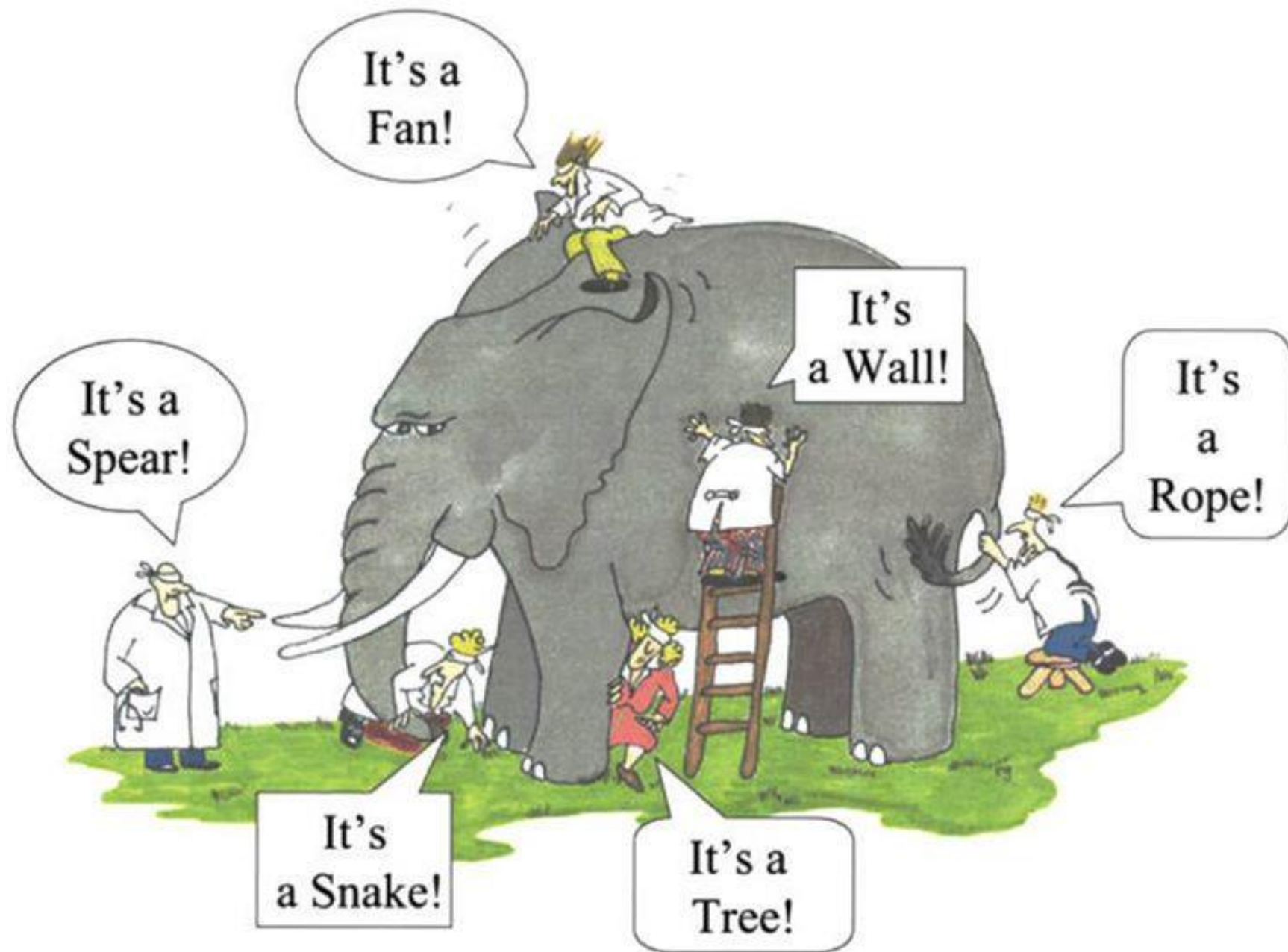
- Consideremos un escenario donde queremos estimar la probabilidad de ocurrencia de un evento.
- En ausencia de un experto, ¿cuál es la mejor manera de realizar la estimación?
- Una encuesta permite estimar la probabilidad de un evento a partir de un número “limitado” de “mediciones”
- La muestra y la metodología juegan un rol fundamental para el éxito de la estimación.



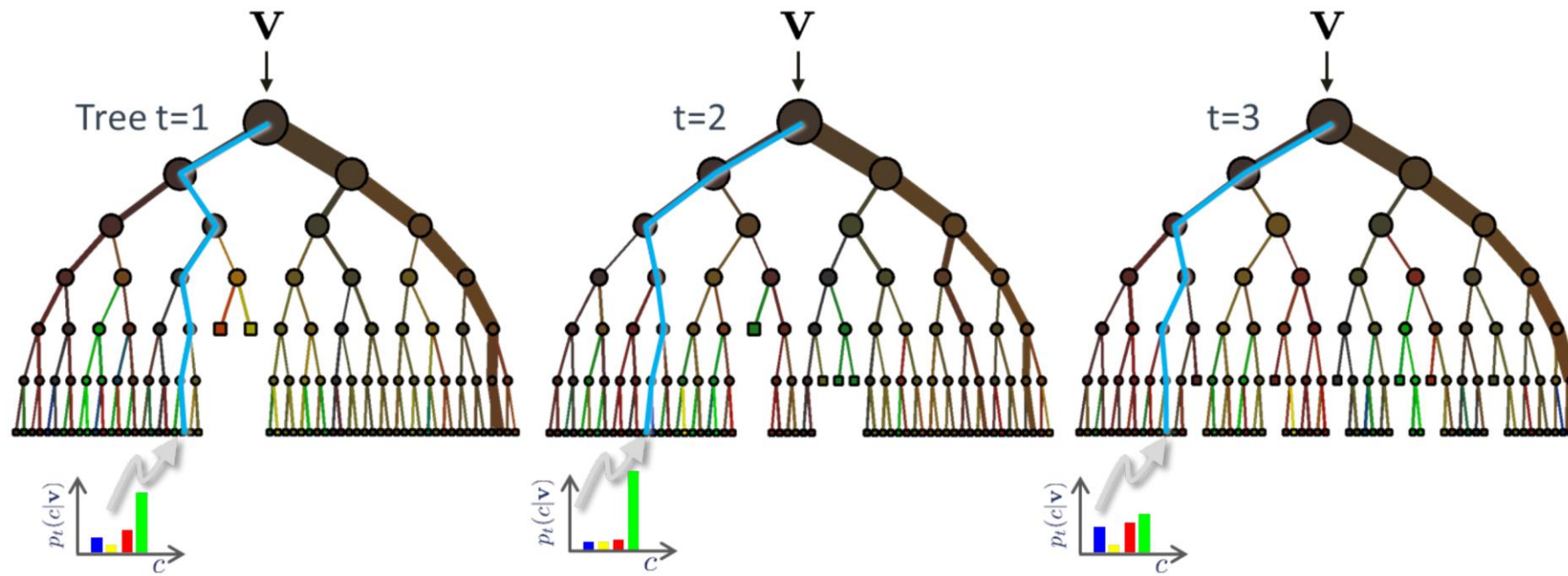
Busquemos un enfoque distinto para enfrentar el sobreajuste

- Desde una perspectiva de ML, si queremos estar seguros de algo, le podríamos preguntar a más de un modelo.
- Así, si le preguntamos a suficientes, es probable que la moda o el promedio de las predicciones sean cercanas a la realidad.
- ¿Basta entonces con entrenar muchos modelos para eliminar el sobreajuste?



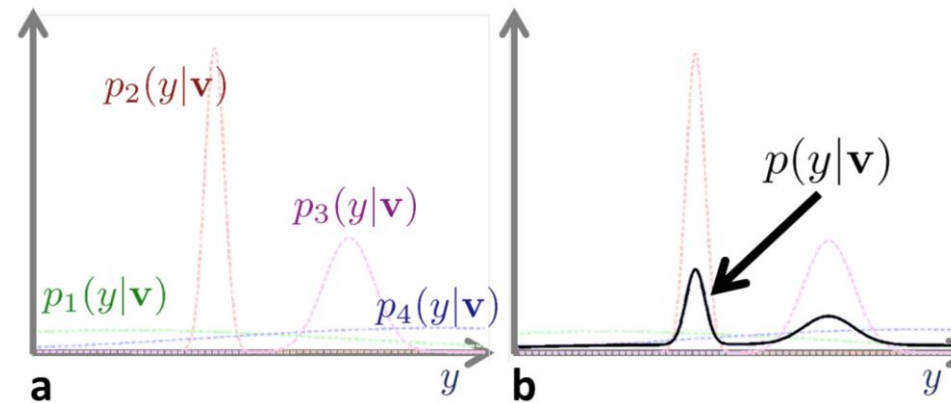


¿Qué condición deben cumplir los modelos para que esta idea funcione?



Ensamblados de modelos con **baja correlación**

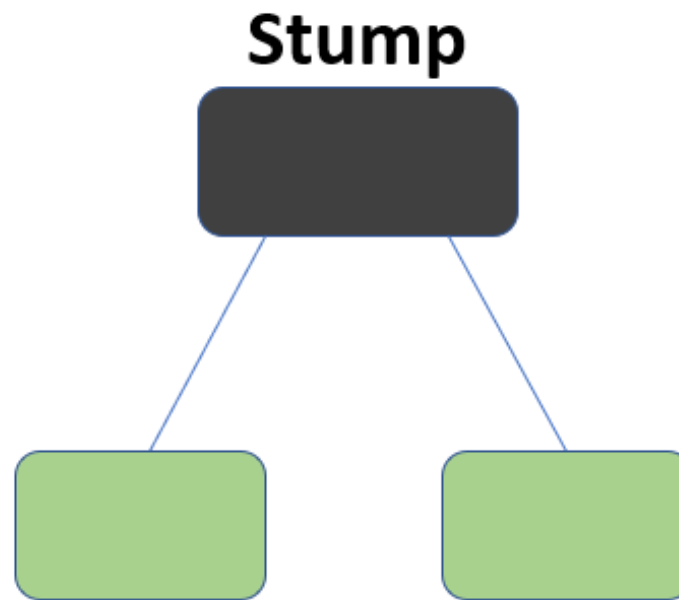
- Si el **patrón de error** (acierto) es **distinto** para todos, es altamente probable que, en promedio, la respuesta del ensamble sea correcta (**errores se cancelan**).
- Para que esto ocurra, todos los **modelos** del ensamble deben ser **débiles**, es decir, ser un poco mejores que dejar la decisión al azar.



¿Cómo logramos esto con los árboles?

La solución parte por restringir al clasificador

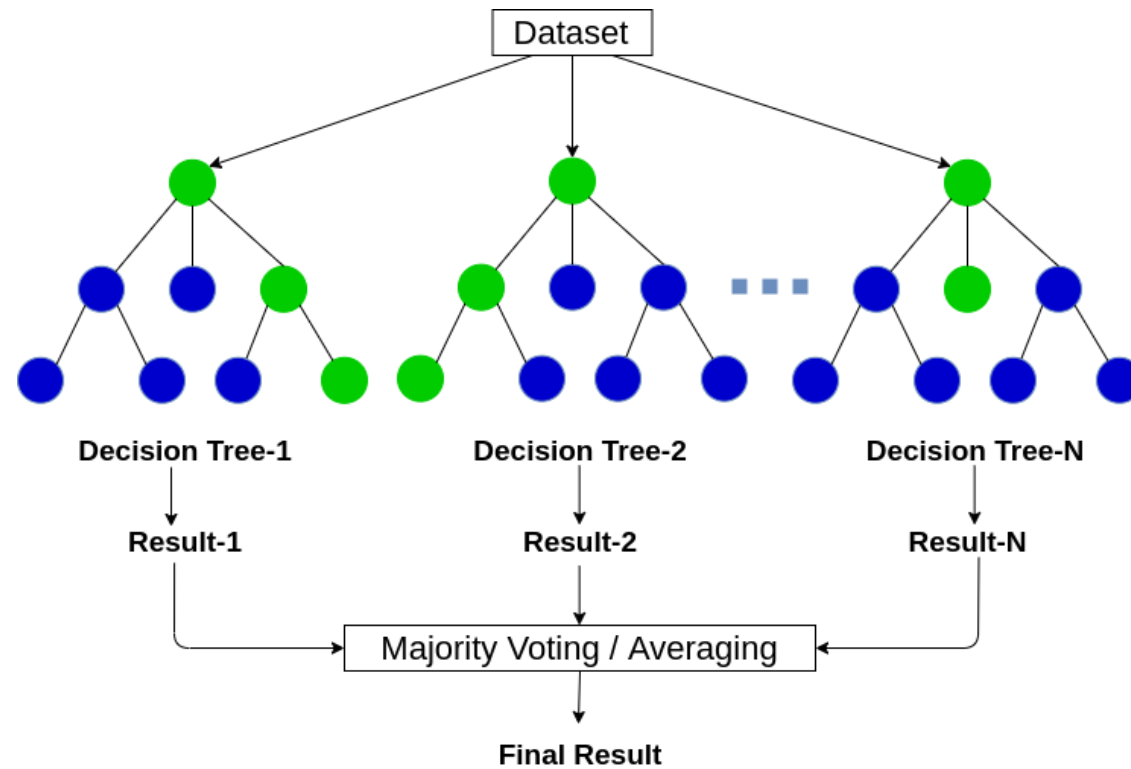
- Si el árbol tiene baja profundidad, es probable que tenga poca evidencia y tome decisiones débiles.
- El caso extremo para esto es un *Decision Stump*, o árbol de profundidad igual a 1.



¿Es esto suficiente?

La solución sigue con las **muestras aleatorias**

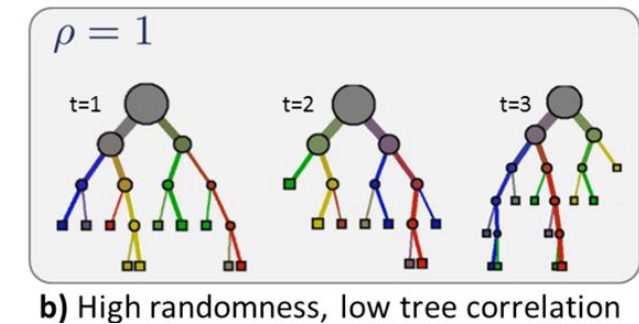
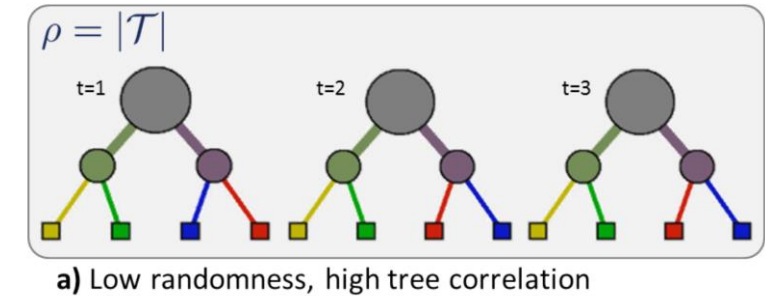
- Si cada modelo ve sólo una parte de los datos (elegida aleatoriamente), es probable que la correlación entre ellos disminuya.



¿Qué pasa si hay una variable muy buena?

La solución está nuevamente en las **muestras aleatorias**, pero ahora de **atributos**

- Cada vez que se hace un **split**, se elige un **subconjunto aleatorio** de los atributos.
- Luego, para cada uno de ellos, se calcula la ganancia de información.
- Otra ventaja de esto es que se limita la profundidad del árbol, reduciendo la complejidad de este.
- Por lo general, se utiliza una muestra proporcional a la **raíz del número de atributos**.



La introducción de todas estas estrategias “aleatorizantes”, transforma a los árboles en *Random Forests*

Algorithm 15.1 *Random Forest for Regression or Classification.*

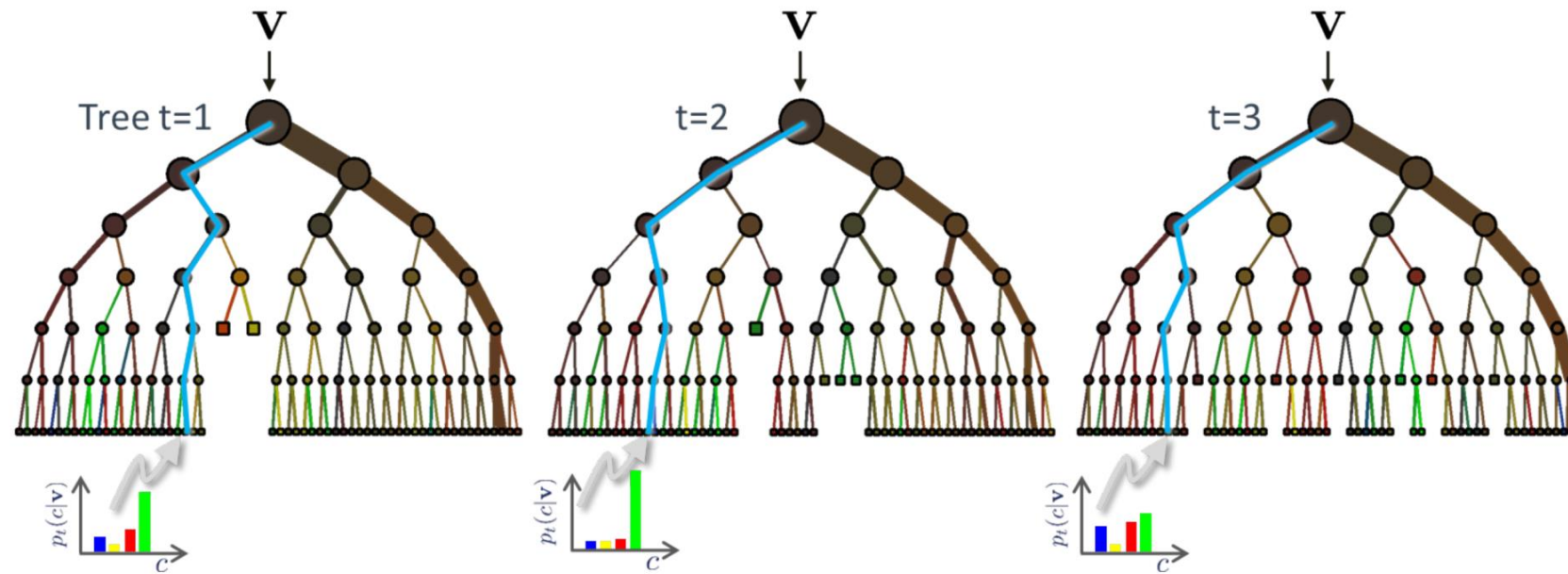
1. For $b = 1$ to B :
 - (a) Draw a bootstrap sample \mathbf{Z}^* of size N from the training data.
 - (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
 - i. Select m variables at random from the p variables.
 - ii. Pick the best variable/split-point among the m .
 - iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.

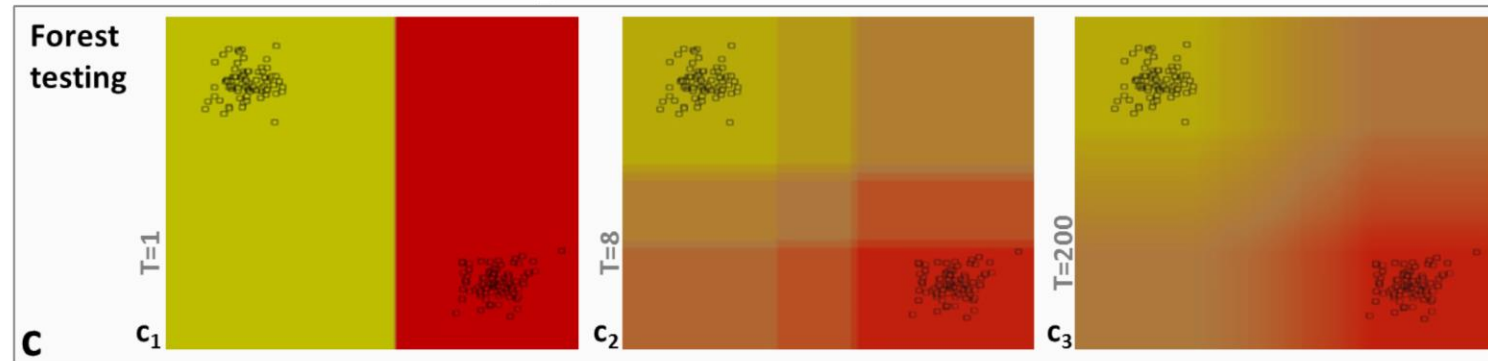
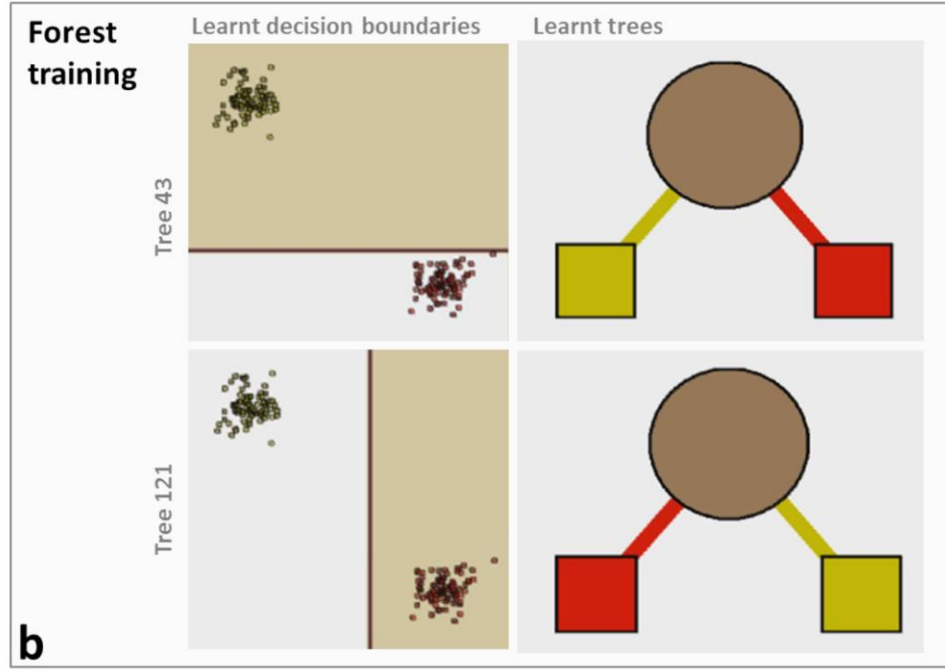
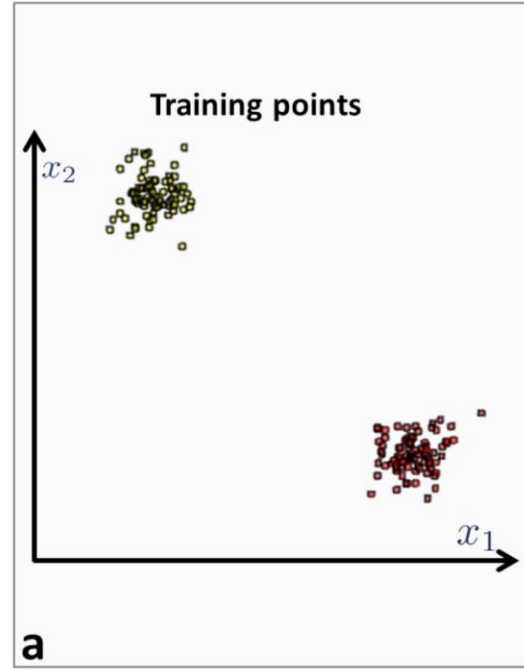
To make a prediction at a new point x :

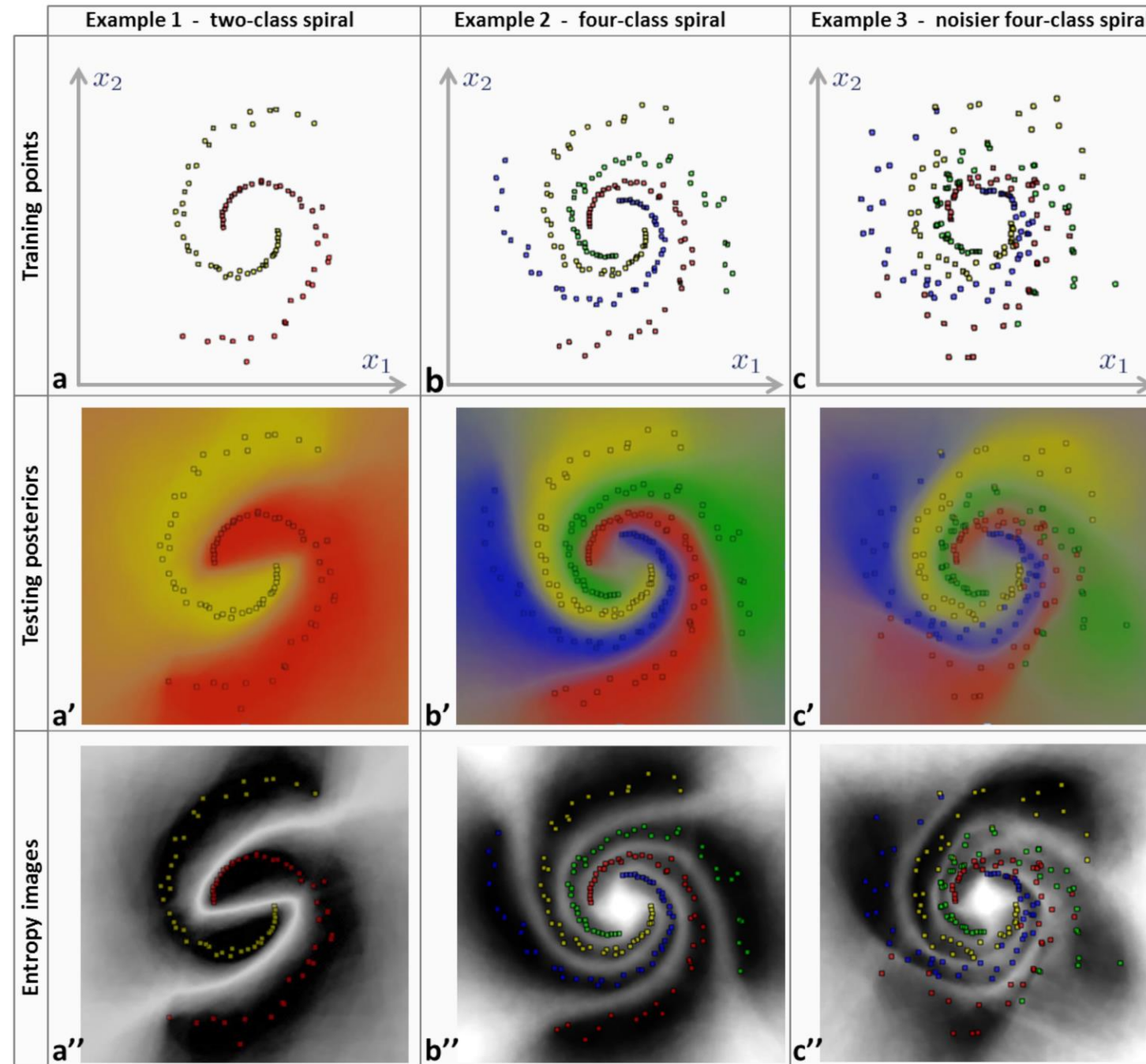
Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B} \sum_{b=1}^B T_b(x)$.

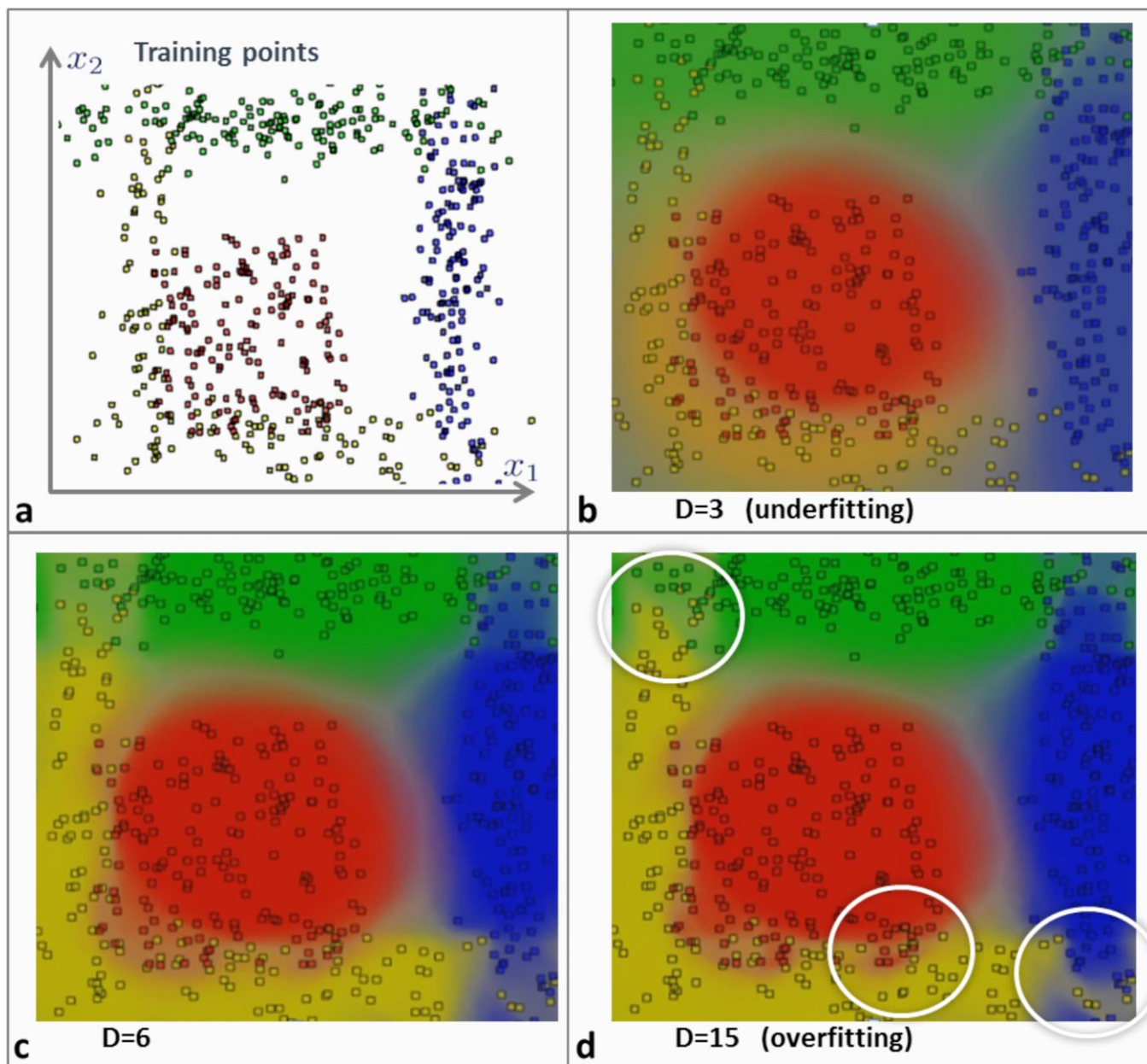
Classification: Let $\hat{C}_b(x)$ be the class prediction of the b th random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \text{majority vote } \{\hat{C}_b(x)\}_1^B$.

La introducción de todas estas estrategias “aleatorizantes”,
transforma a los árboles en *Random Forests*



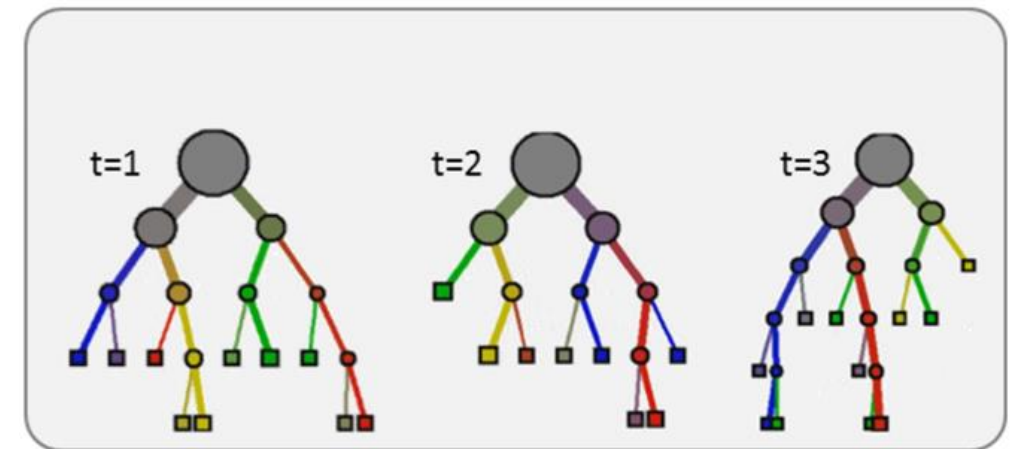






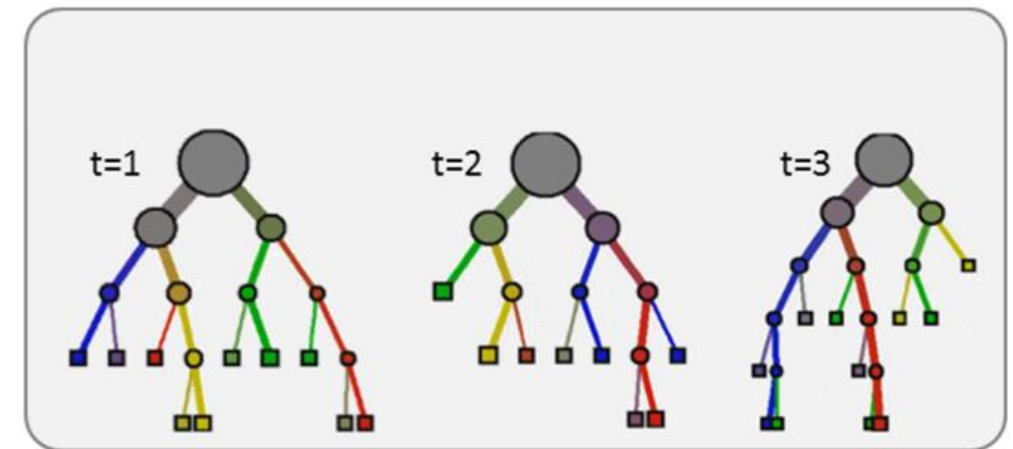
Algunas consideraciones relacionadas con la evaluación

- ¿Cómo podemos cuantificar el poder de generalización de un Random Forest, utilizando únicamente el conjunto de entrenamiento?
- ¿Cómo podemos identificar ahora las variables más relevantes?



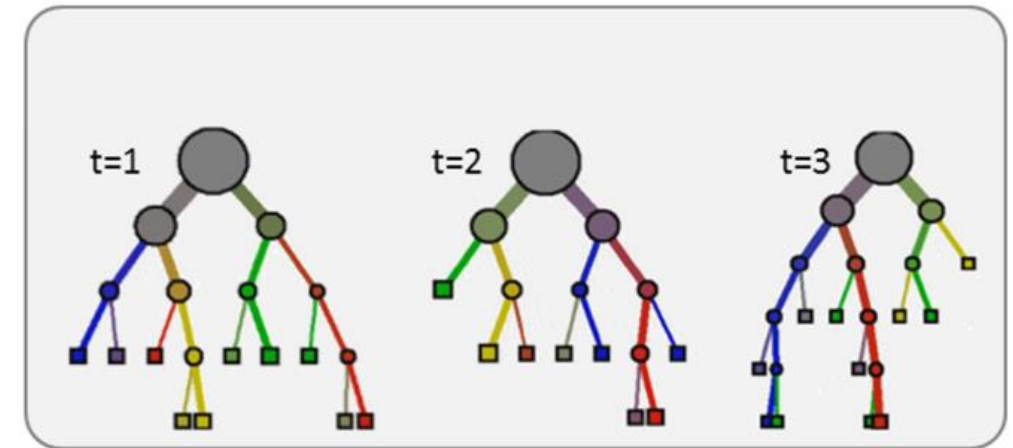
Algunas consideraciones relacionadas con la evaluación

- Es posible cuantificar la generalización directamente a partir del set de entrenamiento, usando ejemplos no vistos por lo árboles (*out-of-bag error*).
- Para medir la *importancia* de una variable, basta con *permutar su valor entre los ejemplos* (¿por qué?).



Random Forests son ampliamente utilizados en la práctica

- Al igual que los árboles de decisión, son modelos fácilmente interpretables.
- Rendimiento es altamente competitivo en problemas con datos *tabulados*.
- Gracias a aleatorización en su construcción, son altamente *resistentes* al *sobreajuste*.



Vamos a Colab...



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación



IIC2613 - Inteligencia Artificial

Random Forests

Hans Löbel

Dpto. Ingeniería de Transporte y Logística
Dpto. Ciencia de la Computación