# Guía de estudio - Modelos de ensamble (Parte I)

 ¡Hola! Te damos la bienvenida a esta nueva guía de estudio.

## ¿En qué consiste esta guía?

La siguiente guía de estudio tiene como objetivo recordar y repasar los contenidos que hemos visto en clase.

Esta guía se focaliza en los modelos de ensambles paralelos de carácter homogéneos. Veremos nuestro primer modelo de ensamble llamado Bagging, en el que se combinan múltiples clasificadores fuertes y dan origen a un clasificador más potente. Veremos el esquema general de este modelo, revisando en detalle cada proceso y sus implicaciones, junto con sus hiper parámetros, principales bondades y problemas asociados a correlaciones entre sus estimadores.

Profundizaremos también en el modelo Random Forest, que puede ser aplicado en problemas de clasificación y regresión utilizando una mezcla de árboles de decisión con muestras bootstrap y un componente extra de aleatoriedad con la selección de atributos, lo que genera un modelo con alto desempeño. Se explicará como usar validación cruzada por medio de *out of bag* que se generan debido a las muestras bootstrap; a través de esto podremos medir el error en cada estimador.

El desarrollo de esta guía considera aspectos teóricos, que luego son expuestos en forma práctica con Python utilizando *sklearn*. Así lograremos trabajar en forma efectiva el detalle de los modelos de ensambles paralelos y aplicarlos a diferentes situaciones y/o proyectos de Machine Learning.

¡Aquí vamos con los modelos de ensambles paralelos homogéneos! ¡No te duermas que esto se pone más interesante cada vez!

**¡Vamos con todo!**

# Tabla de contenidos



**¡Comencemos!**

# Introducción a los ensambles Bagging

Los métodos de ensambles son aquellos que se construyen mediante la combinación de otros modelos. La idea es mejorar la capacidad de generalización de los modelos individuales por separado por medio de esta combinación. Una analogía pueden ser los diagnósticos médicos mediante una junta médica, que permite contar con opiniones de diversos profesionales de la salud con especializaciones diferentes, y combinando estas de forma adecuada obtener un pronóstico que sea más preciso.

## Descripción general de ensambles

Los ensambles implementan lo que se denomina votación por mayoría o votación por pluralidad, para ello toman como input los datos de entrenamiento que son sometidos a múltiples modelos ($C_i$), en que cada uno entrega su estimación ($\hat{y}_i$) y luego a través de votación por mayoría se determina la clasificación para cada observación del conjunto de entrenamiento.

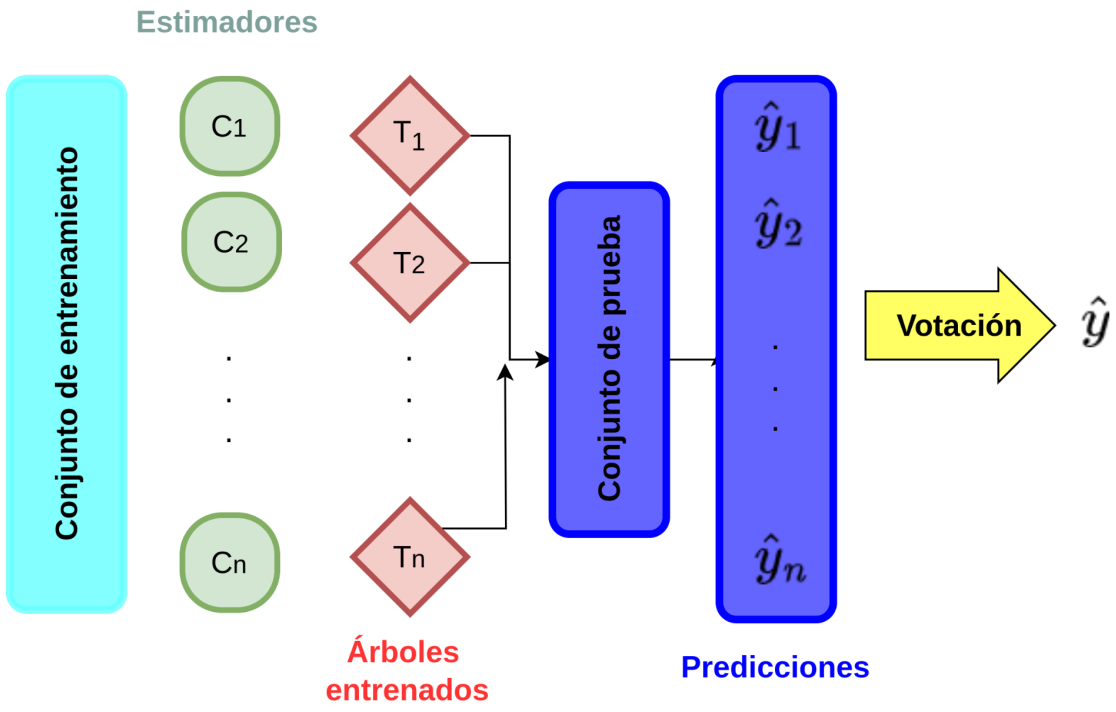


Figura 1. Combinaciones de modelos paralelos
Fuente: elaboración propia.

Este enfoque comprende trabajar con clasificadores independientes, lo cual posibilita el procesamiento separado. Esta característica es muy importante para las actuales arquitecturas multiprocesador y en especial las GPUs que cuentan con cientos y miles de procesadores; por esta razón a este tipo de modelos de ensambles se les conoce como

**paralelos**. También es importante destacar que no hay restricción en cuanto al algoritmos a usar para cada clasificador ($C_i$) del ensamble.

## Proceso modelos Bagging

Bagging es la implementación base de los ensambles paralelos, en el que cada clasificador ($C_i$) que compone el ensamble es entrenado con una muestra bootstrap obtenida de la muestra original, habitualmente del mismo tamaño. Este tipo de muestra corresponde a una muestra aleatoria con reemplazo, de esta forma cada clasificador operará sobre un conjunto de datos diferentes.

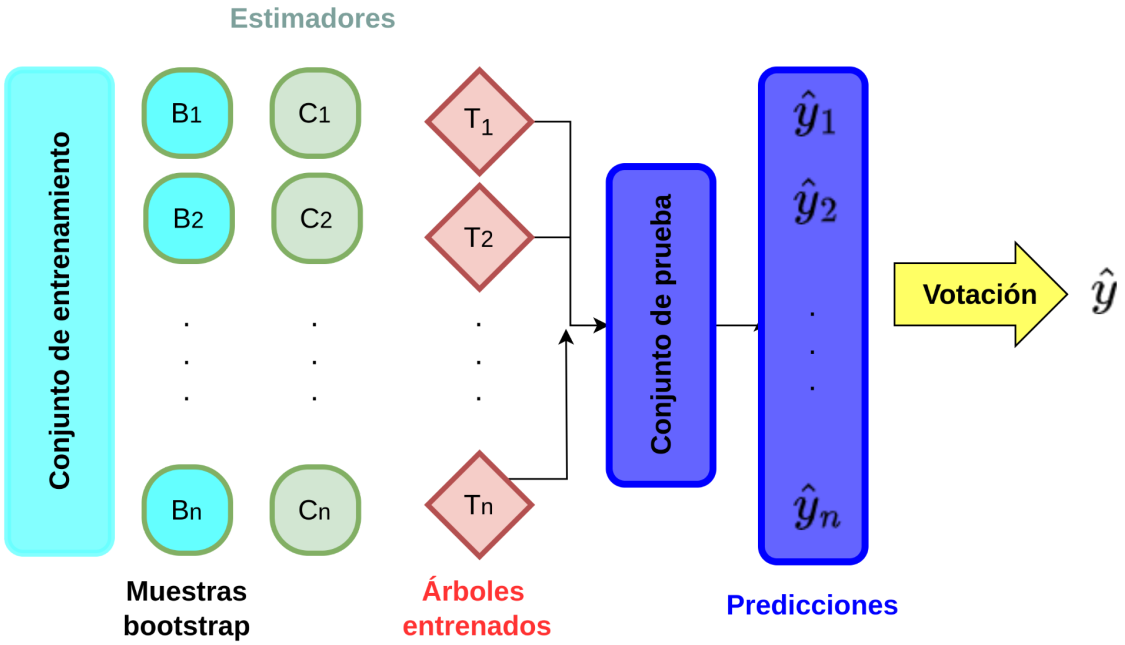


Figura 2. Combinaciones de modelos paralelos, enfoque Bagging
Fuente: elaboración propia.

Bagging presenta un mejor desempeño cuando los modelos que utiliza generan problemas de alta varianza; cuando los modelos que componen Bagging son árboles de decisión se obtienen mejores resultados.

Debido al muestreo bootstrap en Bagging se produce un beneficio ya que en cada muestreo se generan observaciones repetidas, por lo que quedará en aproximadamente un 33% de los datos de entrenamiento que no serán usados en el proceso. Esta bolsa de registros se conoce como ***out-of-bag-sample*** (OOB) y se emplea para la estimación del error, con resultados similares a usar k-fold validación cruzada pero sin tener que separar explícitamente la muestra. Además, para cada predictor $C_i$ se escoge una muestra bootstrap y los ejemplos que no se usen se emplearán para medir el error en cada predictor; luego para medir el error global del modelo Bagging se promedian los errores ***OOB*** de cada predictor, obteniendo una buena estimación del error del modelo de ensambles como si fuese un conjunto separado donde se calcula. Usar esta bolsa (OOB) nos ahorra tiempo de cómputo.

## Actividad guiada: Entrenando un modelo de ensamble paralelo Bagging con árboles de decisión

Trabajaremos sobre una base de datos de primas de seguros, con el objetivo de comparar los resultados de ambos modelos. Para esto, observa el archivo **02 - Bagging.**

# Ensambles Random Forest

Los ensambles Bagging, como se describió anteriormente, presentan problemas de correlación entre sus estimadores o también llamados modelos fuertes. Para solucionar esto aparece Random Forest, un modelo que sigue en general la misma estructura que un modelo Bagging pero agregando un componente extra de aleatoriedad mejorando o impulsando la idea de que los ensambles entregarán mejores resultados cuando existe mayor diversidad. Esto se puede lograr de distintas formas, por ejemplo, inyectando diversidad en los datos al usar diferentes conjuntos de entrenamiento para los clasificadores o aplicando estimadores heterogéneos, entre otros. En el caso de Bagging se genera diversidad al escoger los conjuntos de entrenamiento en forma aleatoria por medio de muestras bootstrap, pero como vimos aún tenemos correlaciones altas entre los clasificadores.

## Descripción modelo Random Forest

Random Forest usa la componente aleatoria de muestras bootstrap de Bagging, y además incorpora elección aleatoria de $K$ atributos (para K < S cantidad de atributos) a ser seleccionados en cada nodo de los árboles de decisión que componen el ensamble. El esquema de proceso Random Forest se puede ver en la Figura 5.

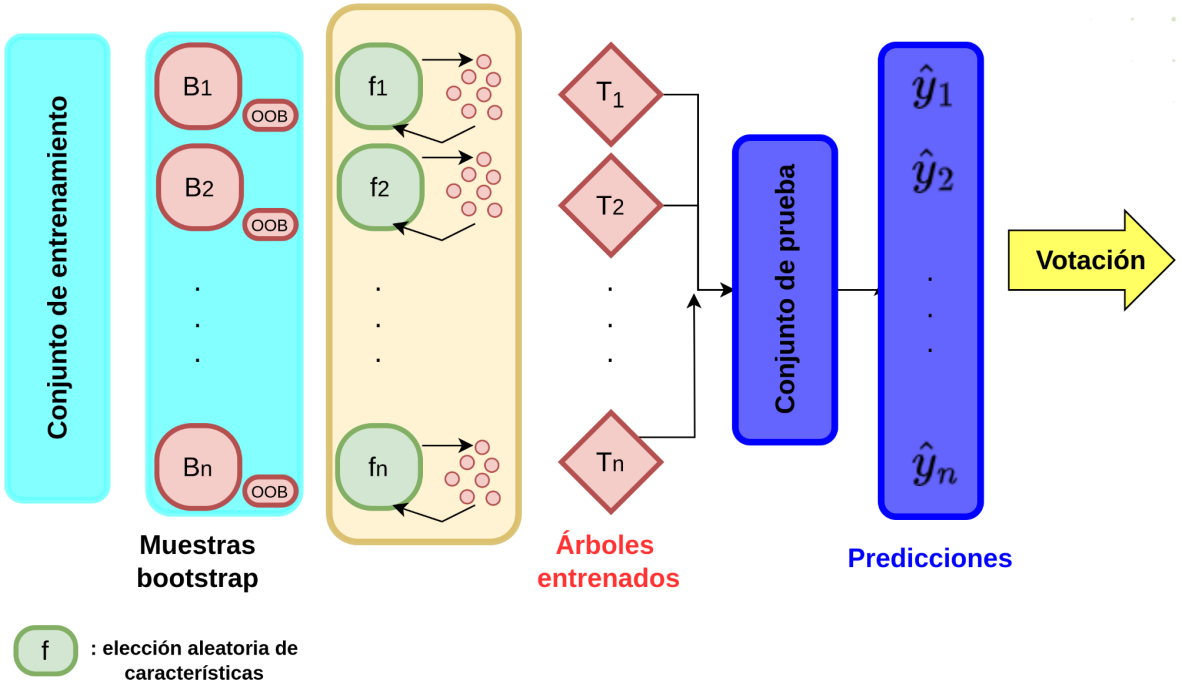


Figura 5. Esquema de funcionamiento Random Forest
Fuente: elaboración propia.

## Out-Of-Bag (OOB)

Ya sea con Random Forest como en los modelos Bagging, se genera una bolsa de observaciones no seleccionadas en cada proceso de muestreo bootstrap para cada clasificador que es parte del ensamble. Estos datos OOB se utilizan para estimar el error cometido en cada clasificador en el momento de ir entrenando los árboles de decisión; así, el error estimado del ensamble se obtiene promediando todos los errores de cada clasificador para estas observaciones OOB.

## Actividad guiada: Entrenando un modelo de ensamble paralelo Random Forest

En esta actividad vamos a implementar un modelo Random Forest para un problema de clasificación, aplicado a datos médicos de personas sanas y enfermas con diagnóstico de diabetes. Los atributos registrados son: género, edad, si sufre o no de hipertensión, si posee o no alguna enfermedad al corazón, nivel de fumador, índice de masa corporal, nivel de Hba1c y el nivel de glucosa en la sangre. Para esto, utiliza el archivo **02 - RandomForest**

## Preguntas de proceso

**Reflexiona:**

- ¿Qué ventaja computacional se puede aprovechar con modelos de ensamble paralelos?
- ¿Incluir diversidad en los modelos de ensamble en que ayuda ?
- ¿A qué se debe que el desempeño de Random Forest supere a Bagging?
- ¿Usar datos OOB en qué ayuda vs a conjuntos de entrenamiento y test?
- ¿Cual es la principal razón por la cual usamos árboles binarios y no n-arios?
- ¿A qué se debe que el desempeño de un modelo de ensamble paralelo se vea beneficiado por la cantidad de estimadores ?
- ¿En qué afecta que los clasificadores que componen un modelo de ensamble muestren alta correlación entre ellos?
- ¿Podemos usar modelos de ensamble para problemas de entrenamiento no supervisado?, ¿Cómo?
- Detalle el motivo por el cual Random Forest usa como clasificadores base a los árboles de decisión

## Referencias bibliográficas

BREIMAN, L (2001a). Random forest. Machine Learning 45, 5-32
BREIMAN, L (1996). Bagging predictors. Machine Learning 24, 123-140

**¡Continúa aprendiendo y practicando!**