pepper money

Sesión 9

Bosques Aleatorios (Random Forest)

Based on material by: Ari Silburt

https://github.com/silburt/RF_DeepDive

Los Bosques Aleatorios pueden parecer intimidantes…

¡Pero en realidad no son tan malos!

Plan

- Árbol de Decisión
- Random Forests (+ Bagging)
- Optimización del Árbol
- Prunning
- Notas Finales

Árbol de Decisión

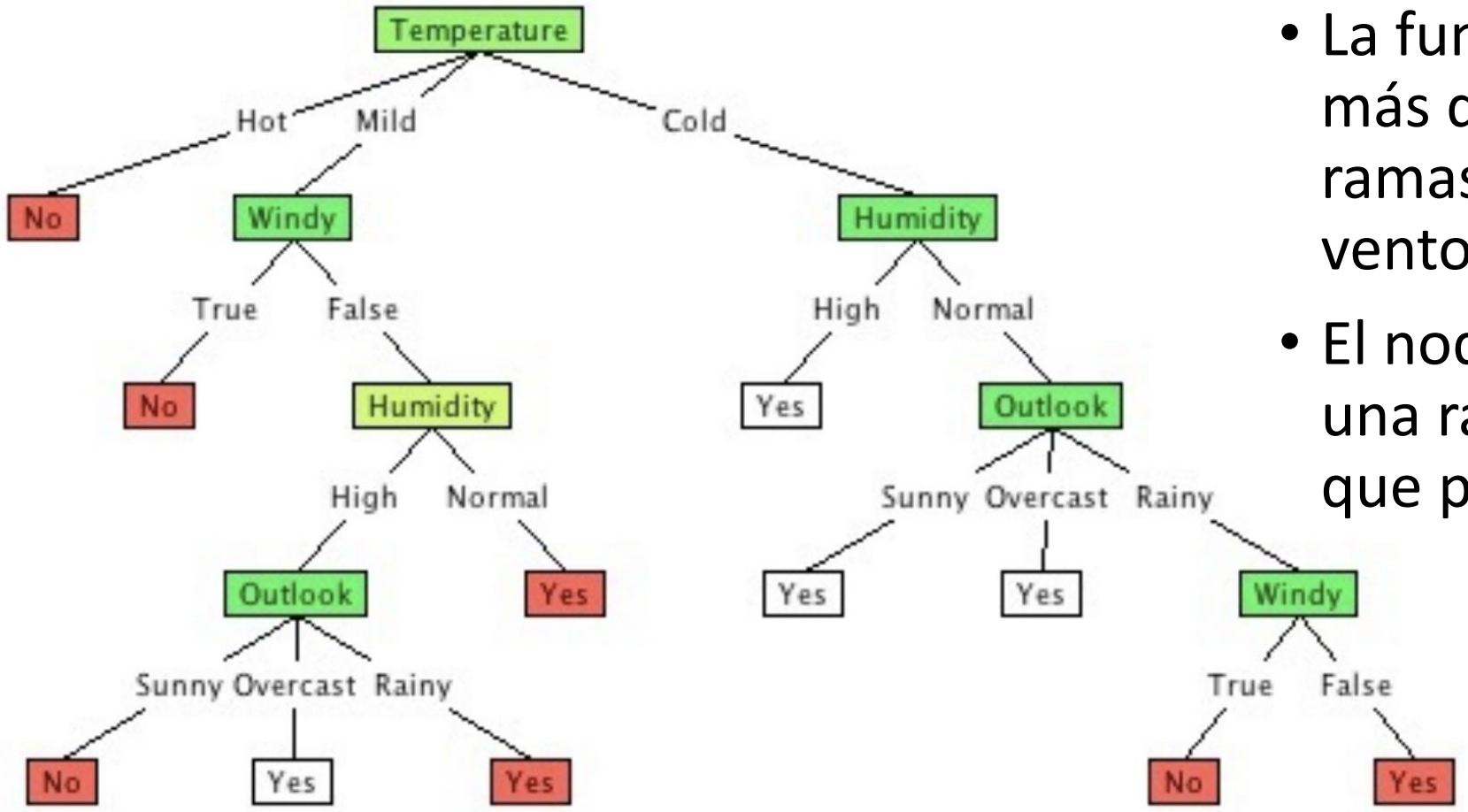
Árbol de Decisión

Example: Should We Play Tennis?

Play Tennis	Outlook	Temperature	Humidity	Windy
No	Sunny	Hot	High	No
No	Sunny	Hot	High	Yes
Yes	Overcast	Hot	High	No
Yes	Rainy	Mild	High	No
Yes	Rainy	Cold	Normal	No

- If temperature is not hot
  - Play
- If outlook is overcast
  - Play tennis
- Otherwise
  - Don't play tennis

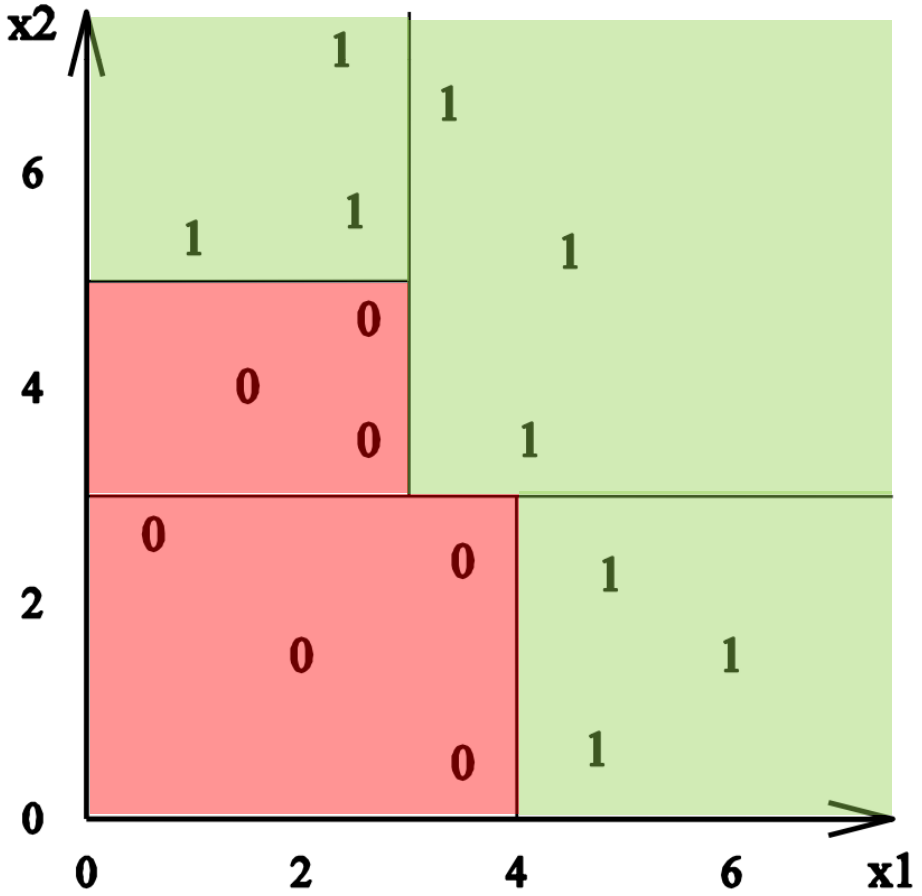
Árbol de Decisión

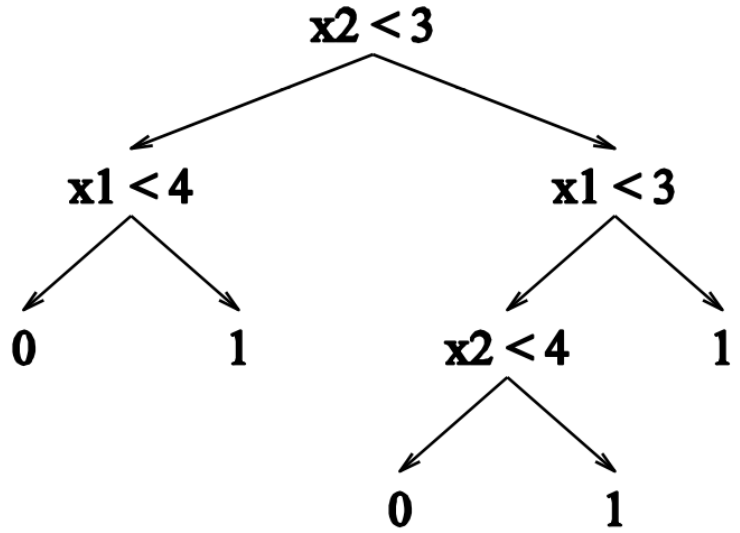


Propiedades

- La función puede aparecer más de una vez en diferentes ramas (por ejemplo, ventosa).
- El nodo puede tener tanto una rama como una hoja que proviene de él.

Árbol de decisiones: pros y contras

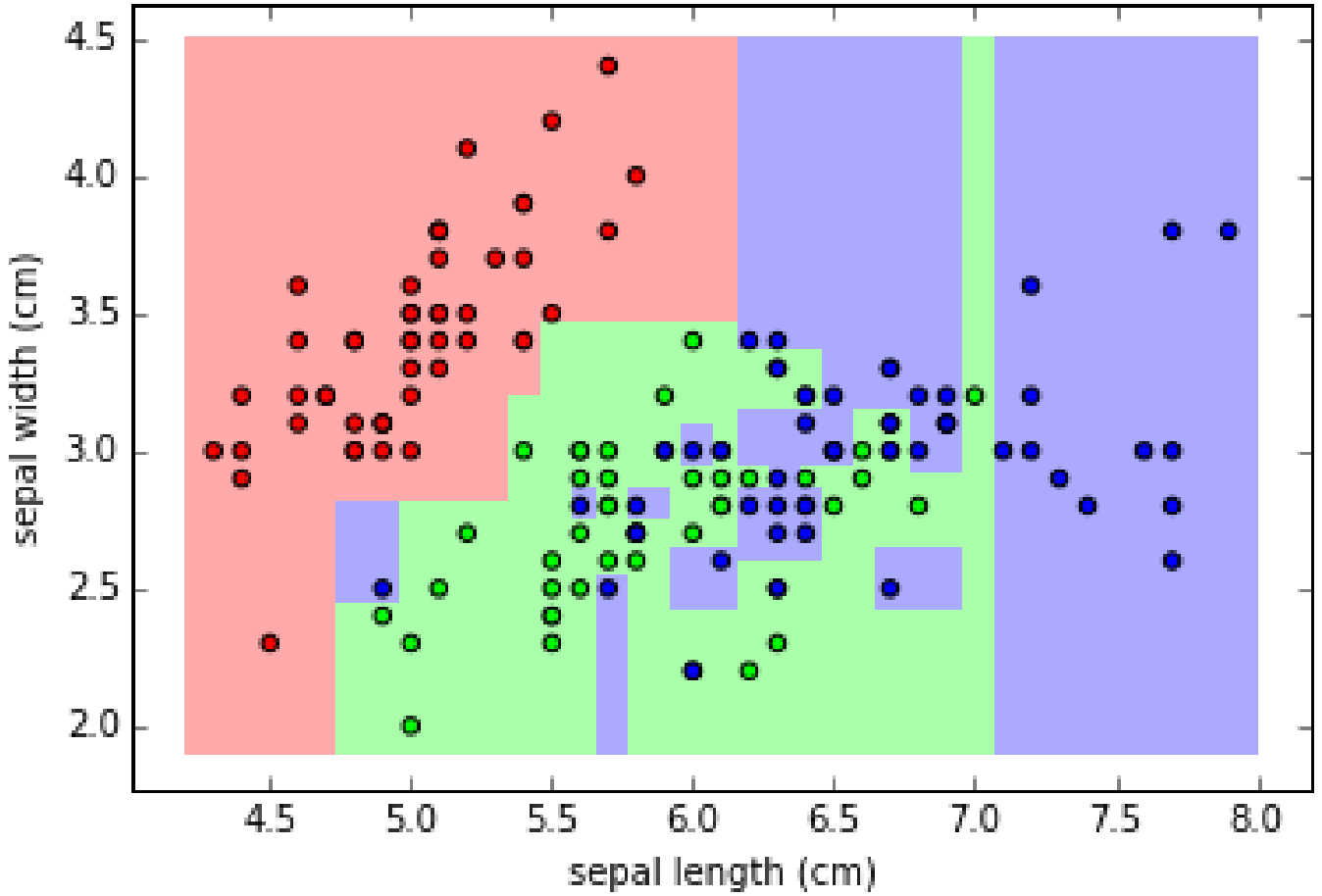




Ventajas:

- Límites de decisión no lineales
- Fácil de interpretar
- Datos numéricos y categóricos

Árbol de decisiones: pros y contras



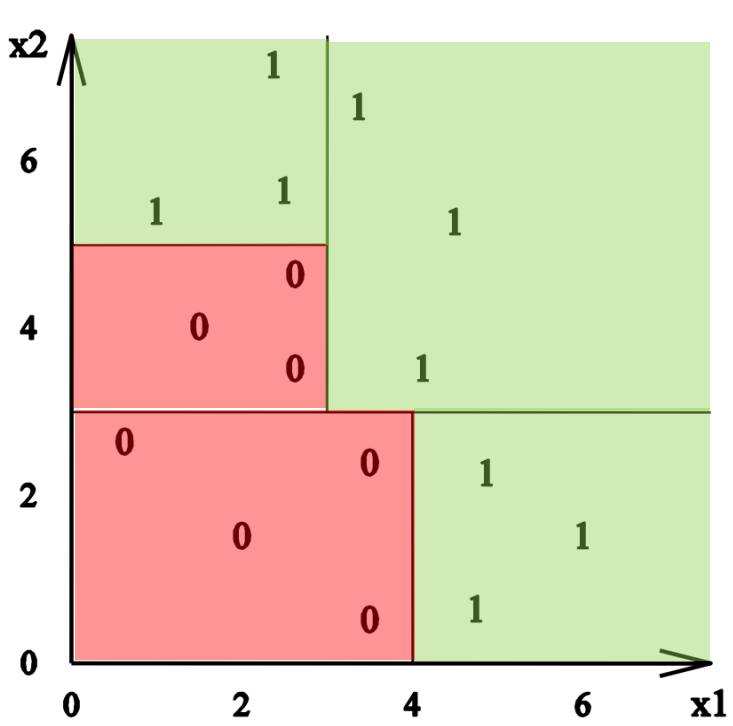
Desventajas:

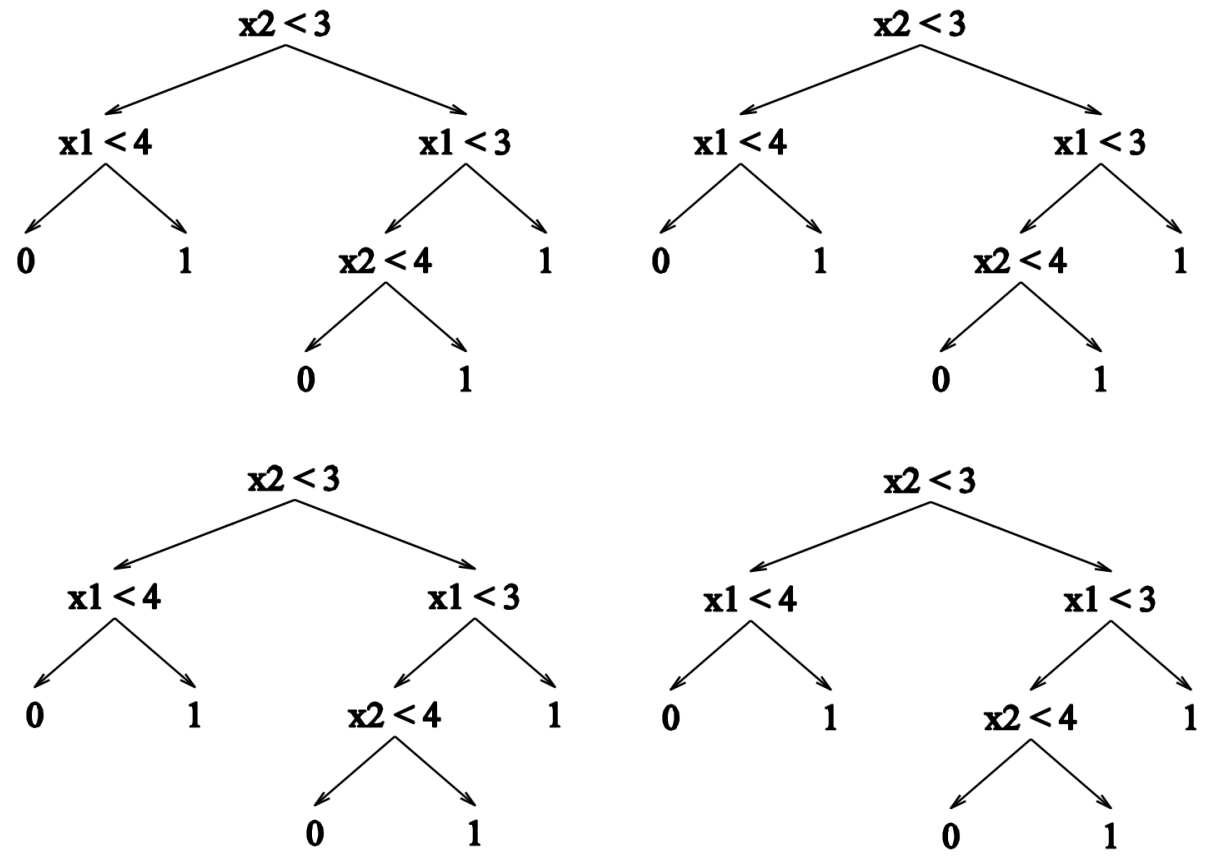
- Fácil de sobreajustar
- Alta varianza (es decir, inestable).
- Esconde la multicolinealidad y solo usa una pequeña parte de la variables (**solo los mejores jugadores**)

Bosques Aleatorios (Random Forest)

Random Forests – Muchos árboles de decisión

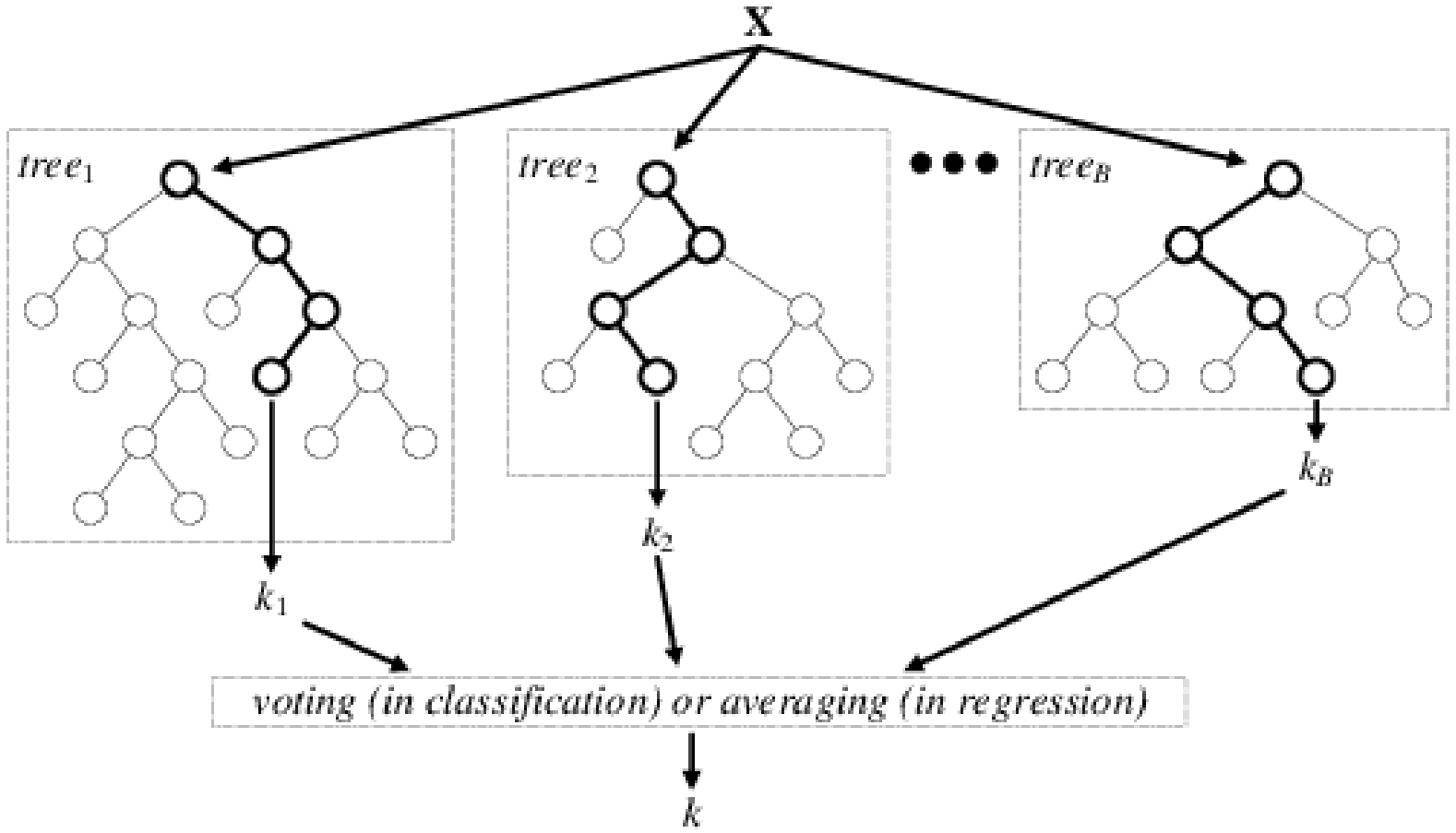
- Podemos adivinar que un Bosque Aleatorio = muchos árboles de decisión. ¿Pero cómo?
- Muchas copias de exactamente el mismo árbol son inútiles...





Random Forests – Muchos árboles de decisión

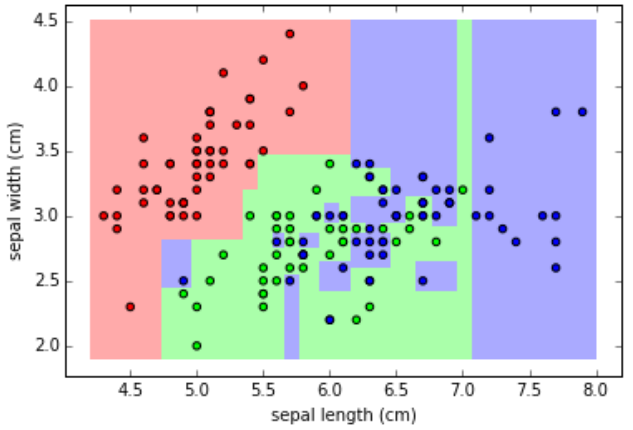
- Bien, entonces queremos alguna variación del árbol, pero ¿cómo...



Random Forests – Muchos árboles de decisión

- Queremos variación de árboles, pero ¿cómo...
- **Variando los árboles de manera que se reduzca la varianza general:**

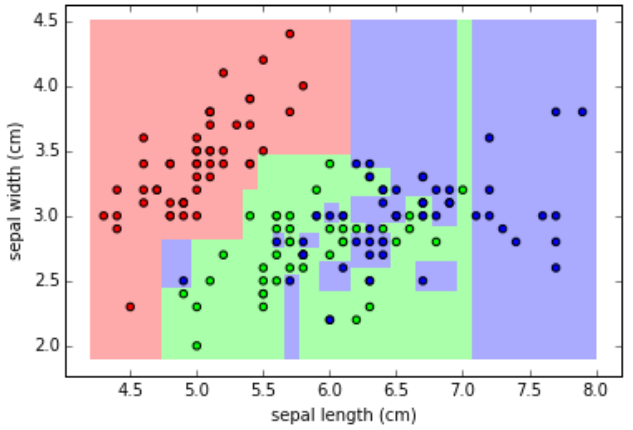
Remember: Decision Tree



Random Forests – Many Decision Trees

- Queremos variación de árboles, pero ¿cómo...
- **Varíe los árboles de manera que se reduzca la varianza general:**
- STATS101:

Remember: Decision Tree



Dado un conjunto de observaciones independientes y no correlacionadas $Z_1, Z_2, \ldots, Z_n$ cada uno con variación σ^2, la varianza de $\bar{Z}$ is $\frac{\sigma^2}{n}$.

Esta es la razón por la que un bosque de árboles idénticos es inútil.

Es por eso que ensamblar muchos modelos juntos siempre mejora los resultados.

¿Mediciones de la velocidad de la luz de Michelson están equivocadas?

- Albert A. Michelson, un físico estadounidense, fue conocido por realizar más de 100 mediciones meticulosas de la velocidad de la luz.
- A lo largo de varias décadas, comenzando en 1878, Michelson refinó sus experimentos utilizando aparatos de espejos giratorios. Su primera medición notable arrojó un resultado de **299,944 ± 51 km/s**, un valor muy cercano al aceptado hoy en día. Sin embargo, la velocidad real de la luz no se encuentra dentro de ese rango de medición.
- No obstante, actualmente sabemos, la velocidad de la luz en el vacío se define con exactitud como **299,792,458** m/s lo que está fuera del rango de Michelson.

Mediciones de la velocidad de la luz de Michelson

- No obstante, si construimos un intervalo de confianza moderno utilizando los mismos 100 experimentos, la velocidad real de la luz está dentro del rango.
- Si usamos las mismas 100 mediciones usando una estimación bootstrapping de la media tenemos:

299.853 ± 15 km/s (IC del 95%) (llegaríamos a mucho mejores resultados usando los mismos datos, velocidad de la luz: 299.792 km/s)

- Código Python por si queréis mirarlo: https://github.com/manoelgadi/Peppermoney/blob/main/08.Analyzing_speed_of_light_measurements.ipynb

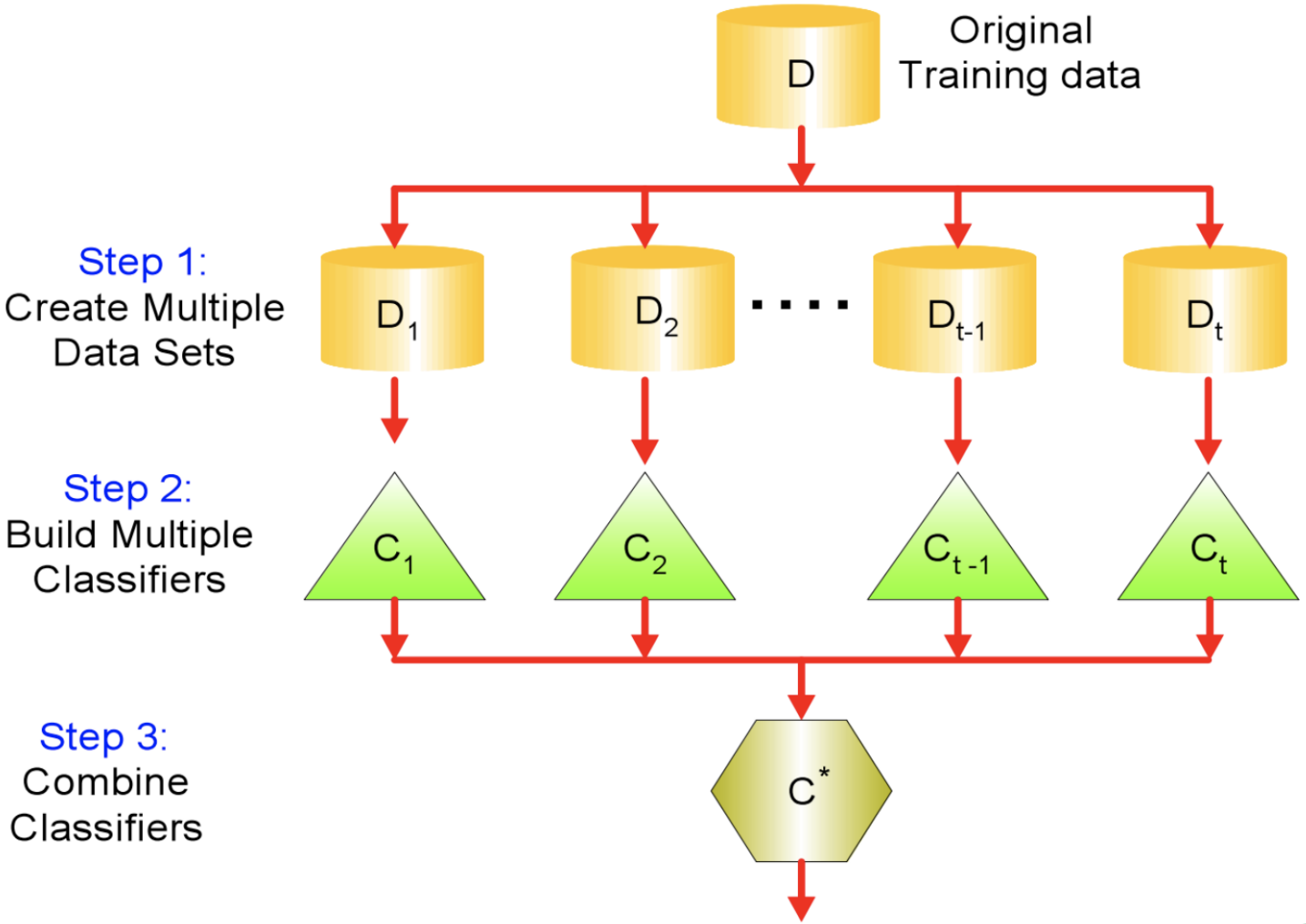
Random Forests – Muchos árboles de decisión

- ¿Cómo hacemos que los árboles sean lo más independientes y descorrelacionados posible?

Random Forests – Aleatorizar datos

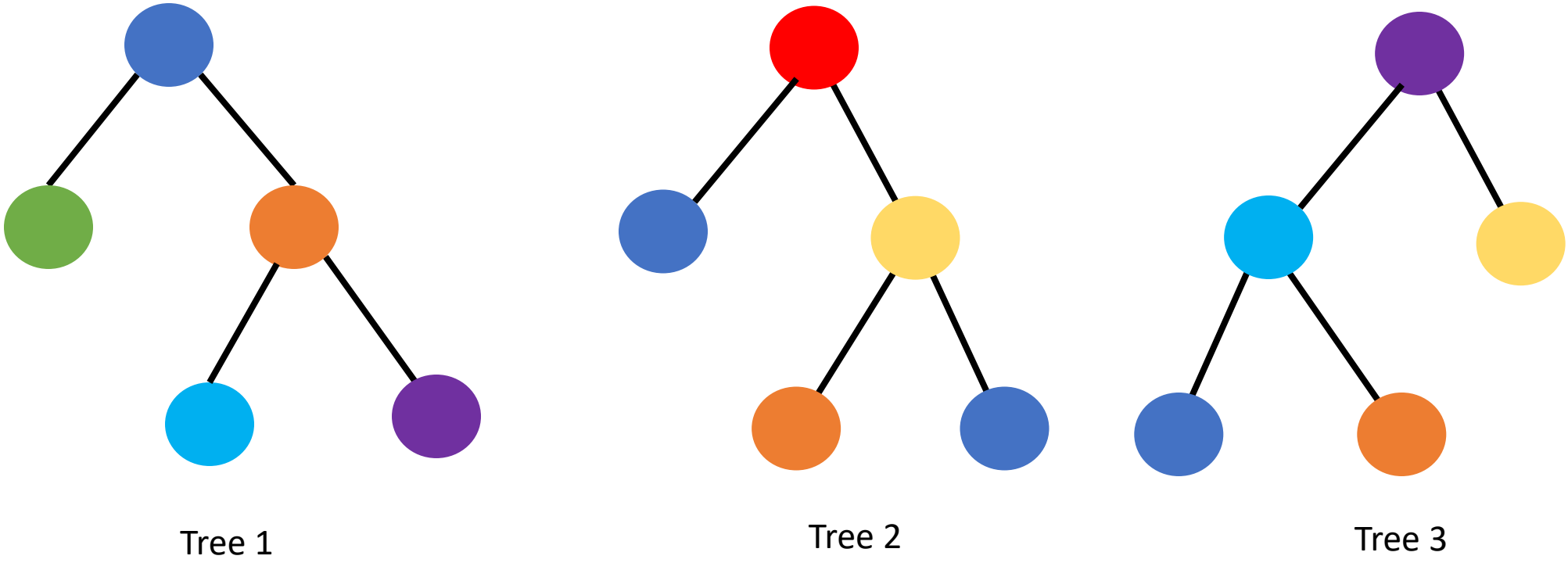
- **Bootstrap sampling**: Given set D containing N training examples, create D' by drawing N examples at random **with replacement** from D
- **Bagging**
  - Create k bootstrap samples $D_1, \ldots, D_k$
  - Train distinct classifier on each D_i
  - Classify new instance by majority vote / average

Random Forests – Randomize Data



[image from the slides of David Sontag]

Random Forests – Randomize Features



Tree 1 Tree 2 Tree 3

Feature 1 Feature 3 Feature 5 Feature 7

Feature 2 Feature 4 Feature 6

Random Forests – Intuition Check

- ¿Qué pasa si se asignan más/menos datos por árbol?

- ¿Qué sucede si selecciona más/menos del total de características por árbol?

Random Forests – Intuición

- ¿Qué pasa si se asignan más/menos datos por árbol?

Menos: los árboles están más descorrelacionados, más oportunidad para que jugadores un poco peores jugar en terrenos que jueguen bien, pero en algún momento muy pocos datos perjudican la formación, es decir en poco espacio y lluvia todos juegan mál.

Más: Los árboles se correlacionan más, pero el entrenamiento de cada árbol mejora, solo los buenos salen a jugar, pero son buenos (¿y si los buenos empiezan a fallar?).

- ¿Qué sucede si selecciona más/menos del total de características por árbol?

Menos: Los árboles están más descorrelacionados, pero en algún momento muchos árboles se vuelven "muertos", es decir, encajan árboles enteros en características sin importancia. Es decir, En algunos casos estas llevando solo el banquillo.

Más: Los árboles se correlacionan más, pero el entrenamiento de cada árbol mejora, solo los buenos salen a jugar, pero son buenos (¿y si los buenos empiezan a fallar?).

Optimización de árboles y Feature Importance

Optimización de árboles – Criterio codicioso (Greedy Criterion)

- Árboles cultivados de acuerdo a lo que es la mejor opción local.
- Criterio codicioso: Gini, Ganancia de Información.
- Criterio menos codicioso: Entropy

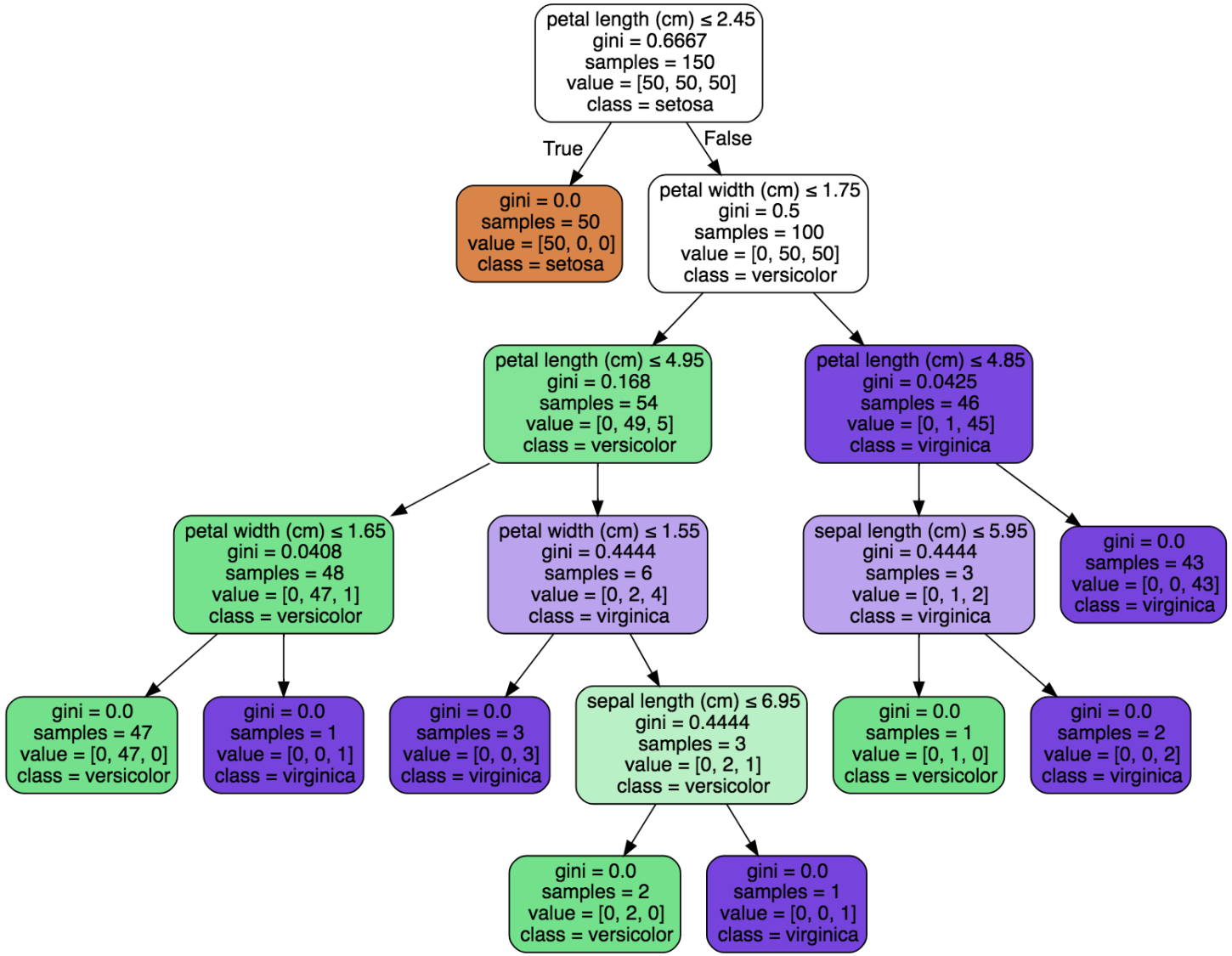
Feature Importance – Gini Impurity

"The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature. It is also known as the Gini importance."

$$\tilde{I}_f = \sum_{j=0}^{O_f} \frac{N_{n_j}}{T} \left(G_{n_j} - \frac{N_{LC_j}}{N_{n_j}} G_{LC_j} - \frac{N_{RC_j}}{N_{n_j}} G_{RC_j} \right), \quad I_f = \tilde{I}_f / \sum \tilde{I}_f$$

$I_f, \tilde{I}_f =$ Normalized/Unnormalized Importance of feature f
$O_f =$ Occurrences of feature f in tree
$N_{n_j} =$ number of samples in node n (for feature j)
$T =$ total number of samples
$G_{n_j} =$ Gini Impurity of node n (for feature j)
$N_{LC_j}, N_{RC_j} =$ number of samples in left/right child node (for feature j)
$G_{LC_j}, G_{RC_j} =$ Gini Impurity of left/right child node (for feature j)

Prunning del Árbol



- El parámetro de complejidad principal es max_depth y min_samples_leaf del árbol.
- Los árboles profundos y con menos datos pueden dividir más los datos, lo que lleva a un sobreajuste.
- ¡Algunos nodos aquí tienen una sola muestra!

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

n_estimators=100
Número de árboles en el bosque. Más árboles pueden mejorar el rendimiento pero aumentan el costo computacional.

criterion='gini'
Función para medir la calidad de una división. /'gini': usa la impureza de Gini. 'entropy': usa la ganancia de información.

max_depth=None
Profundidad máxima de los árboles. None significa que los nodos se expanden hasta que todas las hojas son puras o contienen menos de min_samples_split muestras.

min_samples_split=2
Número mínimo de muestras requeridas para dividir un nodo interno. Puede ser un número entero o una fracción (si es float, representa una proporción del total).

min_samples_leaf=1
Número mínimo de muestras requeridas para estar en una hoja. Ayuda a suavizar el modelo, especialmente en datos ruidosos.

min_weight_fraction_leaf=0.0
Fracción mínima del peso total (en caso de muestras ponderadas) que debe tener una hoja.

max_features='sqrt'
Número máximo de características consideradas para dividir un nodo: 'sqrt' (default): raíz cuadrada del número total de features (común en clasificación). 'log2', 'auto', o un número/floating también son opciones válidas.

max_leaf_nodes=None
Número máximo de nodos hoja. Si se especifica, el árbol crece con el mejor ajuste sin exceder este número.

min_impurity_decrease=0.0
Umbral mínimo de disminución de impureza. Una división solo se realiza si reduce la impureza al menos en este valor.

bootstrap=True
Si se deben usar muestras bootstrap (con reemplazo) al construir árboles. True: método tradicional de Random Forest. False: sin reemplazo.

oob_score=False
Si se debe usar el conjunto **fuera de la bolsa (out-of-bag)** para validar el modelo (útil cuando bootstrap=True).

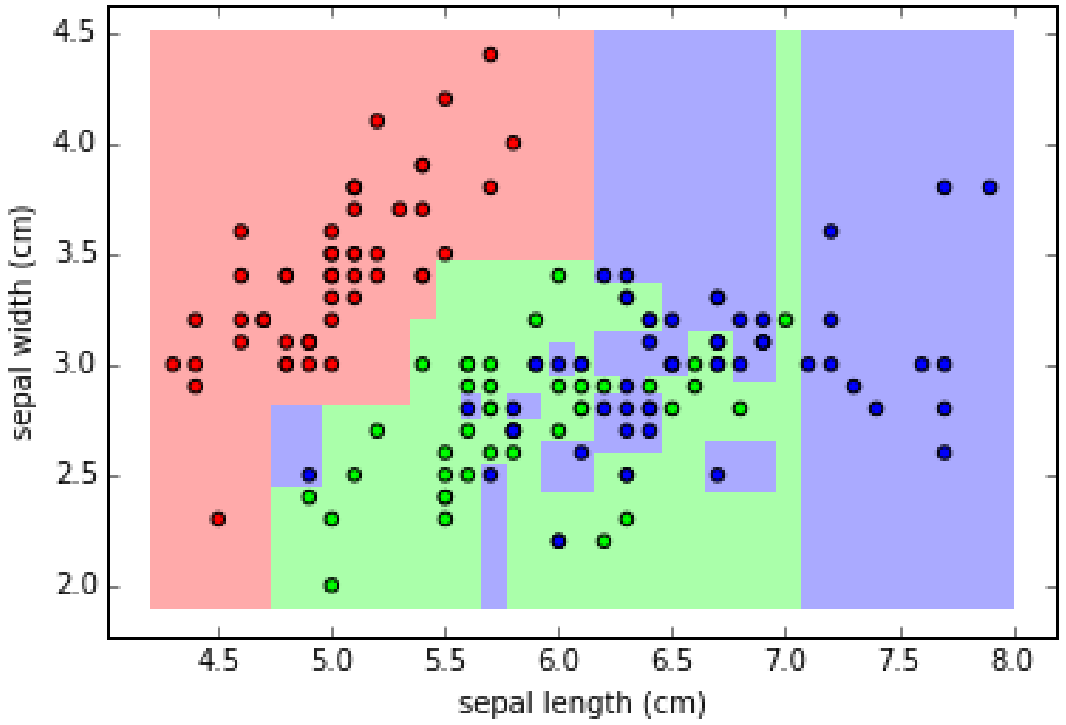
n_jobs=None
Número de trabajos (threads/cores) a usar en paralelo. None: se usa uno. -1: se usan todos los procesadores disponibles.

random_state=None
Semilla del generador aleatorio para reproducibilidad.

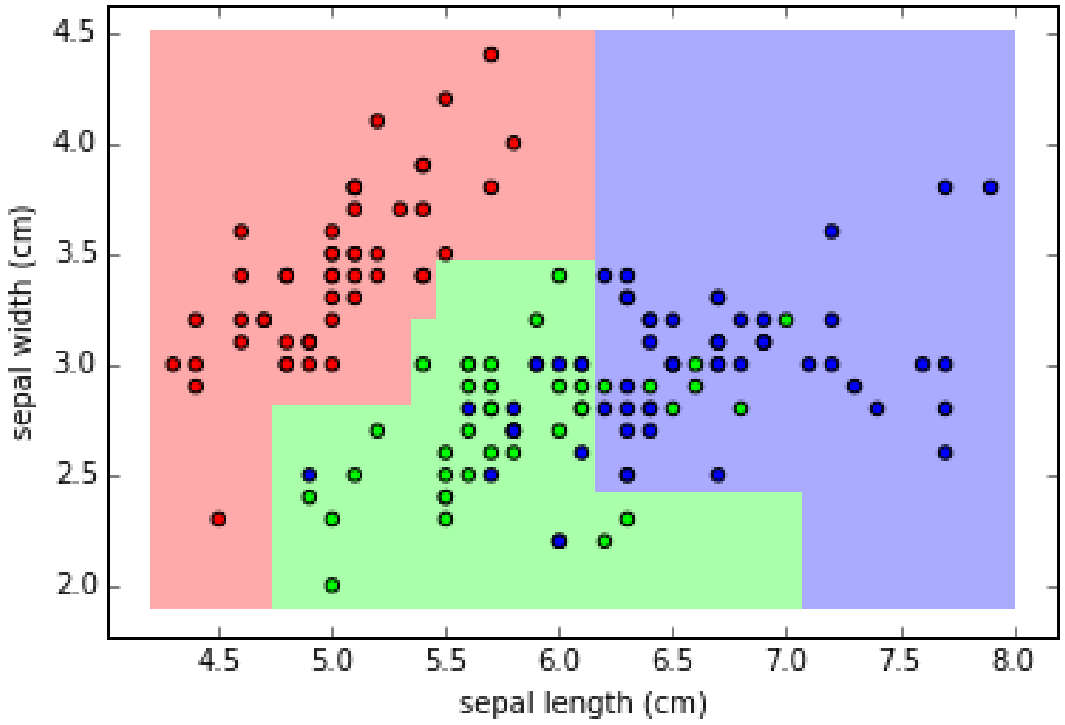
Regularización del Modelo

https://cs.stanford.edu/people/karpathy/svmjs/demo/demoforest.html

Deep, Unregularized Tree



Shallower, Regularized Tree



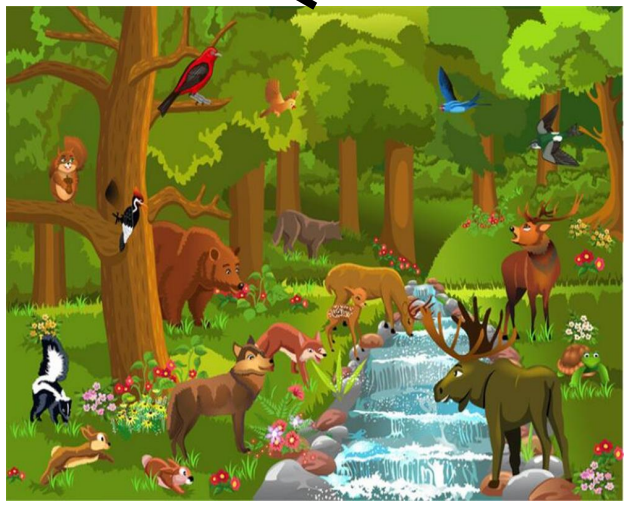
Notas Finales

Fin

Insight_Fellow_i

Miró el reloj/me quedé dormido

Prestó atención todo el tiempo



Links

- https://www2.isye.gatech.edu/~tzhao80/Lectures/Lecture_6.pdf
- http://scikit-learn.org/stable/modules/tree.html
- https://stackoverflow.com/questions/20224526/how-to-extract-the-decision-rules-from-scikit-learn-decision-tree
- http://www.utdallas.edu/~nrr150130/cs7301/2016fa/lects/Lecture_10_Ensemble.pdf
- http://www2.stat.duke.edu/~rcs46/lectures_2017/08-trees/08-tree-advanced.pdf
- https://stackoverflow.com/questions/49170296/scikit-learn-feature-importance-calculation-in-decision-trees
- https://github.com/scikit-learn/scikit-learn/blob/18cdaa69c14a5c84ab03fce4fb5dc6cd77619e35/sklearn/tree/_tree.pyx#L1056
- https://github.com/scikit-learn/scikit-learn/blob/master/sklearn/tree/_criterion.pyx
- https://medium.com/@srnghn/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3