Ayudantía 10

Random Forest, *Boosting* y Presentación T4

Por Juan José Alonso y Gonzalo Fuentes

25 de octubre del 2024

Contenidos

- *Random Forest*
- *Boosting*
- Presentación Tarea 4

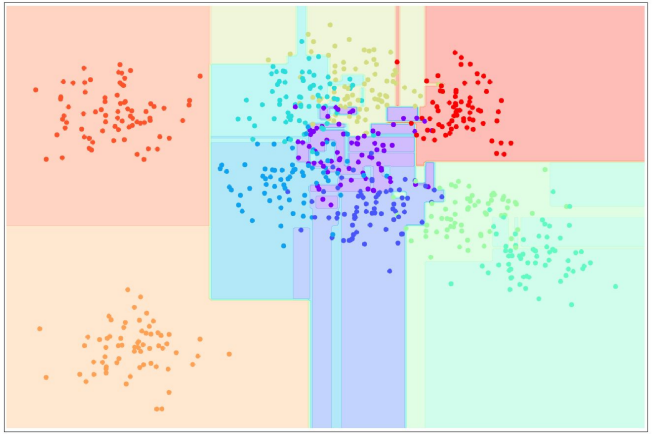
Recordemos...

- El **árbol de decisión** tiene una gran ventaja: es simple y fácil de interpretar...
- ... pero puede sufrir serios problemas de sobreajuste

Recordemos...

¿De dónde proviene el sobreajuste en un árbol de decisión?

- Al crear nuevos nodos en el entrenamiento ("bajar" en el árbol), la **muestra** se hace cada vez más reducida
- Si hay muchos **atributos**, es altamente probable elegir alguno bueno en los datos de entrenamiento, pero que es "inútil" para generalizar




MILK
CHOCOLATE
m&m's
CHOCOLATE CANDIES
PER SERVING
140
NET WT 62.0 OZ (3 LB 14.0 OZ) 1757.7g

*¿Cuántos **m&m's** hay en este tarro?*

*¿Cuántos **m&m's** hay en este tarro?*

*Hay **1124 m&m's** en este tarro*

*¿Qué tan **cerca** estuvo cada participante?*

Pedro dijo que habían **1258** = erró por **134**

María dice que hay **931** = erró por **193**

Juan dice que hay **1025** = erró por **99**

Josefina dice que hay **1594** = erró por **470**

Martín dice que hay **780** = erró por **344**

*¿Cómo podemos **mejorar** este resultado?*

*¿Y si calculamos el **promedio** entre los participantes?*

(1258 + 931 + 1025 + 1594 + 780) / 5 =

5588 / 5 =

1117.6 ≈

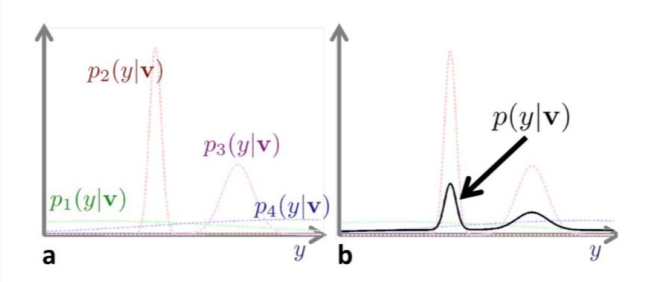
1118

Erraron por **6 *m&m's***

*¿Y si aplicamos el ejemplo de los m&m's a los **árboles de decisión**?*

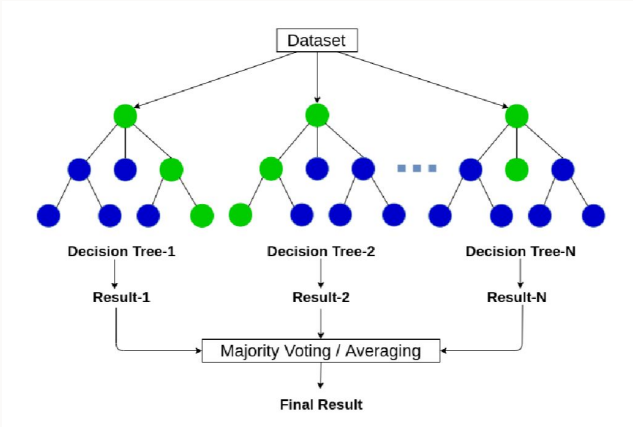
Ensambles de modelos con baja correlación

- Si el **patrón de error** (acierto) es **distinto** para todos, es altamente probable que, en promedio, la respuesta del ensamble sea correcta (los **errores se cancelan**)



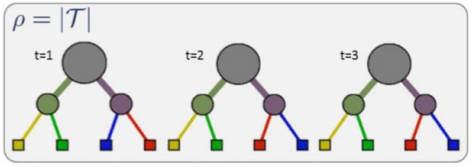
Muestras aleatorias

- Tomemos varios **árboles de decisión** y armemos una gran **“votación”**, como hicimos en el caso de los ***m&m’s***
- Como vimos en la diapositiva pasada, necesitamos que los modelos tengan **baja correlación** para que nuestro plan resulte. **¿Cómo logramos esto?**
- Si cada modelo ve sólo una parte (elegida aleatoriamente), es probable que la **correlación** entre ellos **disminuya**

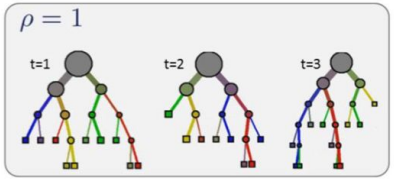


Atributos aleatorias

- Lo anterior no basta: **¿qué pasa si tenemos un atributo muy bueno?**
- La solución es similar: tomamos muestras aleatorias de atributos
- Esto no solo **disminuye la correlación**, sino que también **limita la profundidad del árbol**, reduciendo la complejidad de este



a) Low randomness, high tree correlation

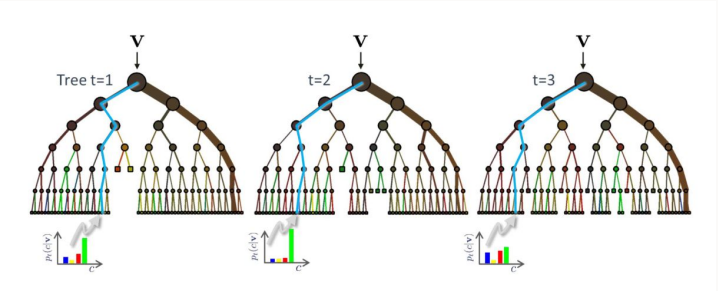


b) High randomness, low tree correlation

Random Forest

Random Forest

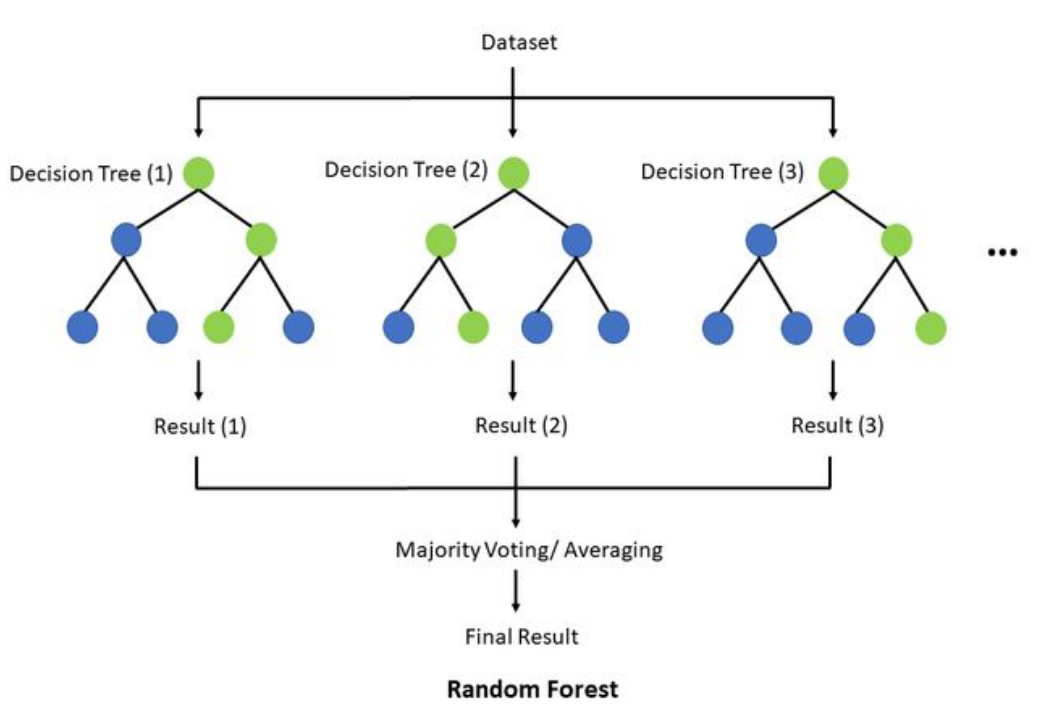
- La introducción de todas estas estrategias basadas en **aleatoriedad**, aplicada a **varios árboles de decisión**, convierten esta técnica en un ensamble llamado ***Random Forest***



Random Forest

- Un árbol para cada muestra aleatoria (subconjunto) de los datos.
- Árboles de **poca profundidad** para evitar el *overfitting*.
- La predicción final se obtiene de un promedio entre las predicciones de los árboles del bosque

Random Forest



Hiperparámetros: RF

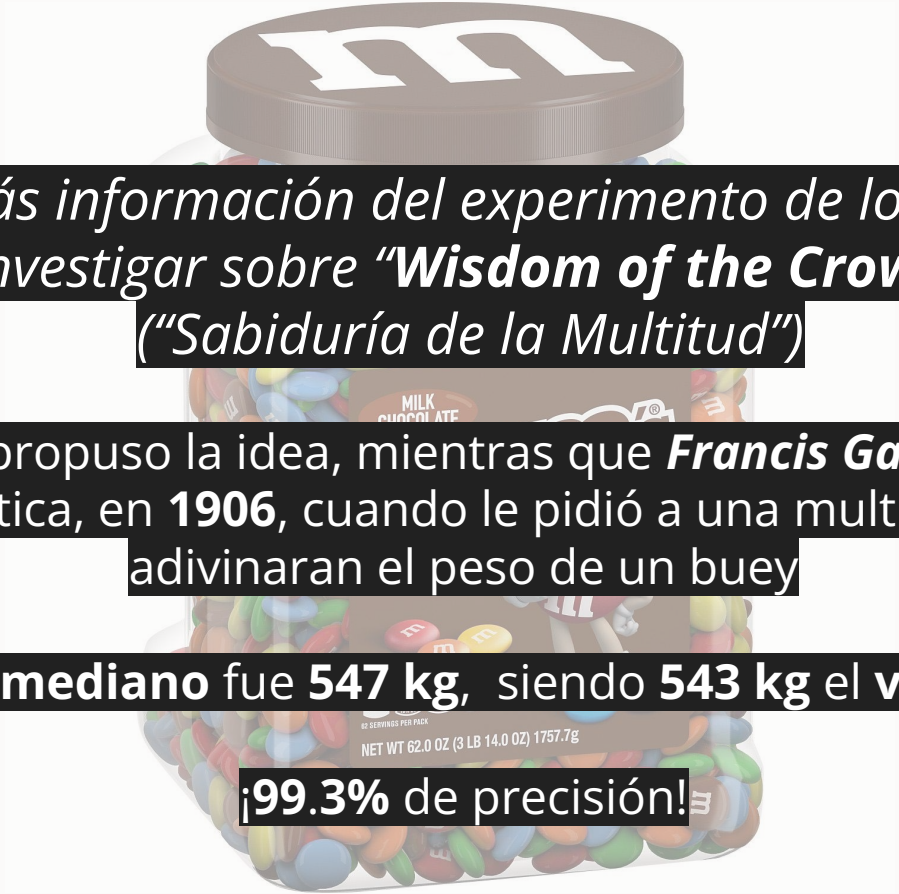
- **Los mismos de un árbol**
- **n_estimators:** número de árboles en el bosque

Ventajas de *Random Forest*

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

Ventajas de *Random Forest*

- Todas estas ventajas hacen que ***Random Forest*** sea un modelo **altamente utilizado**, a día de hoy, en el mundo de *machine learning*



*Para más información del experimento de los **m&m's**, investigar sobre "**Wisdom of the Crowd**" ("Sabiduría de la Multitud")*

Aristóteles propuso la idea, mientras que ***Francis Galton*** la llevó a la práctica, en **1906**, cuando le pidió a una multitud que adivinaran el peso de un buey

El valor **mediano** fue **547 kg**, siendo **543 kg** el **valor real**

¡**99.3%** de precisión!

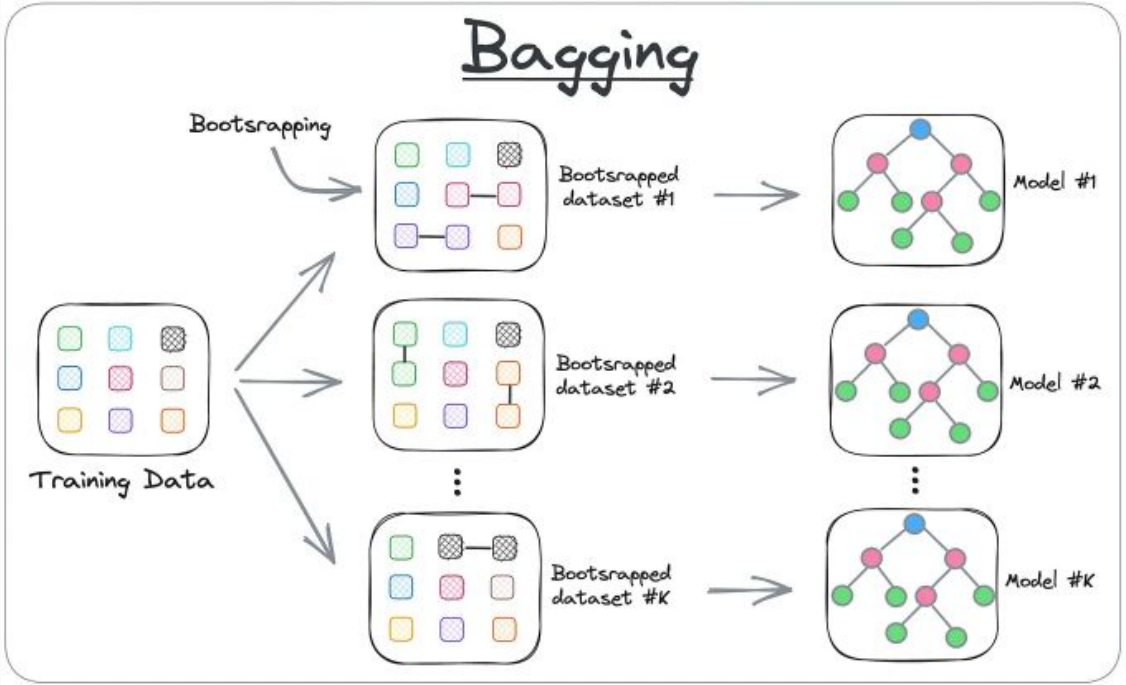
Ensambles

¿Qué son los ensambles?

- Combinación de múltiples modelos con el fin de **crear uno mejor.**

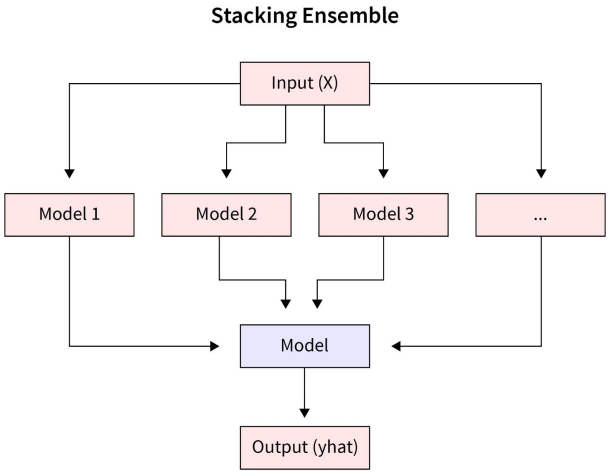
Tipos de ensambles : *Bagging*

- Crear **múltiples datasets** con muestreo **con reemplazo** y entrenar distintos modelos.
- Luego por alguna d**ecisión conjunta**, por ejemplo mayoria de votación se clasifica/predice.
- Random Forest es el clásico de estos modelos



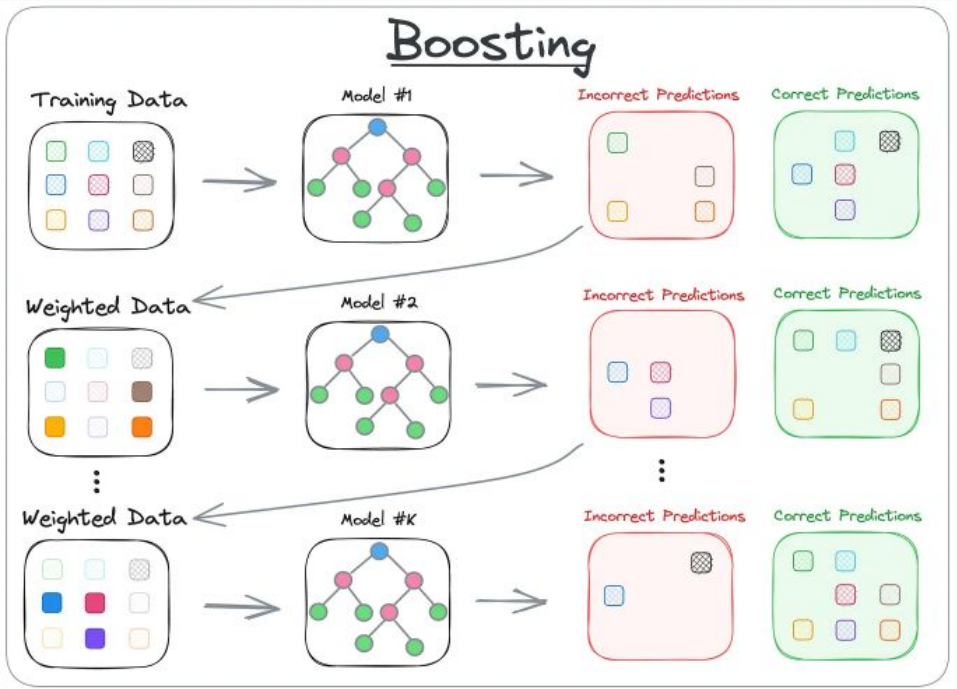
Tipos de ensambles : *Stacking*

- Entrenar varios modelos con el mismo dataset. Se recomienda que sean ***weak learners***.
- Luego ***stackear*** (concatenar) los *outputs* de los modelos y usarlos como *inputs* para entrenar un ***metalearner***.
- Finalmente este *metalearner* genera el *ouput* esperado.

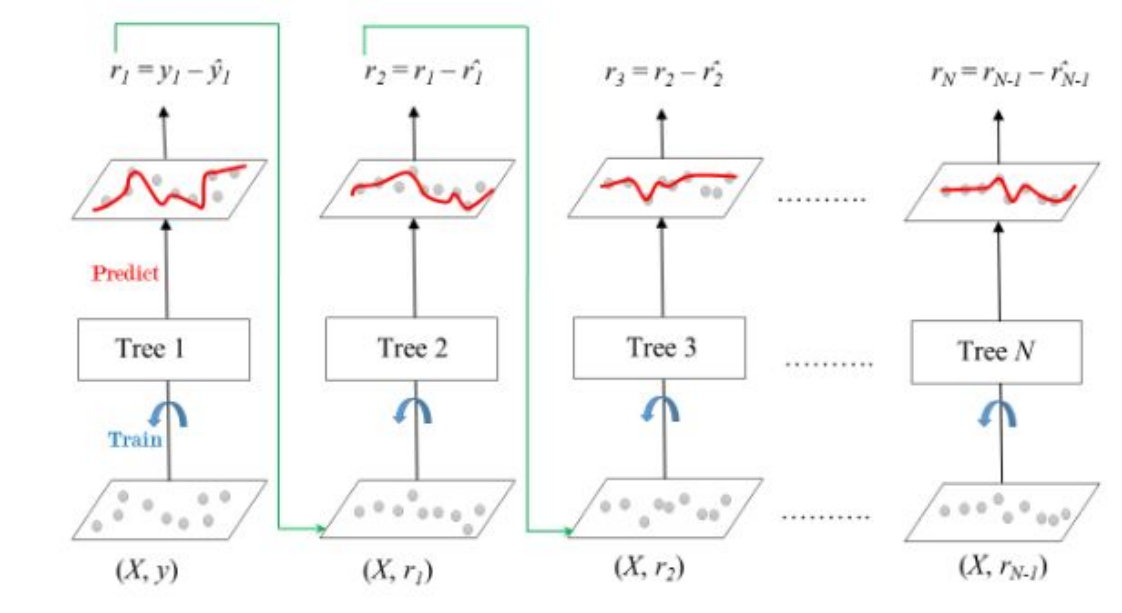


Tipos de ensambles : *Boosting*

- Aprovechar la **información errónea** de una jerarquía de modelos.
- De forma secuencial se entrenan modelos cada vez más débiles, para que estos **aprendan de los errores de los anteriores**.
- Luego se **combinan** para generar el output.



Gradient Boosting



Algorithm 2 Gradient Boosting

Input Data: $x_i \in R^n, y_i \in [-1, +1]$

1: **Initialize** $F_0(x) = argmin_\gamma \sum_{i=1}^{n} L(y_i, \gamma)$

2: **for** $m = 1$ to M **do**

3: Compute the negative gradient

$$r_{i,m} = -\left[\frac{\delta L(y_i, F(x_i))}{\delta F(x_i)}\right]$$

4: Fit tree to targets $r_{i,m}$, with leafs $R_{j,m}$

5: For $j = 1, 2, ...J_m$

$$\gamma_{jm} = argmin_\gamma \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x) + \gamma)$$

6: Update the estimation of F(x)

$$F_m(x) = F_{m-1}(x) + \nu(\gamma_{jm})$$

7: Output $\hat{f}(x) = f_M(x)$

Gradient Boosting : Clasificación

- Si los modelos son de **regresión** el error es simplemente la **diferencia** del valor predicho. ¿Pero qué pasa con los **problemas de clasificación**?

- No podemos usar la diferencia de clases como tal, puesto que no representan algo real*.

- Pero recordando los modelos de clasificación lo que hacen es calcular las **probabilidades de que pertenezcan a una clase**.

XGBoost : eXtreme Gradient Boosting

- Probablemente el algoritmo de boosting más utilizado. Le hace competencia a Deep Learning.

- *Gradient Boosting* **no escala bien para muchos datos.**

- **XGboost:** Es una versión más eficiente, puesto que:
  - Agrega regularizadores L1 y L2 (quitan el overfitting)
  - Paraleliza las tareas
  - Tiene implementado una forma de **parar** con basado en Cross-Validation
  - Corta las zonas menos interesantes de árboles.

Ayudantía 10

Random Forest, *Boosting* y Presentación T4

Por Juan José Alonso y Gonzalo Fuentes

25 de octubre del 2024