



Pontificia Universidad
JAVERIANA
Bogotá

Métodos de Ensamble

Ing. Heriberto Felizzola Msc.

Educación **Continua**

Generamos experiencias educativas

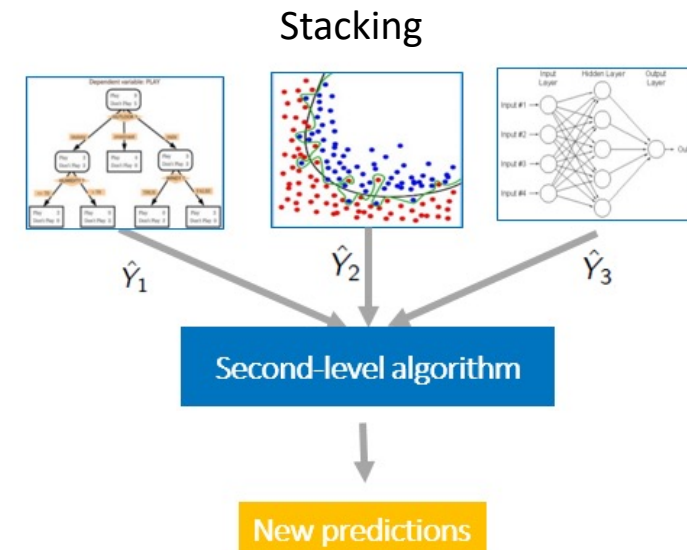
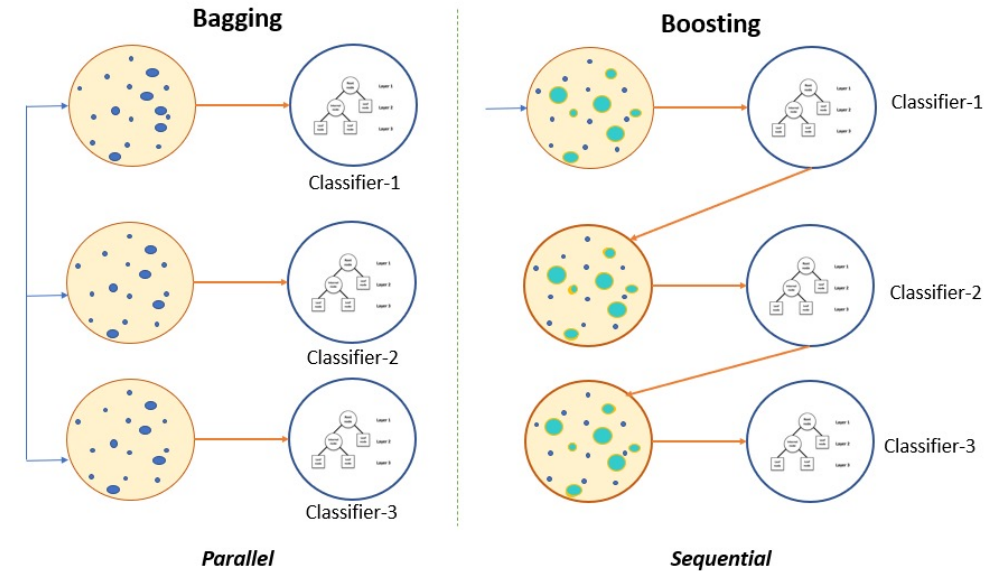
- Suponga que plantea una pregunta compleja a miles de personas al azar y luego agrega sus respuestas.
- En muchos casos encontrará que esta respuesta agregada es mejor que la respuesta de un experto.
- A esto se le llama la sabiduría de la multitud.
- De manera similar, si agrega las predicciones de un grupo de predictores (como clasificadores o regresores), a menudo obtendrá mejores predicciones que con el mejor predictor individual.
- Un grupo de predictores se llama **Ensamblés**; por lo tanto, un algoritmo de aprendizaje en conjunto se denomina **Método de Ensamble**.

Métodos de Ensamble



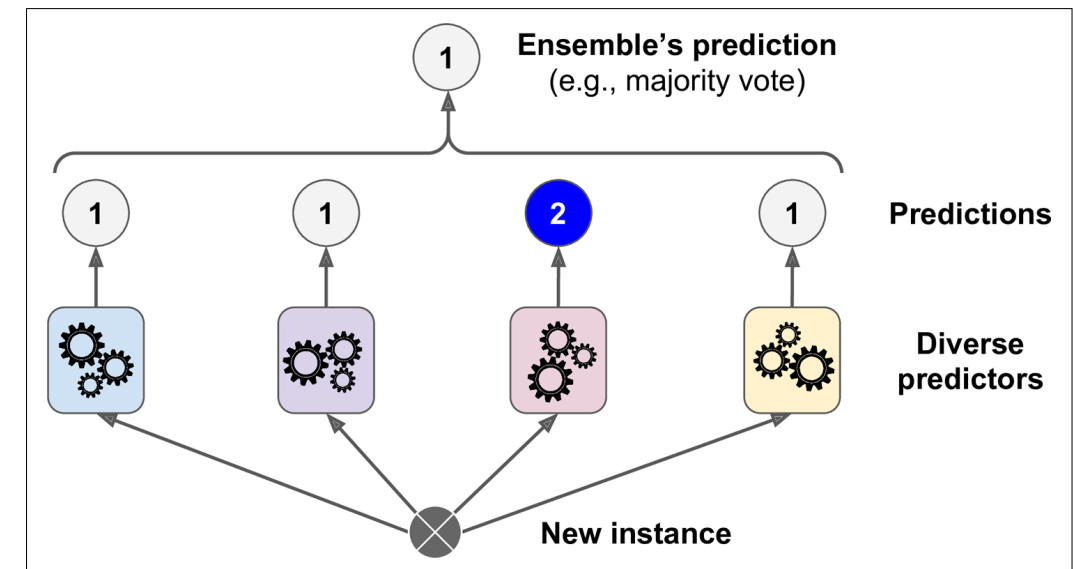
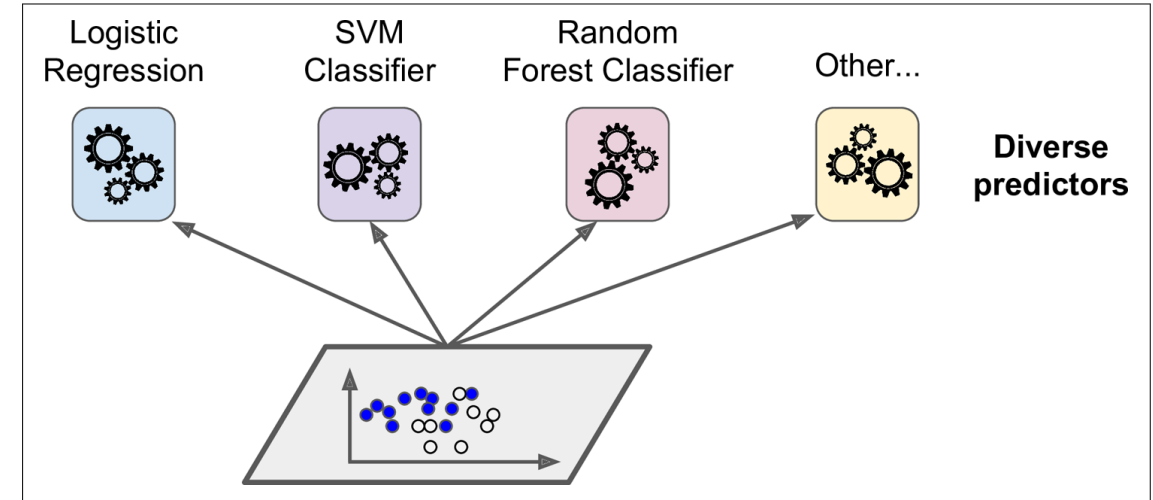
Pontificia Universidad
JAVERIANA
Bogotá

- Un método de ensamble es un enfoque que combina muchos modelos simples.
- La idea es tener "bloques de construcción" para obtener un modelo único y potencialmente muy poderoso.
- Estos modelos de bloques de construcción simples a veces se conocen como aprendices débiles, ya que pueden conducir a predicciones mediocres por sí mismos.
- Estos son métodos de ensamble para los cuales el bloque de construcción puede ser una simple regresión o un árbol de clasificación.
- Los enfoques de ensamble mas utilizados son:
 - Votación simple
 - Bagging (Bootstrap Aggregation)
 - Boosting
 - Stacking

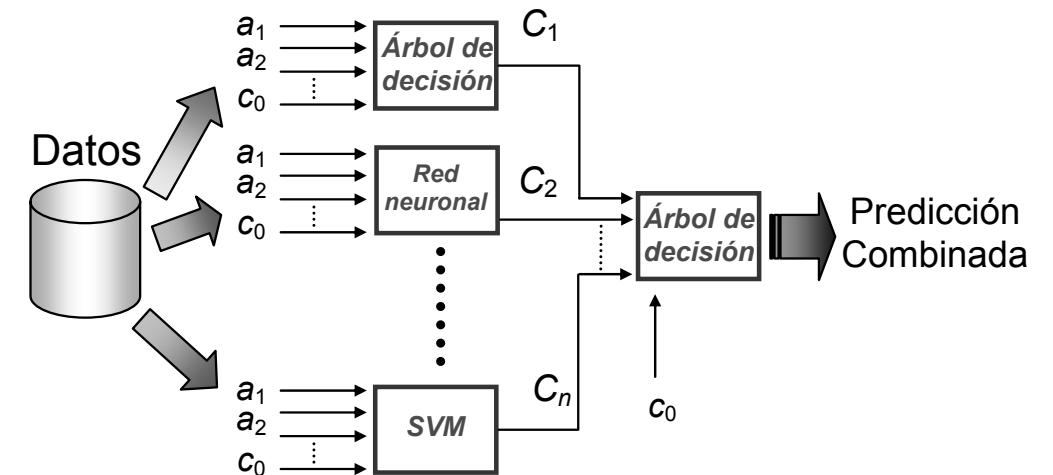


Métodos de votación simple

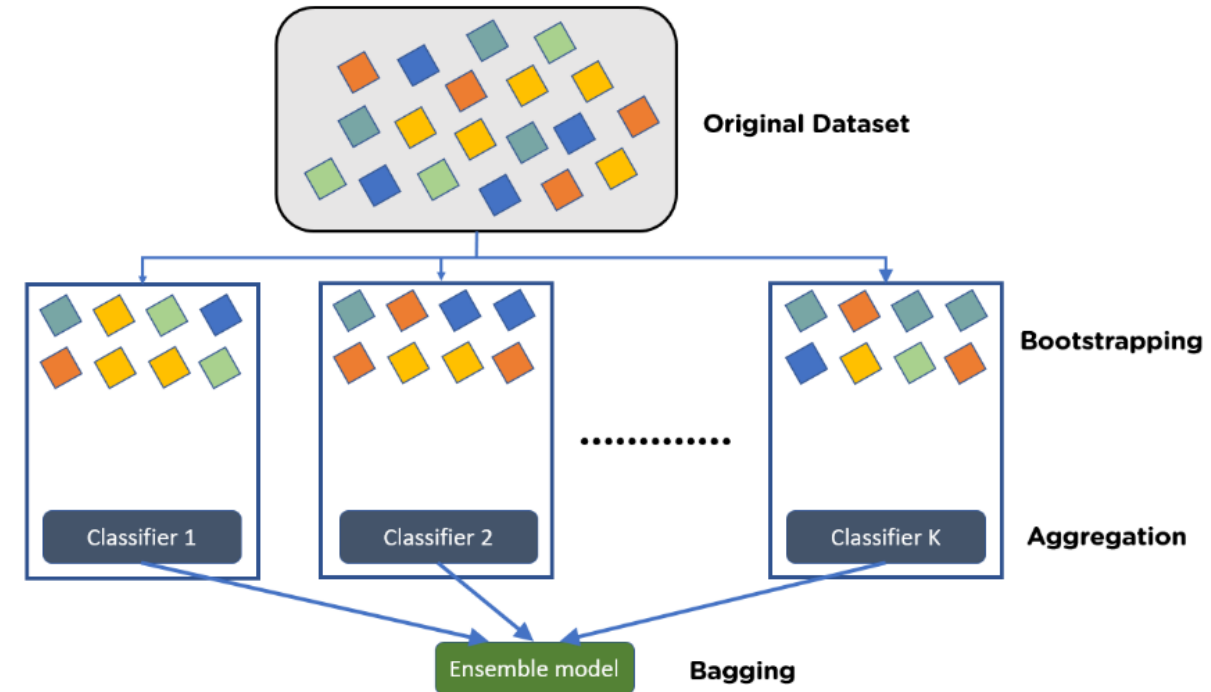
- Suponga que ha entrenado algunos clasificadores y cada uno logra aproximadamente un 80% de precisión.
- Es posible que tenga un clasificador de regresión logística, otro de SVM, un árbol de decisión, un clasificador de K vecinos más cercanos y quizás algunos más.
- Una forma muy sencilla de crear un clasificador aún mejor es agregar las predicciones de cada clasificador y predecir la clase que obtenga la mayor cantidad de votos.
- Este clasificador de voto mayoritario se llama clasificador de votación dura.
- Sorprendentemente, este clasificador de votación a menudo logra una mayor precisión que el mejor clasificador del conjunto.



- Es un método multclasificador bastante simple basado en la combinación de modelos generados por diferentes algoritmos de aprendizaje.
- Dado que cada modelo se aprende con un mecanismo de aprendizaje distinto se consigue que los modelos del conjunto sean diferentes.
- Una primera y simple versión del *stacking* consiste en aprender un modelo con uno de los algoritmos que deseemos utilizar.
- Si, por ejemplo, queremos utilizar árboles de decisión, regresión logística y redes neuronales, generaríamos un modelo con cada técnica y para un mismo conjunto de datos.
- La clasificación se podría realizar mediante una votación mayoritaria.

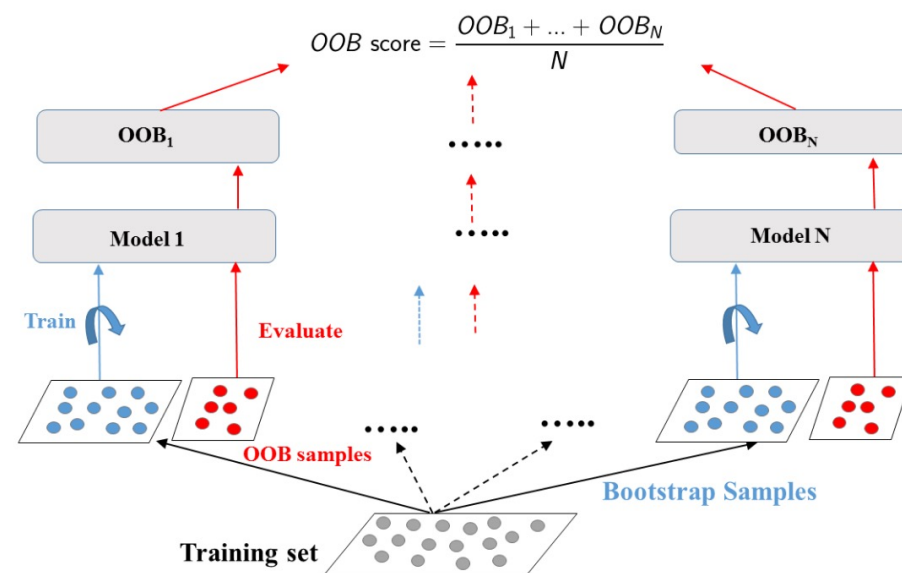


- Una forma de obtener un conjunto diverso de modelos o clasificadores es entrenarlos en diferentes subconjuntos aleatorios.
- Cuando el muestreo se realiza con reemplazo, este método se llama Bagging (bootstrap aggregating).
- Una vez que todos los predictores estén entrenados, el conjunto puede hacer una predicción para una nueva instancia simplemente agregando las predicciones de todos los predictores.
- La agregación de las predicciones se hace por medio de:
 - Promedio-> Regresión
 - Votación-> Clasificación
- Los hiperparámetros a calibrar son:
 - Número de modelos ($n_{\text{estimators}}$)
 - Tamaño de la muestra de entrenamiento para cada modelo (max_samples)
 - Número de variables que se seleccionan para cada modelo (max_features).

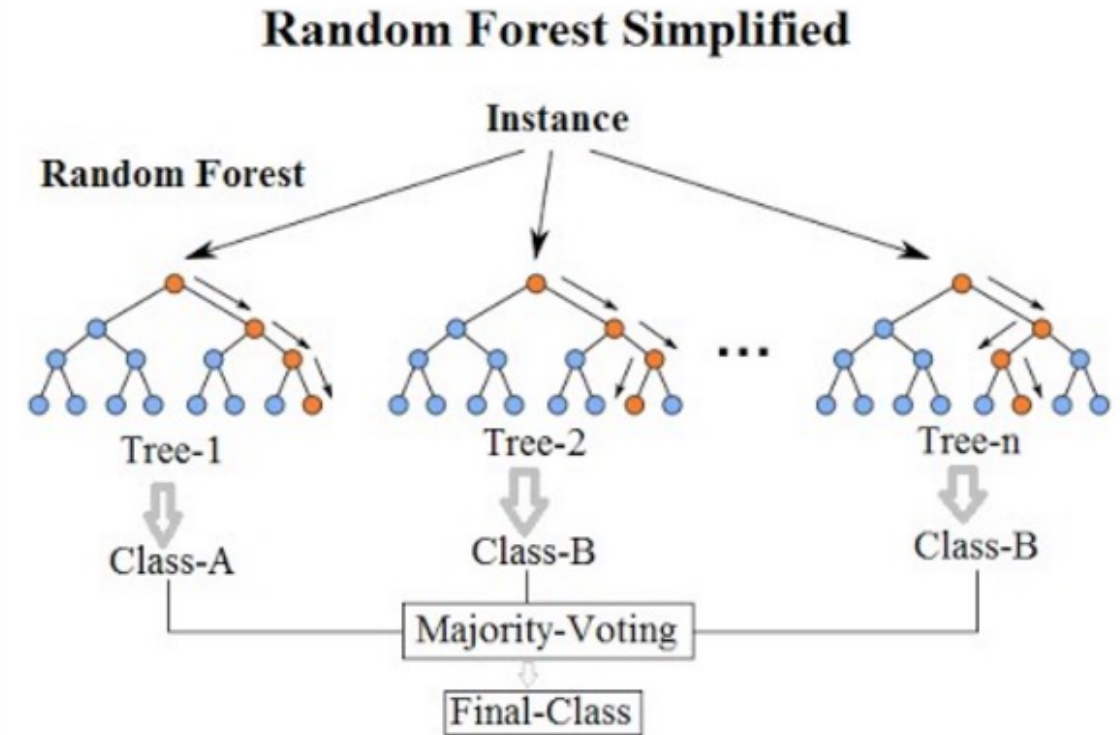


- Con el Bagging, es posible que algunas instancias se muestreen varias veces para cualquier predictor determinado, mientras que es posible que otras no se muestreen en absoluto.
- Bagging muestra m instancias de entrenamiento con reemplazo (bootstrap=True), donde m es el tamaño del conjunto de entrenamiento.
- Esto significa que, en promedio, solo alrededor del 63 % (2/3) de las instancias de capacitación se muestrean para cada predictor.
- El 37 % (1/3) restante de las instancias de capacitación que no se muestrean se denominan instancias *out-of-bag* (oob).
- Dado que un predictor nunca ve las instancias de oob durante el entrenamiento, se puede evaluar en estas instancias, sin necesidad de un conjunto de validación separado.

OOB Evaluation



- El Random Forest es el algoritmo mas conocido que funciona bajo el método Bagging.
- El algoritmo Random Forest introduce una aleatoriedad adicional al generar árboles;
- En lugar de buscar la mejor variable al dividir un nodo, *busca la mejor variable entre un subconjunto aleatorio de características.*
- El algoritmo da como resultado una mayor diversidad de árboles. Lo que generalmente produce un mejor desempeño a comparación de otros enfoque de bagging.
- Los hiperparámetros a calibrar son:
 - Los hiperparámetros del árbol de decisión, en especial la profundidad del árbol
 - Tamaño de la muestra de cada modelo (si se aplica remuestreo de instancias)
 - La cantidad de variables a considerar al buscar la mejor división. Se sugiere iniciar con $\sqrt{\# Variables}$



La predicción final se da por promedio para la regresión o por votación mayoritaria para un problema de clasificación.

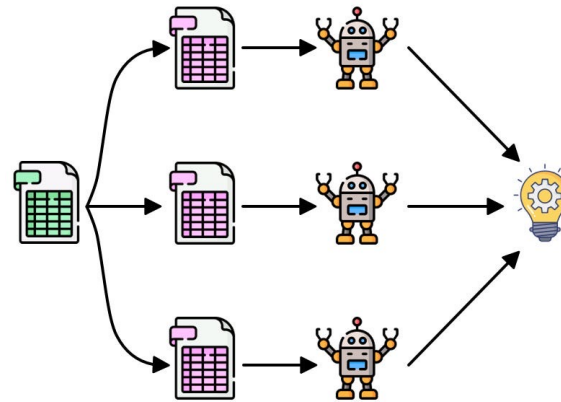
Boosting



Pontificia Universidad
JAVERIANA
Bogotá

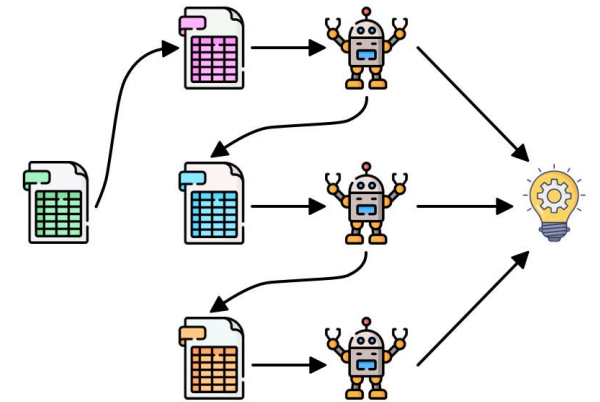
- Boosting se refiere a cualquier método Ensemble que pueda combinar varios modelos débiles en un modelo fuerte.
- La idea general de la mayoría de los métodos de impulso es entrenar predictores secuencialmente, cada uno tratando de corregir a su predecesor.
- El Boosting no implica un muestreo de arranque; en cambio, cada árbol se ajusta a una versión modificada del conjunto de datos original.
- Los algoritmos mas conocidos son:
 - AdaBoost
 - Gradiente Boosting Machine
 - Extreme Gradient Boosting
 - Light Gradient-Boosting Machine

Bagging



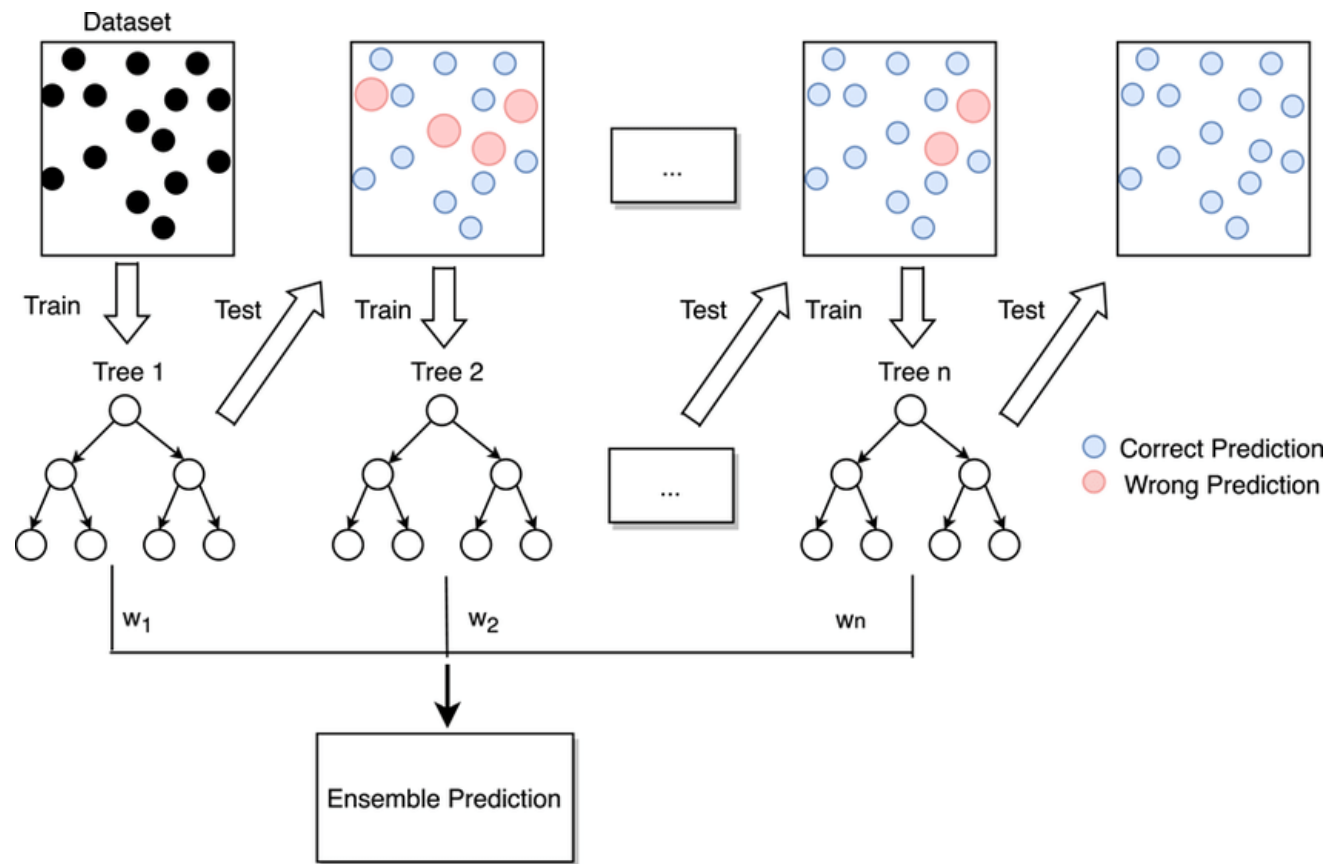
Parallel

Boosting



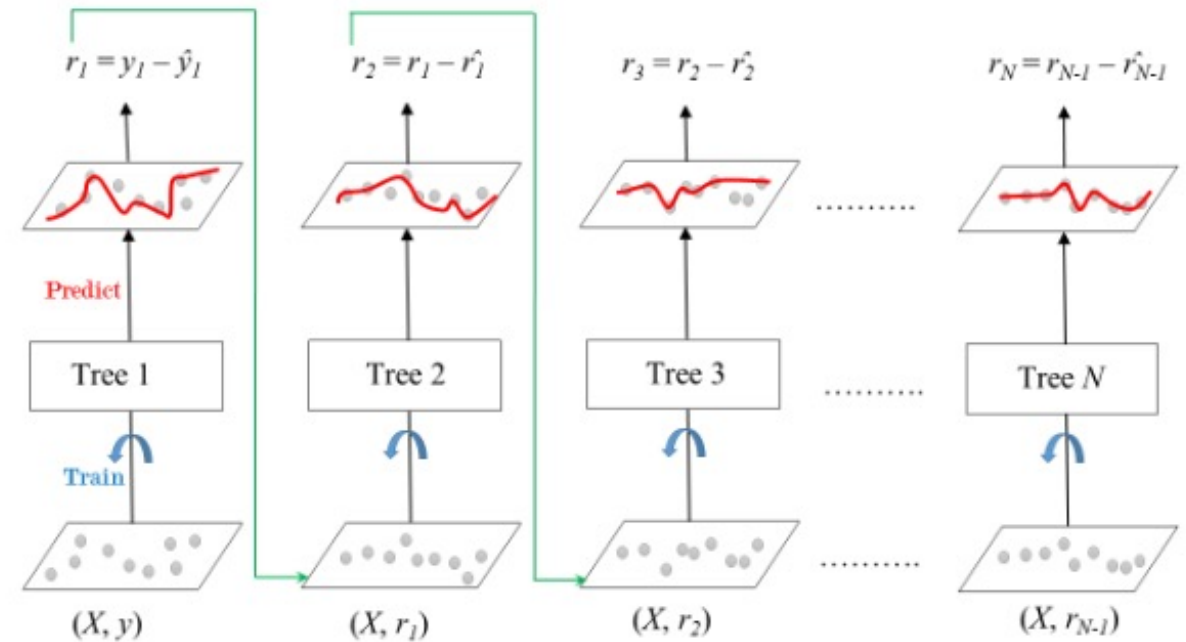
Sequential

- Una forma de que un nuevo modelo corrija a su predecesor es prestar un poco más de atención a las instancias de entrenamiento que el predecesor no adaptó correctamente.
- Esto da como resultado nuevos modelo que se centran cada vez más en los casos mal clasificados o estimados.
- El proceso general que sigue es:
 1. Primero entrena un modelo base (como un árbol de decisión) y lo utiliza para hacer predicciones en el conjunto de entrenamiento.
 2. Luego, el algoritmo aumenta el peso relativo de las instancias de entrenamiento mal clasificadas.
 3. Luego entrena un segundo clasificador, usando los pesos actualizados, y nuevamente hace predicciones en el conjunto de entrenamiento, actualiza los pesos de instancia y repite este procedimiento hasta que se complete el número de modelos deseados.



Gradient Boosting Machine (GBM)

- Es similar al AdaBoost ya los modelos se construyen de forma secuencial tomando como base los resultados de modelos entrenados previamente.
- Sin embargo, en cada etapa, GBM se enfoca en ajustar un nuevo modelo para predecir los errores residuales cometidos por el modelo anterior.
- En otras palabras, se trata de reducir la diferencia entre las predicciones actuales y los valores reales.
- Los modelos débiles en GBM se combinan de manera aditiva, lo que significa que los resultados de los modelos anteriores se suman para formar el modelo fuerte final.
- Los hiperparámetros clave de GBM incluyen el número de árboles (`n_estimators`) y la tasa de aprendizaje (`learning_rate`).



El parámetro de contracción λ , un pequeño número positivo. Esto controla la velocidad a la que el impulso aprende. Los valores típicos son 0,01 o 0,001 y la elección correcta puede depender del problema. Un λ muy pequeño puede requerir el uso de un valor muy grande de modelos para lograr un buen rendimiento.

Ventajas

- Mejora sustancialmente la capacidad predictiva de los modelos simples tales como regresión lineal, regresión logística o árboles de decisión.
- Manejo de la no linealidad y complejidad
- Reducción del sobre ajuste
- Robustez frente a los datos ruidosos o atípicos
- Flexibilidad

Desventajas

- Menos interpretable. A menudo funcionan como cajas negras
- El ajuste de hiperparámetros es más complejo
- Mayor tiempo de entrenamiento
- Requiere gran cantidad de recurso computacional
- Requiere mas datos en comparación a los modelos simples. A menudo no funcionan bien con pocos datos.