Clase 8: Metodología de ensamblaje

Simón Leiva Meza

Pontificia Universidad Católica de Chile
Facultad de Matemáticas
Programa de Magister en Estadística

Bagging
El Bootstrap

El bagging es una metodología de ensamblaje. Esto es un método que toma muchos modelos débiles con la finalidad de generar uno fuerte. bagging (embolsado) viene de "bootstrap aggregating", por lo cual es necesario hacer un repaso al bootstrap antes de seguir.

Bootstrap

Metodología de remuestreo, la cual consiste en estimar la distribución de algún estadístico mediante la aplicación del mismo a muestras con reemplazo. Sea f un estadístico

$$\hat{f} \dot{\sim} f(X_{boot})$$

Donde X_{boot} es una muestra con reemplazo de los datos.

Bagging

Bagging

Sea $\boldsymbol{Z} = \{(y_1, \boldsymbol{x}_1), (y_2, \boldsymbol{x}_2), \ldots (y_n, \boldsymbol{x}_n)\}$ nuestro conjunto de datos de entrenamiento, además sea $\hat{f}(\boldsymbol{x})$ la predicción con el input $\boldsymbol{x}$. Para cada muestra bootstrap $\boldsymbol{Z}_b$, $b = 1, 2, \ldots, B$ ajustamos un modelo, obteniendo la predicción $\hat{f}_b(\boldsymbol{x})$, luego la predicción bagging está dada por:

$$\hat{f}_{bag}(\boldsymbol{x}) = G(\hat{f}_b(\boldsymbol{x}))$$

Donde G es una función de agregación.

Bagging

La función de agregación, puede ser cualquiera que recoja los resultados de los modelos y genere un solo resultado más fuerte. Algunos ejemplos

- **Media:** Se toma el promedio de los resultados obtenidos. Esta se suele usar en los problemas de **regresión**.
- **Votación:** Se elige la categoría con mayor frecuencia entre todas las predicciones. Esto se usa en problemas de **clasificación**.

Bagging
Ejemplo

Considere la tabla loan.csv, usando bagging, ajuste un modelo de regresión logística sencillo con 200 muestras bootstrap y que elijan la predicción mediante un sistema de votación.

Random Forest

El Bagging funciona especialmente bien con algoritmos débiles de alta dispersión pero con poco sesgo. En particular la experiencia a demostrado que con los árboles de decisión se comporta especialmente bien.

Los Random Forest (Breiman, 2001) corresponde a una modificación del bagging con árboles de decisión, la cual consiste en generar muchos árboles no correlacionados y aplicar agregación.

Random Forest

Random Forest

El algoritmo de Random Forest está dado por

1. Para $b = 1$ hasta B:
   a) Extraer una muestra bootstrap $\boldsymbol{Z}$ de tamaño n, igual al tamaño del entrenamiento.
   b) Hacer crecer un árbol T_b a la data bootstrap, repitiendo los siguientes pasos recursivamente para cada nodo terminal del árbol, hasta que el tamaño mínimo del nodo n_{min} es alcanzado.
      i. Seleccionar m variables al azar de las p en total
      ii. Tomar la mejor variable entre las m
      iii. Dividir el nodo en dos nodos hijos
2. Ensamblar los resultados obtenidos

Random Forest
Ejemplo

Ajuste un Random Forest a los mismos datos anteriores. Recuerde evaluar y asegurarse que la muestra esté balanceada.

Random Forest

Los random forest es un clasificador bastante poderoso, aún siendo más débil que otras metodologías (Boosting por ejemplo), es más robusto al ajustar y tiene buen potencial de paralelización. A pesar de no ser una metodología tan explicita en el modelo como un árbol o una regresión, permite interpretarse en el sentido de "importancia de variables". Algunas consideraciones

- Para clasificación, normalmente se usa $m = \sqrt{p}$
- Para regresión, normalmente se usa $m = p/3$
- Al rededor de las 200 muestras bootstrap se estabiliza la precisión.

Random Forest

Dado que los árboles que generan el random forest, generan métricas de eficiencia de las variables en la partición, al ajustar el modelo todas estas son guardadas y acumuladas para todos los árboles en una variable, de esta forma se puede obtener una "importancia relativa" de cada variable.

Ejemplo: Obtenga la importancia relativa de cada variable del último ajuste.

Boosting

El Boosting es una metodología de ensamblaje, la cual busca aprender de errores de clasificadores débiles. al igual que el bagging, busca a partir de un conjunto de clasificadores débiles, obtener uno más robusto. La diferencia con el bagging es que en el boosting el entrenamiento de los modelos es secuencial, buscando aprender de la iteración anterior.
Repasando, si tenemos un conjunto de clasificadores $G_1(x), G_2(x), \ldots, G_m(x)$, el error de predicción de cada clasificador podría medirse como la tasa de desclasificación

$$\text{err}_j = \frac{1}{N} \sum_{i=1}^{N} I(y_i \neq G_j(x_i))$$

Luego la intuición es en lugar del Bagging, que le da el mismo peso a todos, armar un ensamblaje tal que los con menos error tengan más peso. Esta es nuestra primera versión conocida como Adaptative Boosting (AdaBoost)

Boosting
Discrete AdaBoost

A diferencia del Bagging en que se selecciona aleatoriamente una muestra, en el Boosting, esta es ponderada por unos pesos y se realiza un voto ponderado, obteniendo una clasificación como

$$G(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m G_m(x)\right)$$

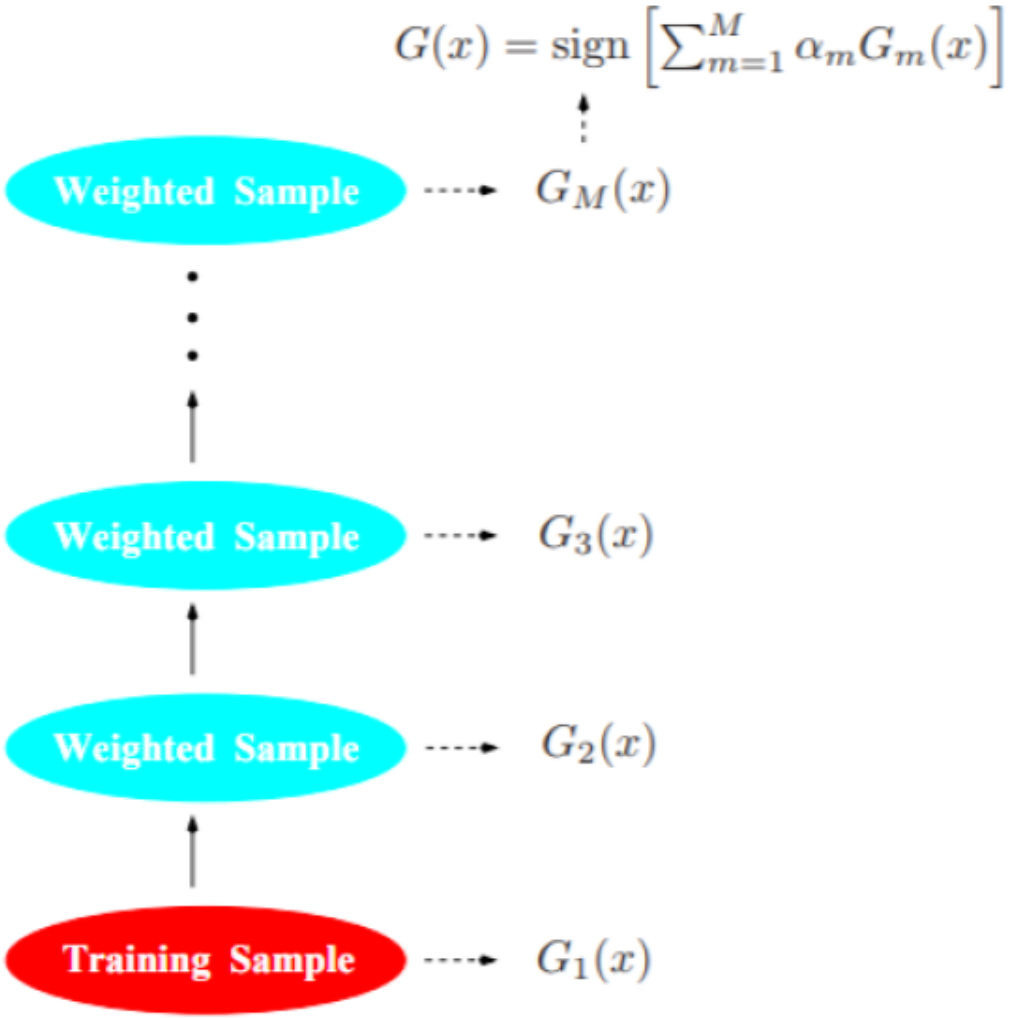
Notar que esta formulación tiene sentido para un clasificador que dicotómico con valores -1 ó 1.
El Algoritmo Adaboost permite encontrar los pesos α con lo cual podremos obtener una nueva clasificación basada en los resultados.

Boosting
Discrete AdaBoost

AdaBoost

1. Iniciar los pesos de cada observación w_i como $w_i = 1/N$ para todo $i = 1 \ldots N$
2. Para $m = 1, \ldots, M$:
   1. Ajustar un clasificador débil $G_m(x)$ a la data de entrenamiento ponderada por los pesos **w**
   2. Calcular
   $$\text{err} = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G(x_i))}{\sum_{i=1}^{N} w_i}$$
   3. Calcular
   $$\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$$
   4. Actualizar w_i como
   $$w_i = w_i \exp(\alpha_m I[y_i \neq G_m(x_i)])$$

Boosting
Discrete AdaBoost



Boosting
Gradient Boosting

Es el tipo de boosting que en general la gente se refiere al hablar de boosting. Bastante popular pues ha logrado capacidad de predicción superior a las metodologías vistas anteriormente. Un Gradient Boosting Model (o Machine dependiendo de la literatura) (GBM), busca ajustar secuencialmente clasificadores débiles, actualizando cada uno por lo que el anterior no pudo predecir bien. Muy informalmente es ir sumando iterativamente a nuestro modelo otro modelo que se ajusta a los residuos del modelo anterior en lugar de ajustarse al entrenamiento.

Boosting
Gradient Boosting

Sea $\boldsymbol{x}, \boldsymbol{y}$ nuestro conjunto de entrenamiento, F un clasificador (o regresor) y sea $L(\boldsymbol{y}, F(\boldsymbol{x})$ una función de pérdida diferenciable. El algoritmo de gradient Boosting es como

Gradient Boosting

1. Iniciar el modelo base y calcular

$$F_0(x) = \text{argmin}_\beta \sum_{i=1}^{N} L(y_i, \beta)$$

2. Para $m = 1, \ldots, M$
   1. Obtener los pseudo-residuos, definidos como

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$$

   para $i = 1, \ldots, M$. Continua

Boosting
Gradient Boosting

Gradient Boosting (Continuación)

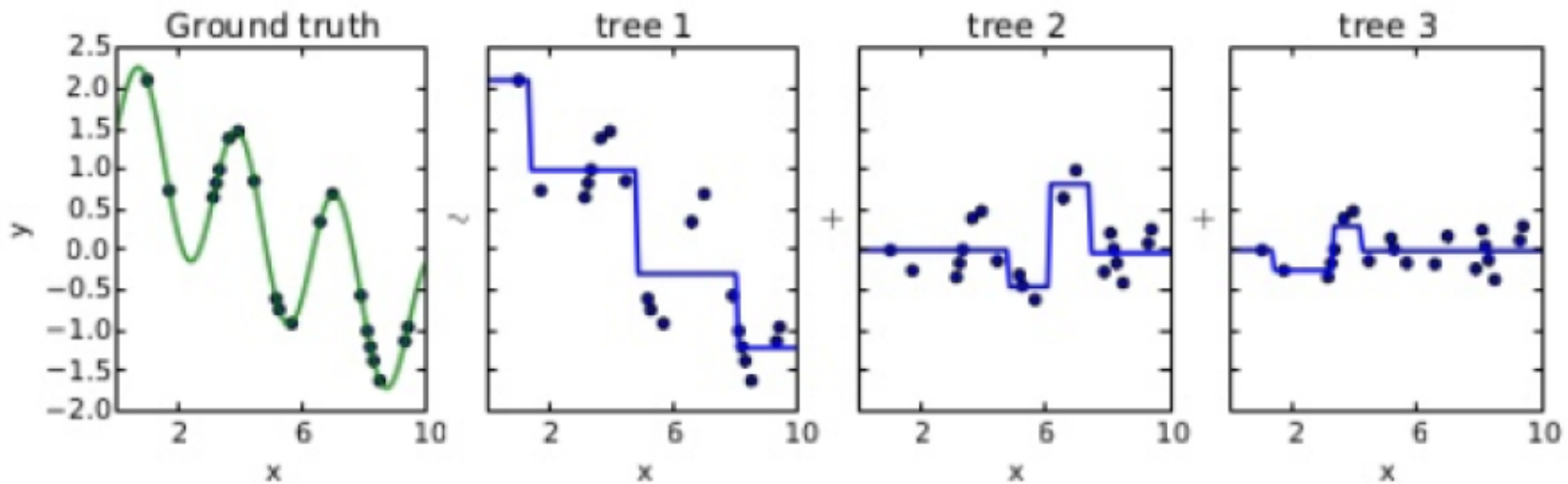
1. Ajustar un modelo $h_m(x)$ a los residuos obtenidos en el paso anterior
2. Calcular el peso del multiplicador γ_m, del nuevo modelo resolviendo

$$\gamma_m = \text{argmin}_\gamma \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \lambda \gamma h_m(x_i))$$

   donde λ es un parámetro llamado **tasa de aprendizaje**
3. Actualizar el modelo como

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

Boosting
Gradient Boosting



Stacking y Blending

En general siempre atacamos los modelos pensamos en

$$y \sim x$$

Qué pasaría si les dijera que si sea M_i un modelo supervisado de clasificación o regresión, también podríamos hacer

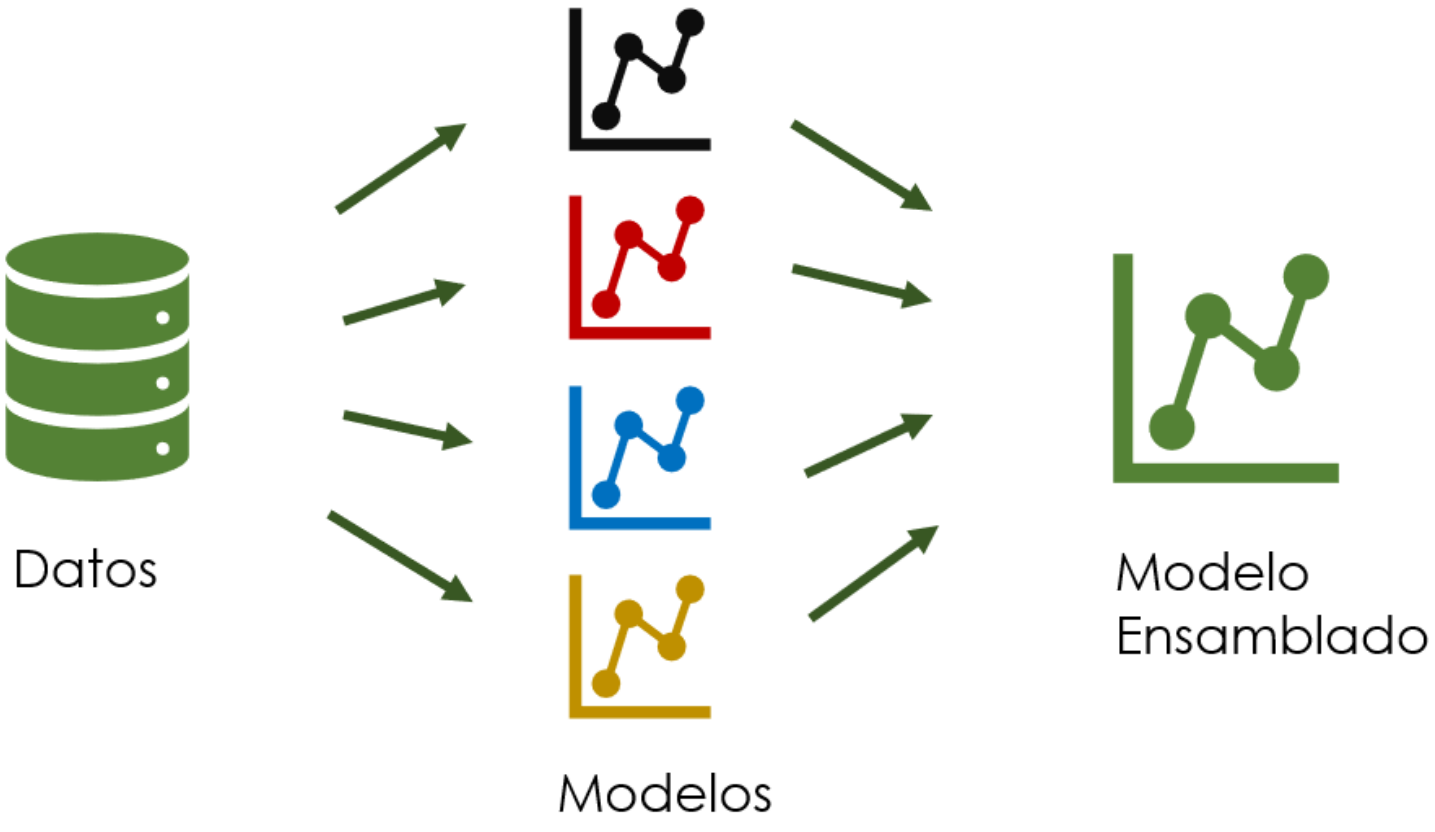
$$y \sim \sum_{i=1}^{M} M_i(x)$$

O incluso

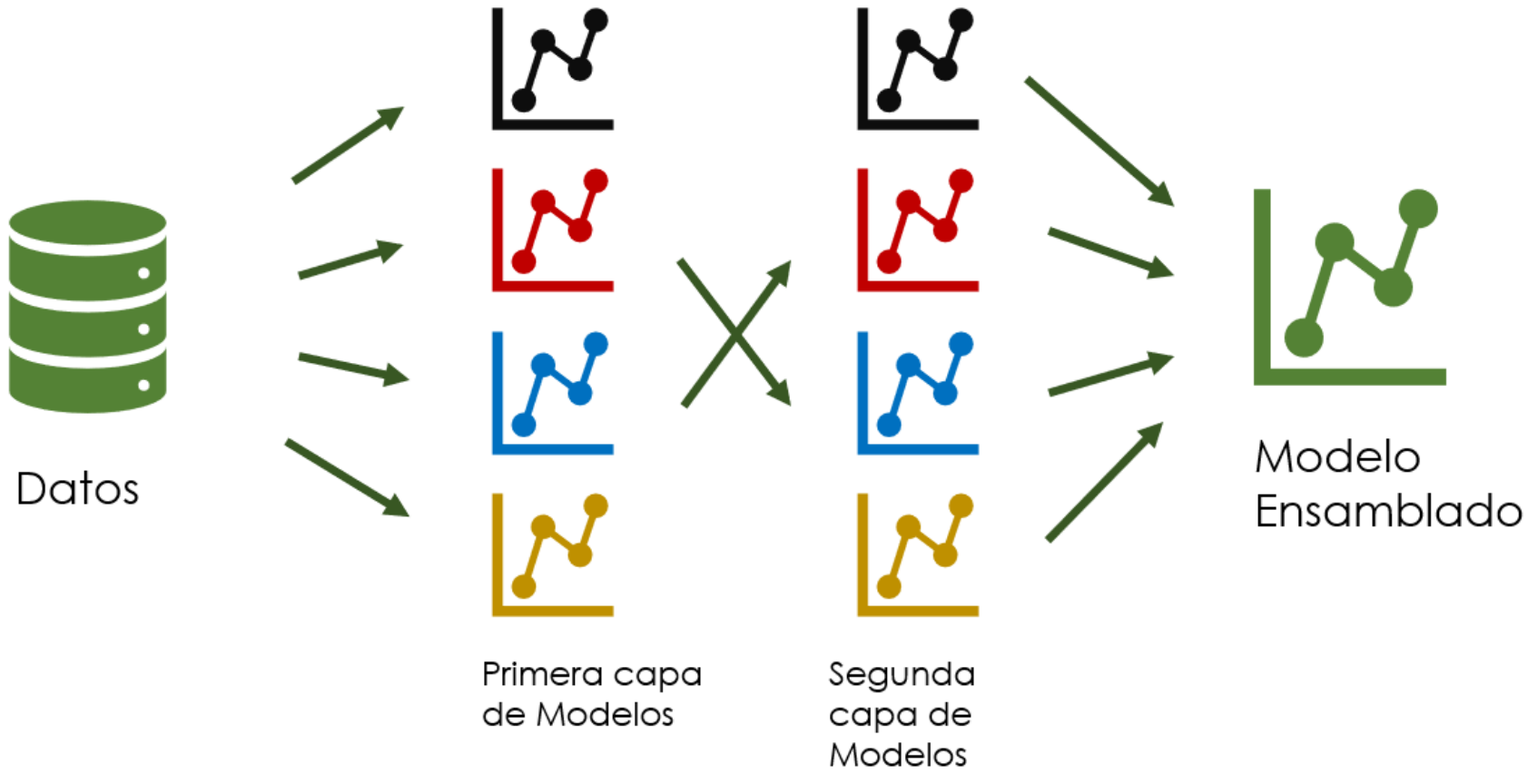
$$y \sim \sum_{i=1}^{M} M_i \left(\sum_{j=1}^{K} M_j(x) \right)$$

Esto es un modelo de modelos, o meta-modelo

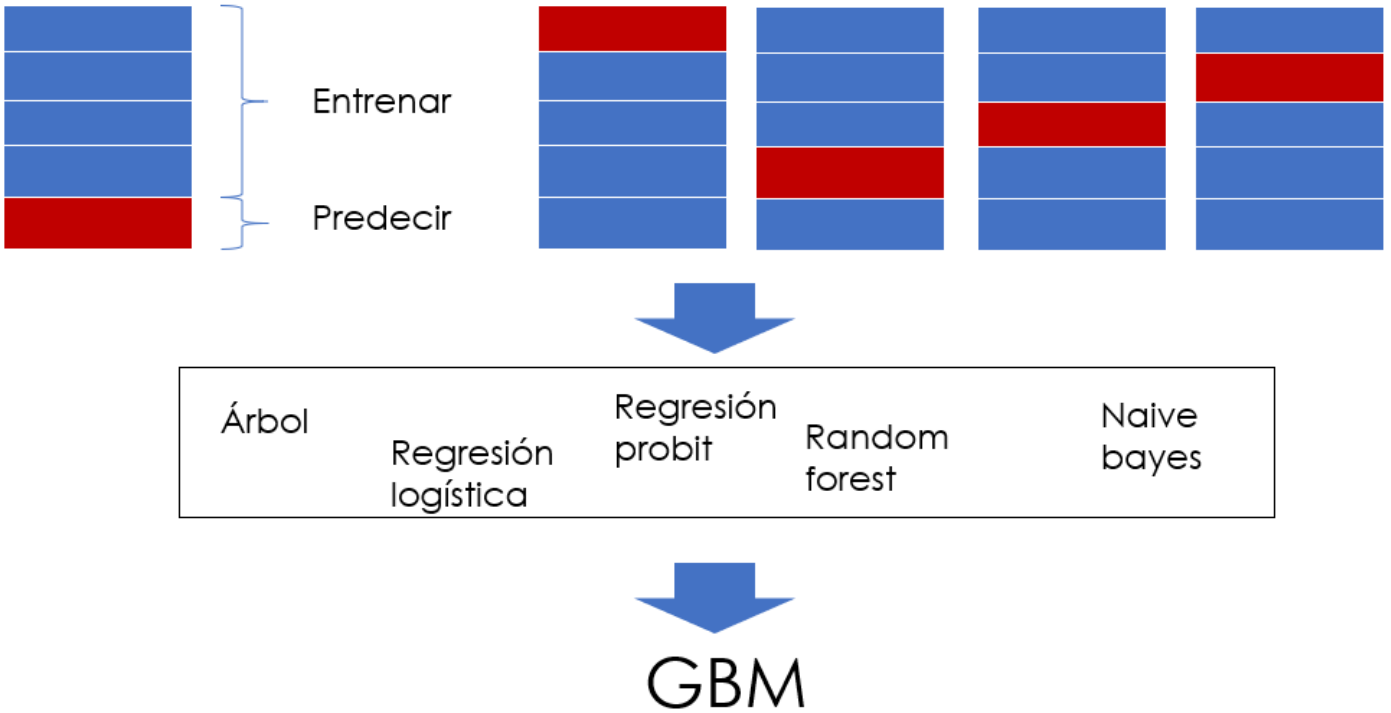
Stacking y Blending



Stacking y Blending

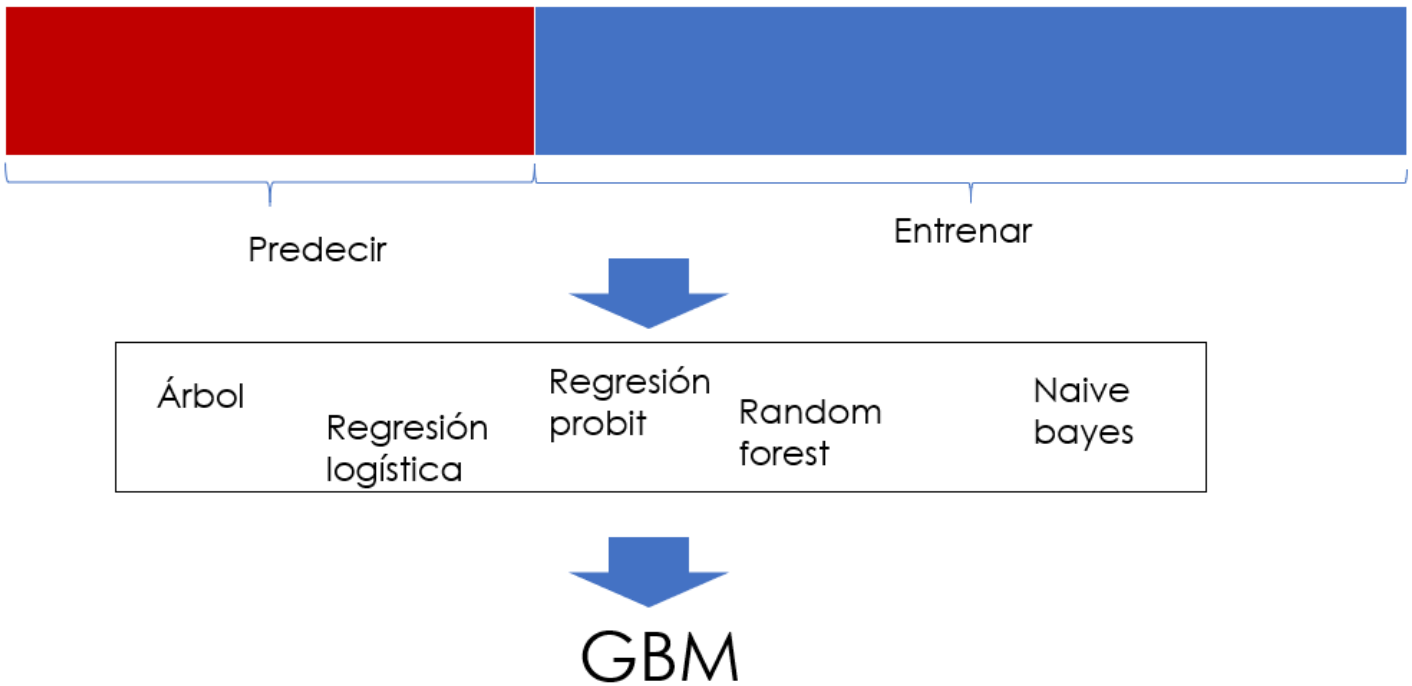


Stacking



Stacking obtiene las predicciones con las cual se ajusta el nuevo modelo utilizando k-fold cross validation, y obteniendo las predicciones en la muestra de validación. A esas predicciones se les ajustan varios modelos, los cuales se ingresan como variables en otro.

Blending



Blending obtiene las predicciones con las cual se ajusta el nuevo modelo utilizando hold-out validation, y obteniendo las predicciones en la muestra de validación. A esas predicciones se les ajustan varios modelos, los cuales se ingresan como variables en otro.

Resumen

Bagging	Permite reducir la variabilidad entre las predicciones. Suele ser estable, preferir en caso de tener predicciones aceptables. Los modelos con alta varianza son los que más mejoran. Los modelos basados en árboles tienen mejor capacidad de tomar las asimetrías pues por detrás categorizan las variables. Los modelos base se entrenan en paralelo
Boosting	Predicciones poderosas, suele tener mejores KPI de ajuste que en el bagging, pero es mucho más inestable que el anterior. Se sugiere usarlo cuando se quiera superar un indicador objetivo. No es interpretable y suele descalibrarse rápidamente. Se recomienda su monitoreo constante. Los modelos base se entrenan secuencialmente.
Stacking y Blending	Se suele tener más control, pues se definen los modelos y los meta modelos. La potencia de las predicciones es variada dependiendo de sus componentes, la complejidad es variable en función de cuantas capas incorporar. De las dos metodologías vistas se diferencian en como usan la muestra de validación.