# Modelos de ensamble (parte I)

Clase sincrónica

{desafío} latam_

# Mapa general de la carrera

*Revisión modular*

SQL para el análisis de datos

Programación con Python para el análisis de datos

Análisis estadístico con Python

Visualización y comunicación de insights

Machine Learning*

Redes neuronales*

*Solo para Data Science

{desafío} latam_

# Modelos avanzados y redes neuronales

| Unidad | Clases (sincrónico) | Autoaprendizaje (asincrónico) | Tutoría (sincrónico) |
|---|---|---|---|
| Modelos de ensamble (Parte I) | 2 horas | Desde 6 horas | 2 horas |
| Modelos de ensamble (Parte II) | 2 horas | Desde 6 horas | 2 horas |
| Modelos de ensamble (Parte III) | 2 horas | Desde 6 horas | 2 horas |
| Redes Neuronales (Parte I) | 2 horas | Desde 6 horas | 2 horas |
| Redes Neuronales (Parte II) | 2 horas | Desde 6 horas | 2 horas |
| Procesamiento y Redes recurrentes (parte I) | 2 horas | Desde 6 horas | 2 horas |
| Procesamiento y Redes recurrentes (parte II) | 2 horas | Desde 6 horas | 2 horas |
| *Prueba* | *0 horas* | *Desde 6 horas* | *0 horas* |
| *Receso* | *0 horas* | *0 horas* | *0 horas* |

# ¿Qué aprenderemos en este módulo?

Al final de este módulo serás capaz de comprender el funcionamiento de modelos avanzados y  redes neuronales artificiales para resolver problemas que involucran modelos complejos.

{desafío} latam_

*Implementar ensambles de modelos en problemas complejos, ajustando diferentes factores para optimizar la predicción.*

- **Unidad 1: Modelos de ensamble**
  (Parte I)

  (Parte II)

  (Parte III)
- **Unidad 2: Redes neuronales**
  (Parte I)

  (Parte II)
- **Unidad 3: Procesamiento y Redes recurrentes**
  (Parte I)

  (Parte II)

*Te encuentras aquí*

{desafío} latam_

# ¿Qué aprenderás en esta sesión?

*Aprenderás de algoritmos de aprendizaje supervisado con árboles de decisión y métodos de ensamble paralelos Bagging y Random Forest. Al finalizar sabrás cómo operan, para qué sirven y cómo implementarlos.*

{desafío}
latam_

# ¿En qué se diferencian los problemas de regresión v/s aquellos de clasificación?

# Problemas de regresión y clasificación

*¿En qué consisten?*

**Regresión**

La variable objetivo (Target) es un valor continuo. Por ejemplo, si queremos predecir cuál será la temperatura de un paciente que posee ciertos síntomas.

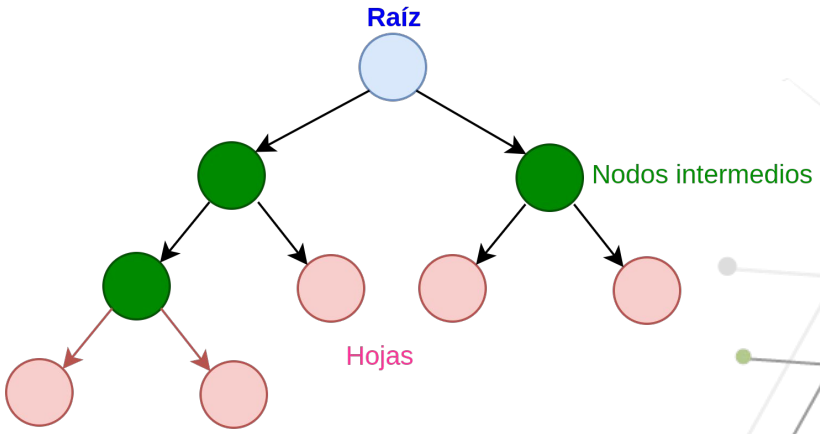
**Clasificación**

La variable Target corresponde a un valor discreto o categórico, codificado. Por ejemplo, si tenemos una base de datos con millones de tweets y queremos poder predecir para cada tweet si es que este representa un sentimiento negativo, neutro o positivo.

# Árboles de decisión

## *Definición*

- Algoritmo de aprendizaje supervisado
- Aplicable a problemas de regresión como de clasificación.
- Separa la muestra en dos regiones cada vez, escogiendo en forma ***óptima*** *los* ***atributos*** *y sus* ***cortes***, generando una especie de árbol inverso.




{desafío} latam_

# Principales Ventajas y Desventajas

## *Principales ventajas y desventajas*

**Ventajas**

- Fáciles de interpretar
- No son afectados por variables en diferentes escalas
- Robustos a valores outliers
- Permiten modelar situaciones no lineales

**Desventajas**

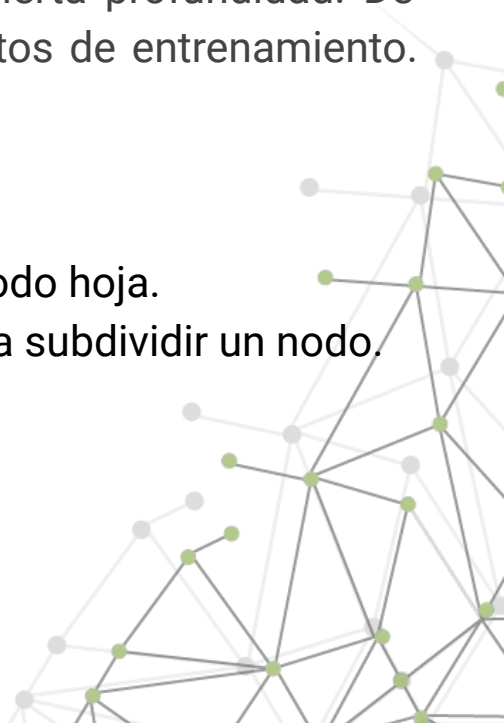
- Overfitting
- Inestabilidad
- Problemas con clases minoritarias

# Overfitting

## *¿Cómo evitarlo?*

Podemos **podar**, es decir, restringir el crecimiento del árbol hasta cierta profundidad. De esta forma regulamos que el árbol no se ajuste en exceso a los datos de entrenamiento. Además, regulamos los siguientes parámetros:

- ***max_depth:*** Máxima profundidad del árbol.
- ***min_samples_leaf:*** Cantidad mínima de observaciones para un nodo hoja.
- ***min_samples_split:*** Número de registros mínimos permitidos para subdividir un nodo.



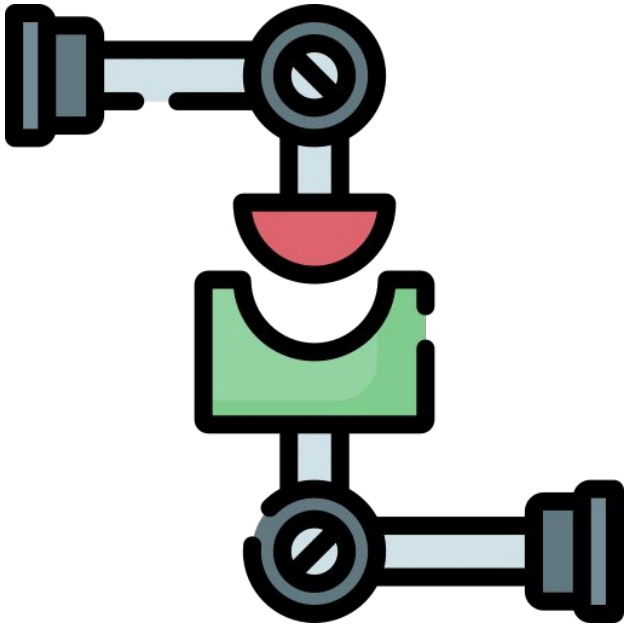
# /* Ensamble Bagging */

# Ensamble

## *Definición*

- Los Ensambles son algoritmos que se componen de dos o más modelos y que juntos persiguen potenciarse para lograr un mejor desempeño.

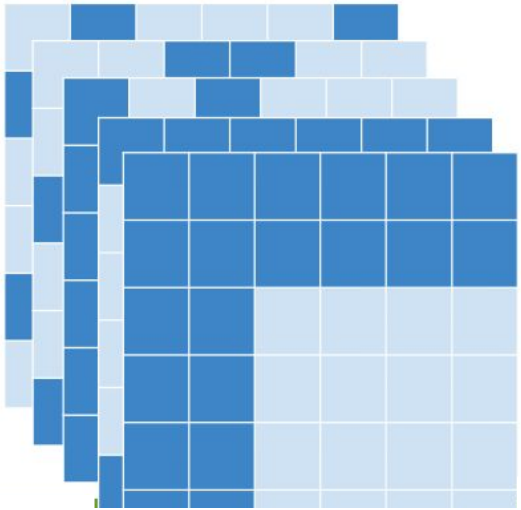
- Para decidir la clasificación (o valor) de los resultados individuales de cada clasificador, se elige democráticamente por “mayoría de votos”.



# Ensamble

## *Random Forest*



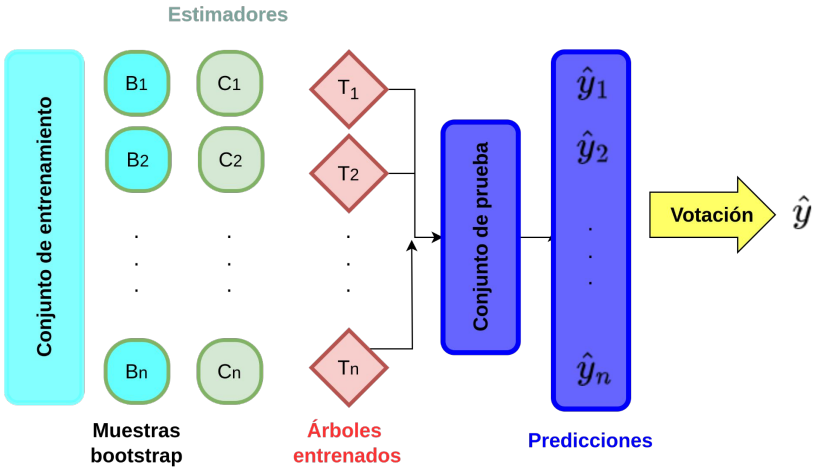
$$\hat{Pr}_1 = h_1(X_{Train})$$
$$\hat{Pr}_2 = h_2(X_{Train})$$
$$\hat{Pr}_3 = h_3(X_{Train})$$
$$\hat{Pr}_4 = h_4(X_{Train})$$
$$\hat{Pr}_5 = h_5(X_{Train})$$

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

# Ensamble Bagging

## *Definición*

- Es un algoritmo de ensamble paralelo.
- Cada estimador usa una muestra Bootstrap.
- Los estimadores (modelos) pueden ser cualquier tipo de clasificador. Por defecto en sklearn se usa predictores de árbol de decisión.
- Las estimaciones de cada modelo se combinan por medio de una votación para predecir la salida.

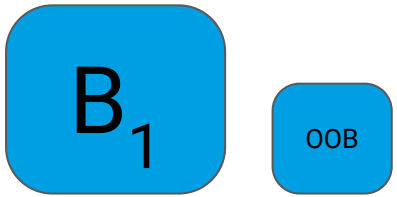



{desafío} latam_

# Ensamble Bagging

## *Características*

- En Bagging, por defecto todos los clasificadores tienen el mismo peso.
- Cada conjunto de entrenamiento obtenido por Bootstrap dejará disponible observaciones sin usar llamadas ***Out Of Bag.***




- Bagging, en general, supera el desempeño de los Árboles de decisión. Sin embargo, presenta problemas ya que los modelos que lo componen resultan ser altamente correlacionados.

# /* Ensamble Random Forest */

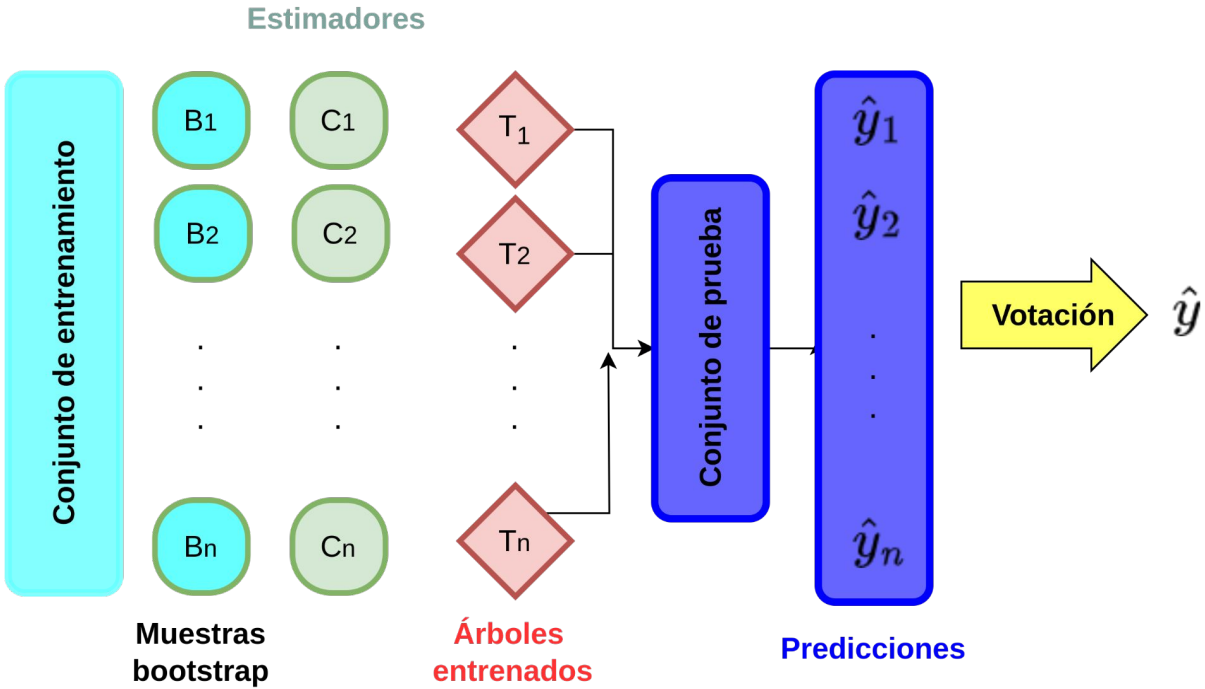
# Random Forest

## *Definición*

- Random Forest (RF) se compone exclusivamente de estimadores de árboles de decisión.
- Permite superar el problema de Bagging de estimadores altamente correlacionados.
- Incluye un componente extra de aleatoriedad, en el que para cada nodo de un árbol de decisión se selecciona en forma aleatoria un subconjunto de atributos que participarán en la selección.

# Random Forest

## *Definición*




{desafío} latam_

# Random Forest

## *Ventajas y desventajas*

**Ventajas**

- Alto desempeño (mejor que Bagging)
- Al igual que Bagging se genera durante el entrenamiento una muestra *Out Of Bag*.
- Robusto frente a outliers.
- Entrega información respecto a la importancia de los atributos.
- En general registra bajo sobreajuste

**Desventajas**

- Complejo de interpretar
- Bajo rendimiento en problemas lineales

# ¡Manos a la obra!
# "Predicción de precios de vivienda"


{desafío}
latam_

# ¡Manos a la obra!

***Predicción de precios de vivienda***

Veremos a continuación la implementación de estos modelos con Python, para lo que trabajaremos en Colaboratory. Abre el archivo destinado para esto y sigue las instrucciones de tu profesor.

{desafío} latam_

# Desafío
# Modelos de ensamble (parte I)


{desafío} latam_

# Desafío

***"Modelos de ensamble (parte I)"***

- Descarga el archivo "Desafío".
- Tiempo de desarrollo asincrónico: desde 2 horas.
- Tipo de desafío: individual.

**¡AHORA TE TOCA A TI! 💪**

{desafío} latam_

# Ideas fuerza

Los ***Árboles de decisión*** *poseen* ***alta interpretabilidad*** *pero sufren de un ajuste excesivo de los datos de entrenamiento* ***overfitting***.

*Los ensambles permiten construir modelos con mejor rendimiento por medio de la* ***combinación de modelos. Bagging*** *puede ser usado con modelos* ***homogéneos o heterogéneos***

***Random Forest*** *posee un* ***alto rendimiento***, *corrigiendo las* ***altas correlaciones*** *de los estimadores de Bagging.*
***Su rendimiento no es favorable para problemas lineales.***

“La suma de las partes entrega  como resultado un valor mayor que el todo”
*Diversidad*

# Próxima sesión...

*Ensambles Secuenciales por medio de Boosting, Adaboost, GradientBoosting y  XGBoost*

{desafío}
latam_

{desafío}
latam_

Academia de
talentos digitales