Bagging y Random Forests _

Motivación

¿Qué son?

- Bagging y Random Forest se conocen como ensambles paralelos.
- Un ensamble paralelo busca evaluar la decisión de un conjunto de modelos entrenados, para posteriormente promediarla.
- El principal problema de modelos como los árboles de decisión es el hecho que generamos una representación única de los datos.

Limitantes de modelos de instancia única

- Cuando entrenamos modelos de instancia única, generamos representaciones limitadas del fenómeno.
- Al seleccionar un modelo dentro de una serie de candidatos, estamos desechando información relevante de los **clasificadores débiles** (Kearns y Valliant, 1989).
- La elección de un modelo específico conlleva a una elección deliberada entre sesgo y varianza:
  - Esto es aún más importante cuando hablamos de árboles de decisión.
- Los métodos de ensambles paralelos permiten agregar múltiples visiones sobre el mismo problema.

Bagging

{desafío}
latam_

Mecanismo de Bagging

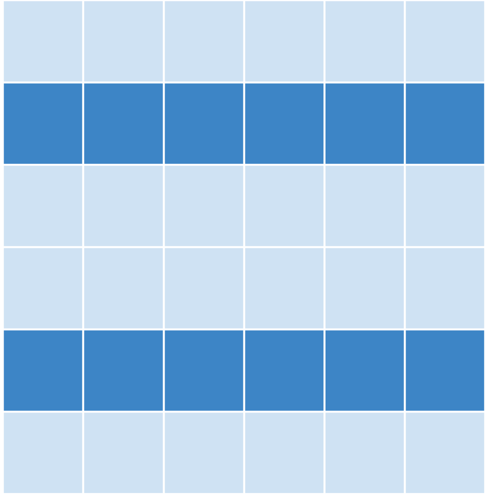
Datos de
Entrenamiento

Mecanismo de Bagging

Cantidad
de modelos

Datos de
Entrenamiento

Subsampling
con Bootstrap



Bootstraping

El objetivo de Bagging es implementar múltiples árboles para regularizar su comportamiento.

Problema: Si entrenamos sobre los mismos datos, incurrimos en sesgo optimista y overfit.

Solución: Para cada modelo implementado en nuestro ensamble, entrenarlo con un subconjunto de datos.

Este subconjunto de datos provendrá de lo que se conoce como **bootstrapping:** Una técnica de muestreo con reemplazo.

Mecanismo de Bagging

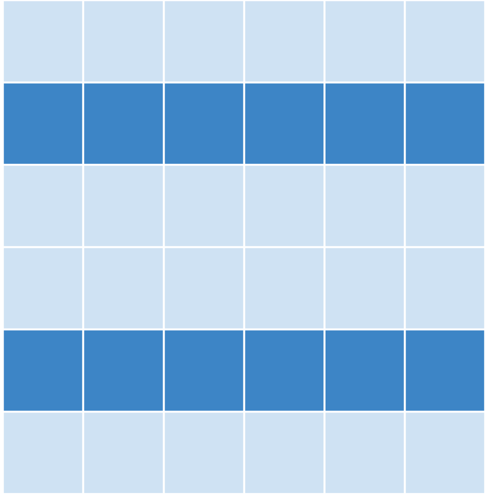
Datos de Entrenamiento

Cantidad de modelos

Subsampling con Bootstrap

Tipo de modelos

Entrenamiento de árbol sin poda



$$\hat{\Pr}_1 = h_1(X_{\text{Train}})$$

Mecanismo de Bagging

Datos de Entrenamiento

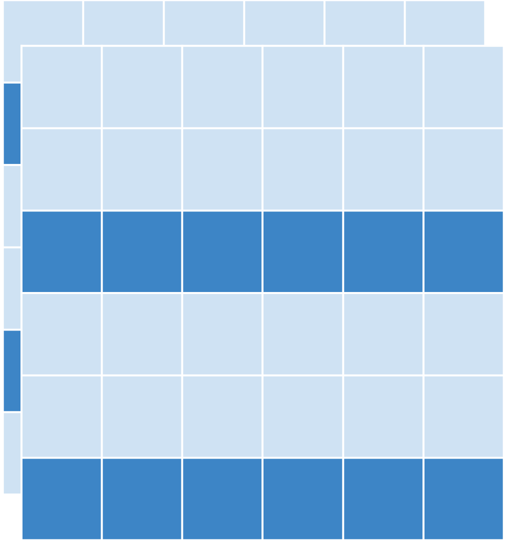
Cantidad de modelos

Subsampling con Bootstrap

Tipo de modelos

Entrenamiento de árbol sin poda

$$\hat{\text{Pr}}_1 = h_1(X_{\text{Train}})$$
$$\hat{\text{Pr}}_2 = h_2(X_{\text{Train}})$$

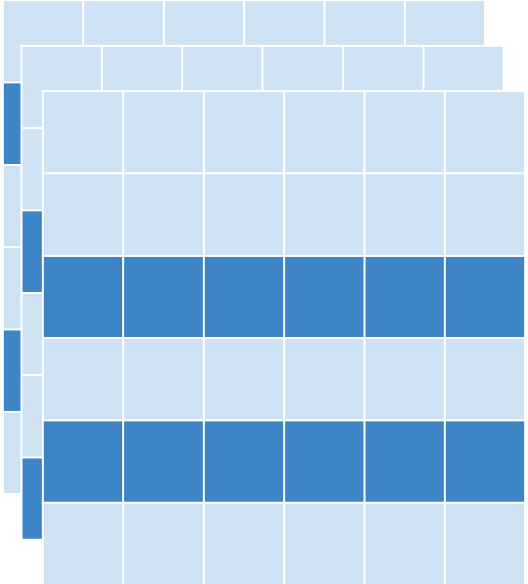


Mecanismo de Bagging

Cantidad de modelos

Tipo de modelos

Datos de Entrenamiento → Subsampling con Bootstrap → Entrenamiento de árbol sin poda



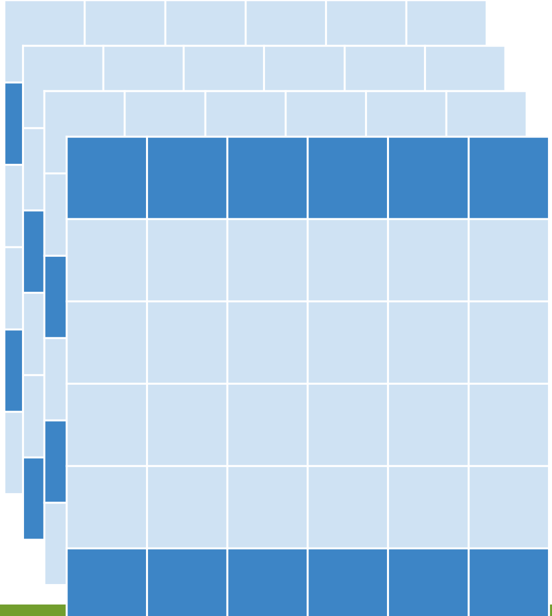
$$\hat{\text{Pr}}_1 = h_1(X_{\text{Train}})$$
$$\hat{\text{Pr}}_2 = h_2(X_{\text{Train}})$$
$$\hat{\text{Pr}}_3 = h_3(X_{\text{Train}})$$

Mecanismo de Bagging

Cantidad de modelos

Tipo de modelos

Datos de Entrenamiento → Subsampling con Bootstrap → Entrenamiento de árbol sin poda



$$\hat{\text{Pr}}_1 = h_1(X_{\text{Train}})$$

$$\hat{\text{Pr}}_2 = h_2(X_{\text{Train}})$$

$$\hat{\text{Pr}}_3 = h_3(X_{\text{Train}})$$

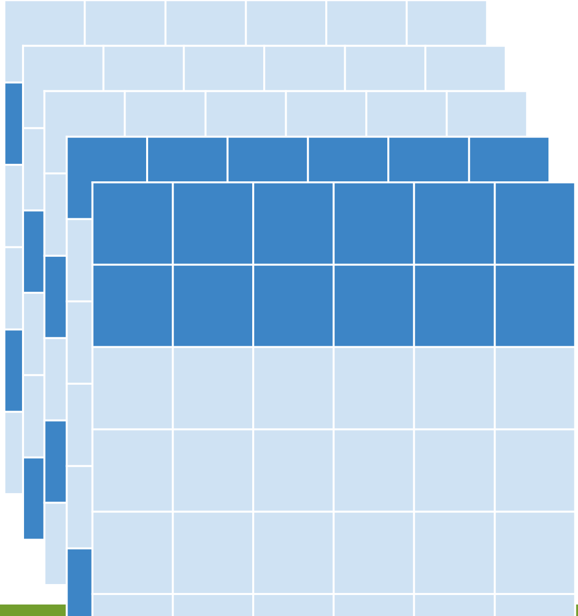
$$\hat{\text{Pr}}_4 = h_4(X_{\text{Train}})$$

Mecanismo de Bagging

Cantidad
de modelos

Tipo de
modelos

Datos de Entrenamiento → Subsampling con Bootstrap → Entrenamiento de árbol sin poda

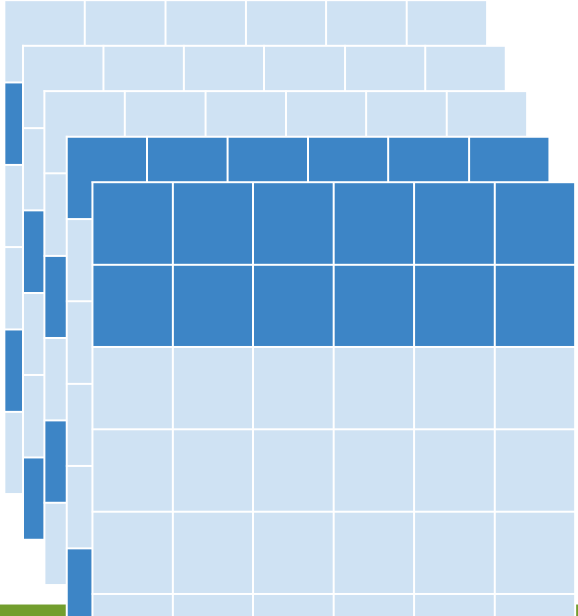


$$\hat{\Pr}_1 = h_1(X_{\text{Train}})$$
$$\hat{\Pr}_2 = h_2(X_{\text{Train}})$$
$$\hat{\Pr}_3 = h_3(X_{\text{Train}})$$
$$\hat{\Pr}_4 = h_4(X_{\text{Train}})$$
$$\hat{\Pr}_5 = h_5(X_{\text{Train}})$$

{desafío}
latam_

Mecanismo de Bagging

Datos de Entrenamiento → (Cantidad de modelos) → Subsampling con Bootstrap → (Tipo de modelos) → Entrenamiento de árbol sin poda → Promedio de las predicciones individuales



$$\hat{\Pr}_1 = h_1(X_{\text{Train}})$$
$$\hat{\Pr}_2 = h_2(X_{\text{Train}})$$
$$\hat{\Pr}_3 = h_3(X_{\text{Train}})$$
$$\hat{\Pr}_4 = h_4(X_{\text{Train}})$$
$$\hat{\Pr}_5 = h_5(X_{\text{Train}})$$

$$\mathcal{H}(x) = \text{sign}\left(\sum_{t=1}^{M} h_t(x)\right)$$

Random Forests

{desafío} latam_

Mecanismo de Random Forest

Datos de
Entrenamiento

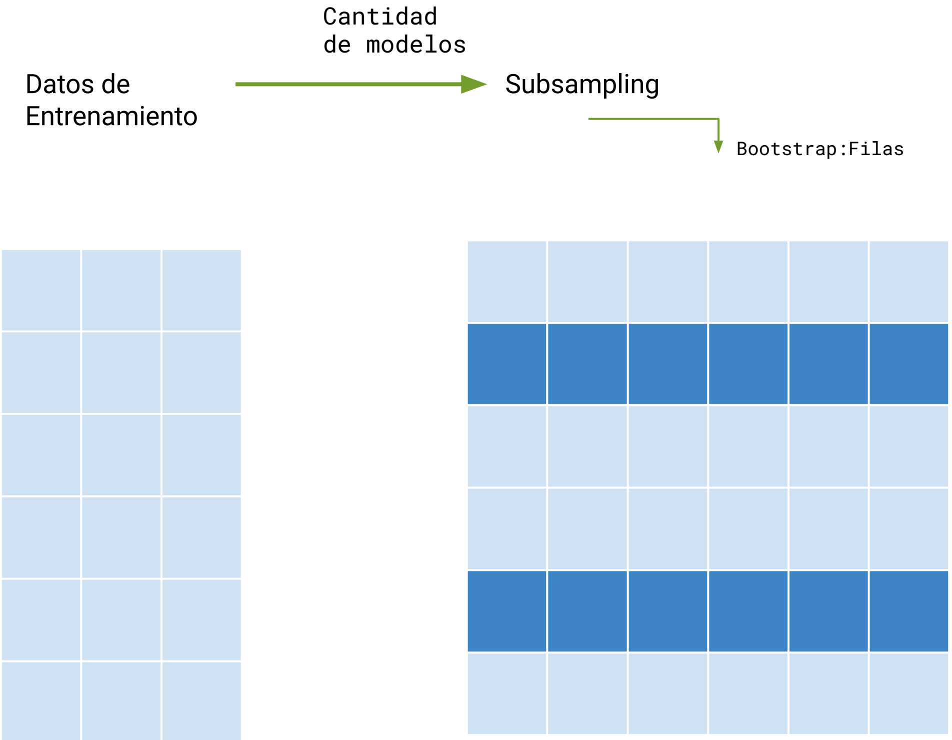
Mecanismo de Bagging

Cantidad
de modelos

Datos de
Entrenamiento → Subsampling

{desafío}
latam_

Mecanismo de Bagging



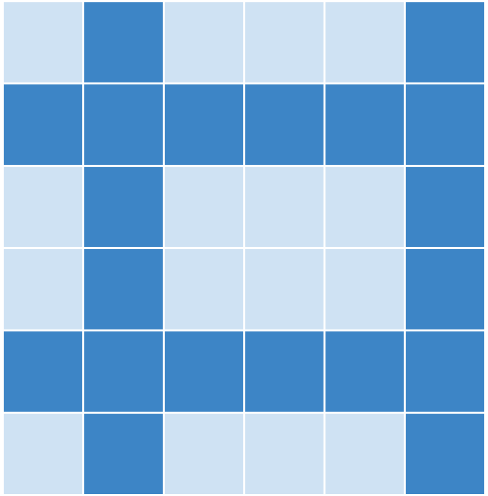
Mecanismo de Bagging

Cantidad
de modelos

Datos de
Entrenamiento

Subsampling

Bootstrap:Filas
Random Attributes: Columnas



Selección Aleatoria de Atributos

- **Problema de Bagging:** Tiende a generar soluciones con sesgo, dado que al entrenar con todos los atributos generamos correlación entre las predicciones de los árboles.
- Para resolver el problema de correlación entre clasificadores, se incluye un mecanismo aleatorio de selección de atributos.
- Durante la construcción de árboles, Random Forests selecciona un subconjunto de atributos de manera aleatoria y prosigue de igual manera con el entrenamiento y selección de particiones. Breiman (2001) sugiere dos formas de definir la cantidad de atributos aleatorizados:
  - El logaritmo del número de atributos.
  - La raíz cuadrada del número de atributos.

Mecanismo de Bagging

Cantidad
de modelos

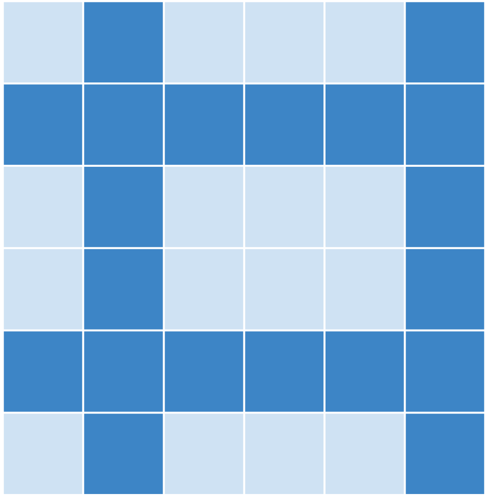
Tipo de
modelos

Datos de
Entrenamiento

Subsampling

Entrenamiento de
árbol sin poda

Bootstrap:Filas
Random Attributes: Columnas



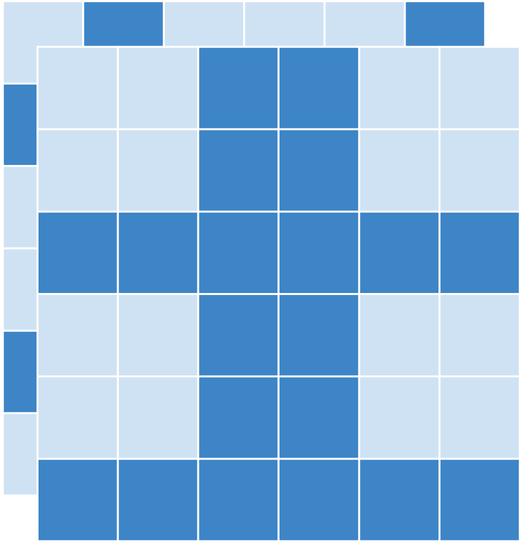
$$\hat{\text{Pr}}_1 = h_1(X_{\text{Train}})$$

Mecanismo de Bagging

Datos de Entrenamiento → (Cantidad de modelos) → Subsampling → (Tipo de modelos) → Entrenamiento de árbol sin poda

Subsampling:

```
Bootstrap:Filas
Random Attributes: Columnas
```

$$\hat{\Pr}_1 = h_1(X_{\text{Train}})$$

$$\hat{\Pr}_2 = h_2(X_{\text{Train}})$$

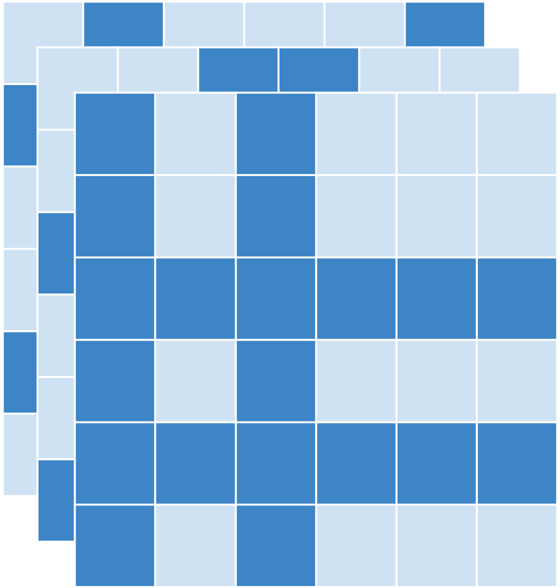
{desafío} latam_

Mecanismo de Bagging

Datos de Entrenamiento → Cantidad de modelos → Subsampling → Tipo de modelos → Entrenamiento de árbol sin poda

Subsampling:

Bootstrap:Filas
Random Attributes: Columnas

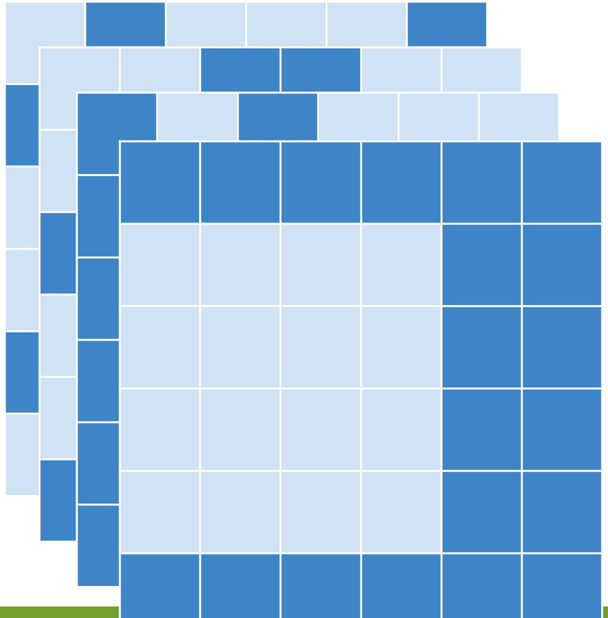


$$\hat{\Pr}_1 = h_1(X_{\text{Train}})$$
$$\hat{\Pr}_2 = h_2(X_{\text{Train}})$$
$$\hat{\Pr}_3 = h_3(X_{\text{Train}})$$

Mecanismo de Bagging

Datos de Entrenamiento → (Cantidad de modelos) → Subsampling → (Tipo de modelos) → Entrenamiento de árbol sin poda

Bootstrap:Filas
Random Attributes: Columnas

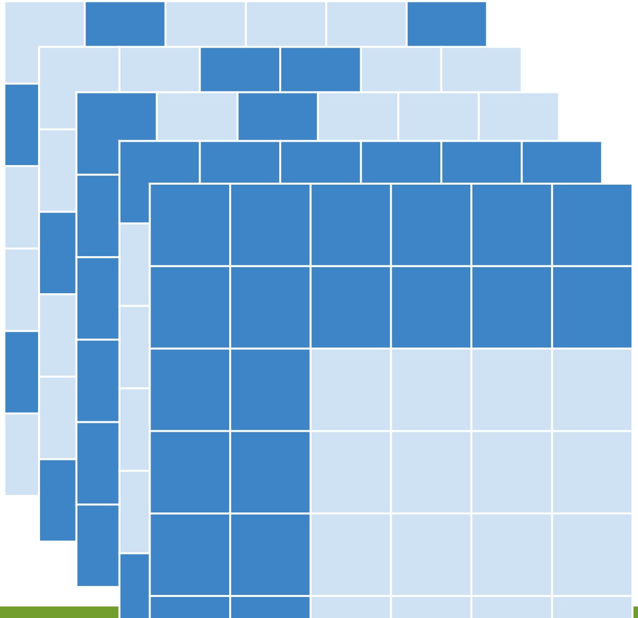


$$\hat{\Pr}_1 = h_1(X_{\text{Train}})$$
$$\hat{\Pr}_2 = h_2(X_{\text{Train}})$$
$$\hat{\Pr}_3 = h_3(X_{\text{Train}})$$
$$\hat{\Pr}_4 = h_4(X_{\text{Train}})$$

Mecanismo de Bagging

Datos de Entrenamiento $\xrightarrow{\text{Cantidad de modelos}}$ Subsampling $\xrightarrow{\text{Tipo de modelos}}$ Entrenamiento de árbol sin poda

Bootstrap:Filas
Random Attributes: Columnas

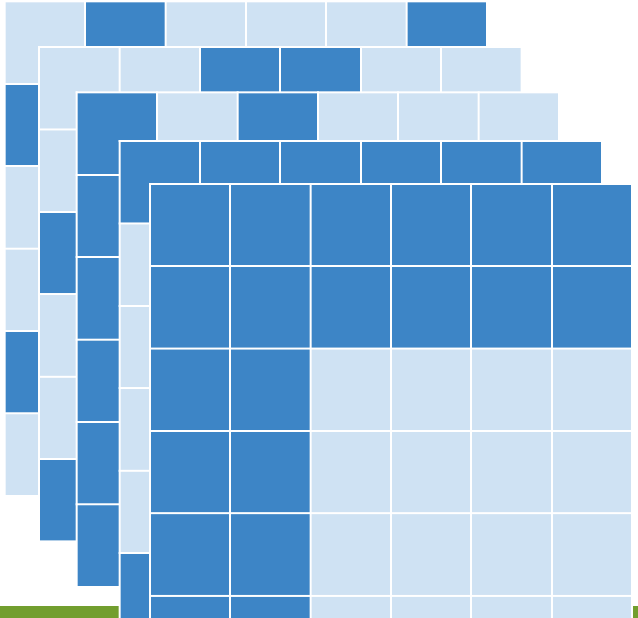


$$\hat{\text{Pr}}_1 = h_1(X_{\text{Train}})$$
$$\hat{\text{Pr}}_2 = h_2(X_{\text{Train}})$$
$$\hat{\text{Pr}}_3 = h_3(X_{\text{Train}})$$
$$\hat{\text{Pr}}_4 = h_4(X_{\text{Train}})$$
$$\hat{\text{Pr}}_5 = h_5(X_{\text{Train}})$$

Mecanismo de Bagging

Datos de Entrenamiento → (Cantidad de modelos) → Subsampling → (Tipo de modelos) → Entrenamiento de árbol sin poda → Promedio de las predicciones individuales

```
Bootstrap:Filas
Random Attributes: Columnas
```

$$\hat{\mathrm{Pr}}_1 = h_1(X_{\text{Train}})$$
$$\hat{\mathrm{Pr}}_2 = h_2(X_{\text{Train}})$$
$$\hat{\mathrm{Pr}}_3 = h_3(X_{\text{Train}})$$
$$\hat{\mathrm{Pr}}_4 = h_4(X_{\text{Train}})$$
$$\hat{\mathrm{Pr}}_5 = h_5(X_{\text{Train}})$$

$$\mathcal{H}(x) = \text{sign}\left(\sum_{t=1}^{M} h_t(x)\right)$$

{desafío} latam_

Out of Bag

- **Problema con los Ensambles:** Son computacionalmente costosos. Realizar validación cruzada o búsqueda de grilla puede ser hasta prohibitivo.
- Bagging/Random Forest devuelven un **error fuera de la bolsa:** En base a las observaciones excluidas de cada bootstrap, generemos una predicción de ésta.
- **Idea:** generar una aproximación a la tasa de errores con validación cruzada en base a los datos ignorados en el bootstrap de cada modelo.
- Para obtener un estimado out-of-bag (OOB), necesitamos de dos pasos:
  - Identificar las observaciones.
  - Estimar el error predictivo en las observaciones.

Out of Bag - Identificación de observaciones

$$\mathcal{H}^{\text{out-of-bag}}(\mathbf{x}) = \underset{y \in \mathcal{Y}}{\text{argmax}} \sum_{t=1}^{T} \mathbb{I}(h_t(\mathbf{x}) = y) \cdot \mathbb{I}(\mathbf{x} \notin D_t)$$

Out of Bag - Identificación de observaciones

$$\mathcal{H}^{\text{out-of-bag}}(\mathbf{x}) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \sum_{t=1}^{T} \mathbb{I}(h_t(\mathbf{x}) = y) \cdot \mathbb{I}(\mathbf{x} \notin D_t)$$

Identificación de
observaciones
out of bag

Out of Bag - Identificación de observaciones

$$\mathcal{H}^{\text{out-of-bag}}(\mathbf{x}) = \underset{y \in \mathcal{Y}}{\text{argmax}} \sum_{t=1}^{T} \mathbb{I}(h_t(\mathbf{x}) = y) \cdot \mathbb{I}(\mathbf{x} \notin D_t)$$

Identificación de observaciones predichas correctamente a nivel de clasificador débil.

{desafío} latam_

Out of Bag - Identificación de observaciones

$$\mathcal{H}^{\text{out-of-bag}}(\mathbf{x}) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \sum_{t=1}^{T} \mathbb{I}(h_t(\mathbf{x}) = y) \cdot \mathbb{I}(\mathbf{x} \notin D_t)$$

Optimización de las clases correctamente predichas fuera del bootstrap.

Out of Bag - Estimación del error

$$\varepsilon^{\text{out-of-bag}} = \frac{1}{|D|} \sum_{\mathbf{x},\mathbf{y} \in D} \mathbb{I}\left(\mathcal{H}^{\text{out-of-bag}}(x \neq y) \right)$$

Out of Bag - Estimación del error

$$\varepsilon^{\text{out-of-bag}} = \frac{1}{|D|} \sum_{\mathbf{x},\mathbf{y} \in D} \mathbb{I}\left(\mathcal{H}^{\text{out-of-bag}}(x \neq y) \right)$$

Identificación de la tasa de clasificación errónea en el out-of-bag

Out of Bag - Estimación del error

$$\varepsilon^{\text{out-of-bag}} = \frac{1}{|D|} \sum_{\mathbf{x},\mathbf{y} \in D} \mathbb{I}\left(\mathcal{H}^{\text{out-of-bag}}(x \neq y)\right)$$

Identificación de los errores de clasificación a nivel de ensamble

Out of Bag - Estimación del error

$$\varepsilon^{\text{out-of-bag}} = \frac{1}{|D|} \sum_{\mathbf{x},\mathbf{y} \in D} \mathbb{I}\left(\mathcal{H}^{\text{out-of-bag}}(x \neq y)\right)$$

Ajuste por la cantidad de datos

{desafío} latam_

Academia de talentos digitales

www.desafiolatam.com