



Argentina  
programa  
4.0



Universidad  
Nacional  
de San Martín

# Módulo 3

## Aprendizaje Automático



Argentina  
programa  
4.0



Universidad  
Nacional  
de San Martín

# Módulo 3

# Aprendizaje Automático

*Semana 2. Ensembles, Pasting / Bagging*

# Contenidos del módulo

## ML Clásico

- Árboles de Decisión
- Métodos de Ensemble
  - Bagging / Pasting → Random Forests
  - Boosting
- Support Vector Machines

## Deep Learning

- Redes Neuronales
- Redes Neuronales Convolucionales
- Auto-Encoders / Auto-Encoders Variacionales
- Redes Neuronales Recurrentes (LSTM, otras)
- Extras:
  - Generative Adversarial Networks (GAN)
  - Reinforcement Learning

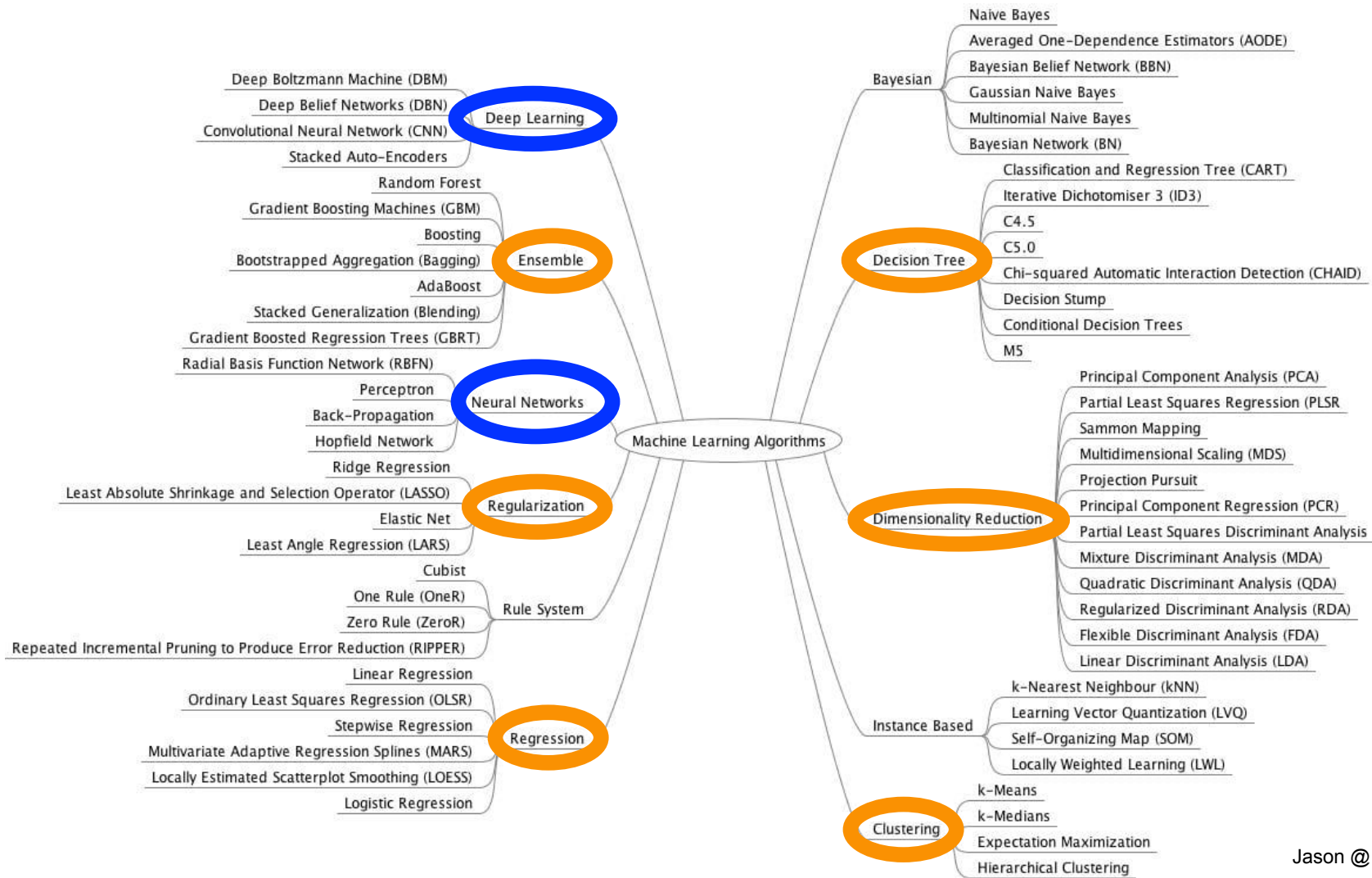
# Contenidos del módulo

## ML Clásico

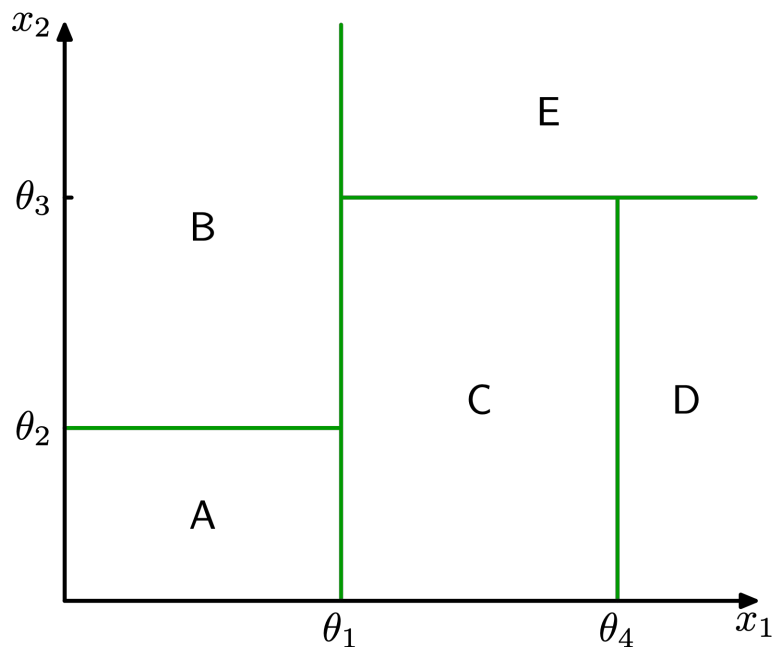
- Árboles de Decisión
- Métodos de Ensemble
  - Bagging / Pasting → Random Forests
  - Boosting
- Support Vector Machines

## Deep Learning

- Redes Neuronales
- Redes Neuronales Convolucionales
- Auto-Encoders / Auto-Encoders Variacionales
- Redes Neuronales Recurrentes (LSTM, otras)
- Extras:
  - Generative Adversarial Networks (GAN)
  - Reinforcement Learning



# Árboles de decisión



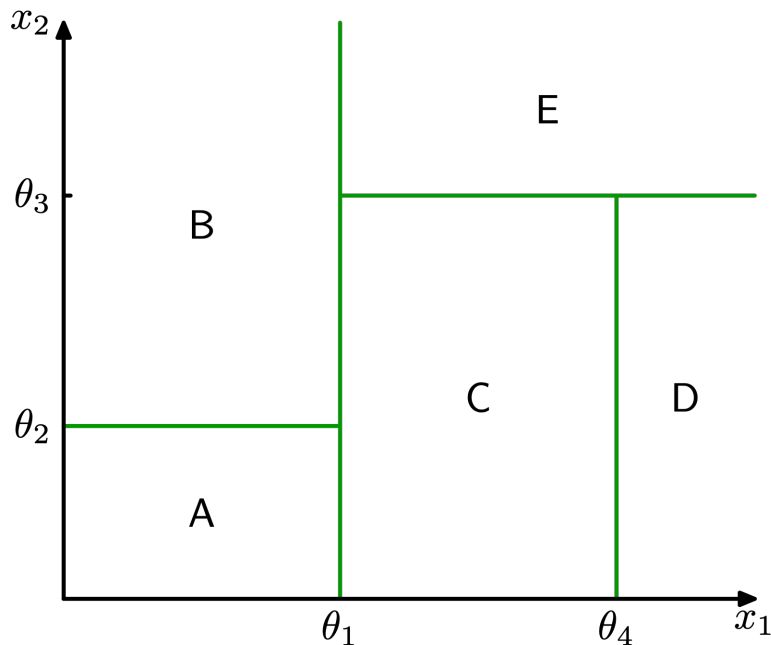
Dado un conjunto de datos, puede ser útil dividir el espacio de características en diferentes regiones, aplicando umbrales secuenciales a las variables.

Se ajusta un **modelo simple** diferente en cada región. Por ejemplo, asignar una clase (o un valor constante en regresión) a todas las muestras que caen en esa región.

En el caso de la clasificación, la probabilidad predicha para la clase puede estimarse con la fracción de muestras de entrenamiento en esa región.

Figura de Bishop

# Árboles de decisión



## Hiperparámetros

- `max_depth`. La profundidad máxima del árbol.
- `min_samples_split`. El número mínimo de muestras necesarias para dividir un nodo interno. Si el número de muestras es menor que este parámetro, el nodo interno se convierte en una hoja.
- `min_samples_leaf`. Número mínimo de muestras (o fracción de muestra si se proporciona un punto flotante) que debe haber en un nodo hoja. Un punto de división en cualquier profundidad sólo se considerará si deja al menos `min_samples_leaf` muestras de entrenamiento en cada una de las ramas izquierda y derecha.
- `max_leaf_nodes`. Una vez crecido el árbol, sólo se conservan los mejores `max_leaf_nodes` nodos hoja. Los mejores nodos se definen a partir de la reducción relativa de la impureza.
- `min_impurity_decrease`. Un nodo será dividido solo si esta división induce una disminución de la impureza mayor o igual a este valor.

Nota: árbol salvaje x árbol regularizado

# Árboles de decisión

## Ventajas

- \* Fácil de entender e interpretar: Los árboles se pueden visualizar. White box.
- \* Requiere poca o nula preparación de los datos.
- \* Predicciones muy rápidas.

## Desventajas

- \* Árboles demasiado complejos que no generalizan bien los datos.
- \* Inestables frente a pequeñas variaciones en los datos
- \* No son buenos para la extrapolación.
- \* Árbol óptimo es un problema NP-completo.
- \* Árboles sesgados si algunas clases dominan. Se recomienda equilibrar el conjunto de datos antes de ajustarlo con el árbol de decisión.



# Ensembles: combinación de modelos para mejorar el desempeño

## Comités

- Combinar modelos - que podrían ser malos aprendices individualmente - para producir un mejor resultado.
- A diferencia de los árboles de decisión (DT), los miembros del comité producen predicciones de las mismas regiones del espacio de características, y son sus resultados los que se combinan.
- Los modelos que combinamos pueden ser el mismo modelo actuando en diferentes conjuntos de datos, o diferentes tipos de modelos, o una combinación de ambos

## Combinando predicciones

Regresión

**promedio** de las predicciones del modelo

Clasificación

**moda** de la distribución (clase más votada)  
moda pesada (si se proporciona una probabilidad)

Fuerte

votación suave

# Ensembles: combinación de modelos para mejorar el desempeño

## Comités

- Combinar modelos - que podrían ser malos aprendices individualmente - para producir un mejor resultado.
- A diferencia de los árboles de decisión (DT), los miembros del comité producen predicciones de las mismas regiones del espacio de características, y son sus resultados los que se combinan.
- Los modelos que combinamos pueden ser el mismo modelo actuando en diferentes conjuntos de datos, o diferentes tipos de modelos, o una combinación de ambos

## Combinando predicciones los errores disminuyen!

Error del comité

$$E_{\text{COM}} = \frac{1}{M} E_{\text{AV}}.$$

$$\begin{aligned} \mathbb{E}_{\mathbf{x}} [\epsilon_m(\mathbf{x})] &= 0 \\ \mathbb{E}_{\mathbf{x}} [\epsilon_m(\mathbf{x}) \epsilon_l(\mathbf{x})] &= 0, \quad m \neq l \end{aligned}$$

Nota: eso requieren que los errores de los comités sean totalmente independientes, cosa que en general no son

# Árboles extremadamente aleatorizados: ExtraTrees

Un ejemplo con distintos modelos actuando en el mismo conjunto de datos

- Para hacer ensambles necesitamos de resultados independientes
- Árboles extremadamente aleatorizados / *extremely randomized trees*

## Algunos parámetros

- `splitter`: Para cada característica, el algoritmo encuentra la mejor división y calcula su importancia (la reducción de impurezas), dada por el parámetro `criterion` (que puede ser `gini` o `entropy`). Si `splitter` se elige como `best`, se elige el mejor corte de la mejor característica. Si `splitter='random'`, se eligen cortes al azar para cada característica y se utiliza el mejor entre ellos.
- `max_features`: En cada división, se considera sólo un subconjunto aleatorio de características `max_features`.

# Conjuntos (*Ensembles*): por qué eso es una buena idea, en principio

Errores de los modelos individuales son promediados

$$y_m(\mathbf{x}) = h(\mathbf{x}) + \epsilon_m(\mathbf{x}).$$

$$m = 1, \dots, M$$

$$E_{\text{AV}} = \frac{1}{M} \sum_{m=1}^M \mathbb{E}_{\mathbf{x}} [\epsilon_m(\mathbf{x})^2].$$

Error del comité

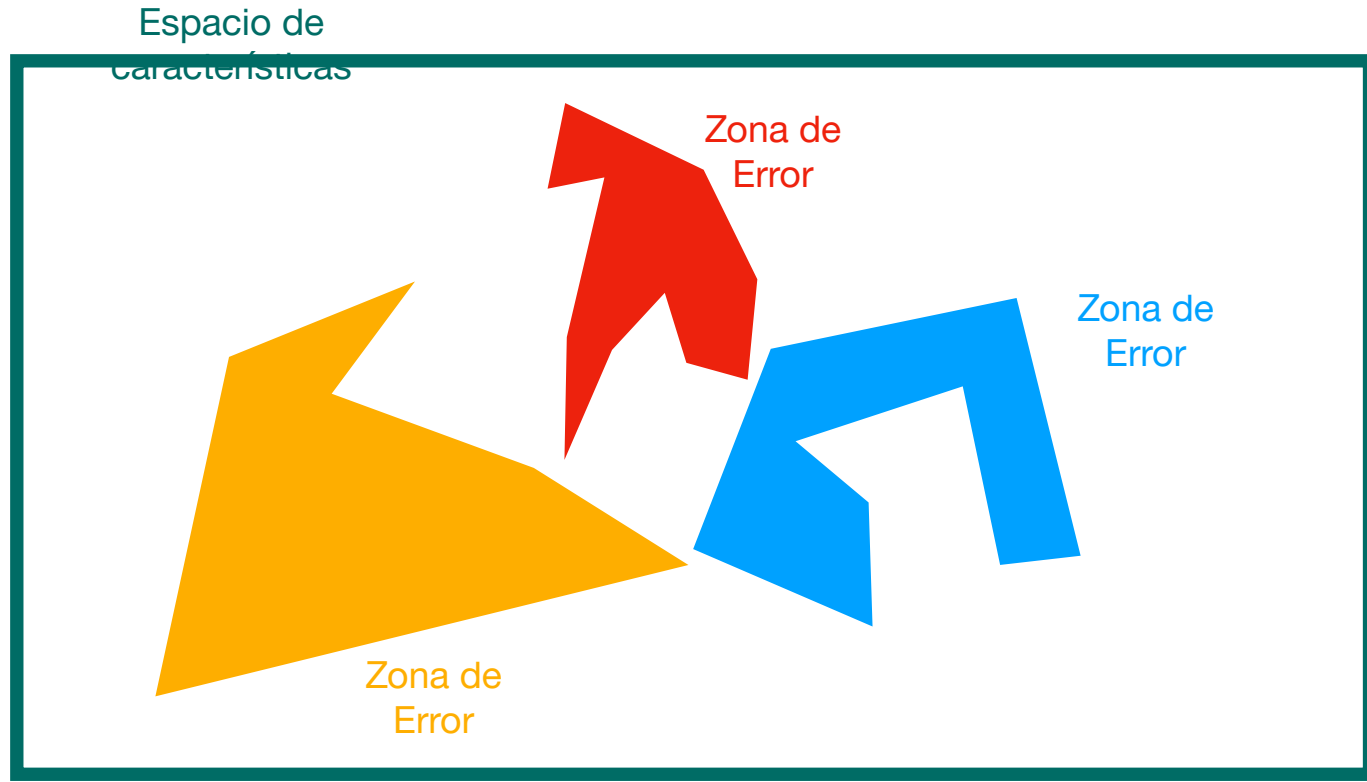
$$E_{\text{COM}} = \frac{1}{M} E_{\text{AV}}.$$

Advertencia: solo válido si

$$\begin{aligned} \mathbb{E}_{\mathbf{x}} [\epsilon_m(\mathbf{x})] &= 0 \\ \mathbb{E}_{\mathbf{x}} [\epsilon_m(\mathbf{x})\epsilon_l(\mathbf{x})] &= 0, \quad m \neq l \end{aligned}$$

(eso nunca ocurre exactamente)

# *Ensembles*: combinación de modelos para mejorar el desempeño



Inspirado en diapos de A. Farall

# Generando nuevos conjuntos de datos: *Pasting* y *Bagging*

Error del comité

$$E_{\text{COM}} = \frac{1}{M} E_{\text{AV}}.$$

$$\begin{aligned} \mathbb{E}_{\mathbf{x}} [\epsilon_m(\mathbf{x})] &= 0 \\ \mathbb{E}_{\mathbf{x}} [\epsilon_m(\mathbf{x})\epsilon_l(\mathbf{x})] &= 0, \quad m \neq l \end{aligned}$$

Nota: en general los errores de los comités no son independientes

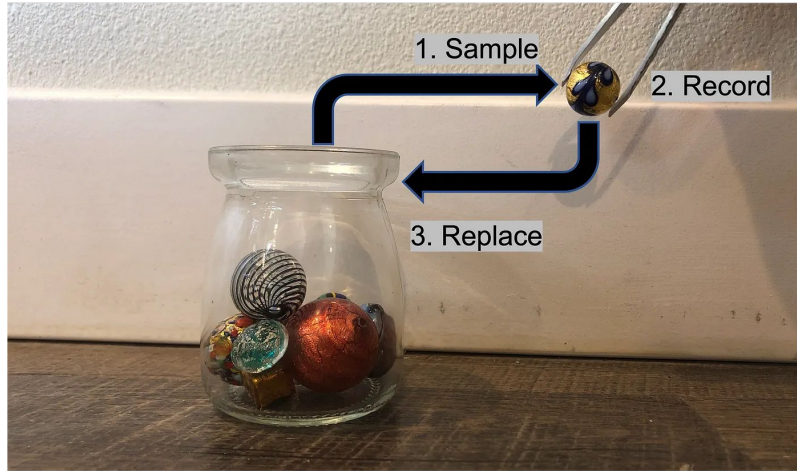
Podemos obtener un comité de estimadores diferentes a partir de un único estimador base, si lo entrenamos en distintos conjuntos de datos.

Generación de diferentes conjuntos de datos mediante la selección aleatoria de muestras o características:

- submuestreo aleatorio (sin reemplazo) del conjunto de entrenamiento: *Pasting*
- submuestreo aleatorio con reemplazo: *Bootstrap aggregating (Bagging)*
- subconjunto de características: *Random Subspaces*
- subconjuntos tanto de muestras como de características: *Random Patches*

# Generación de submuestras con y sin reemplazo

## *Bootstrap* : sampleo con reemplazo



## Sampleo sin reemplazo

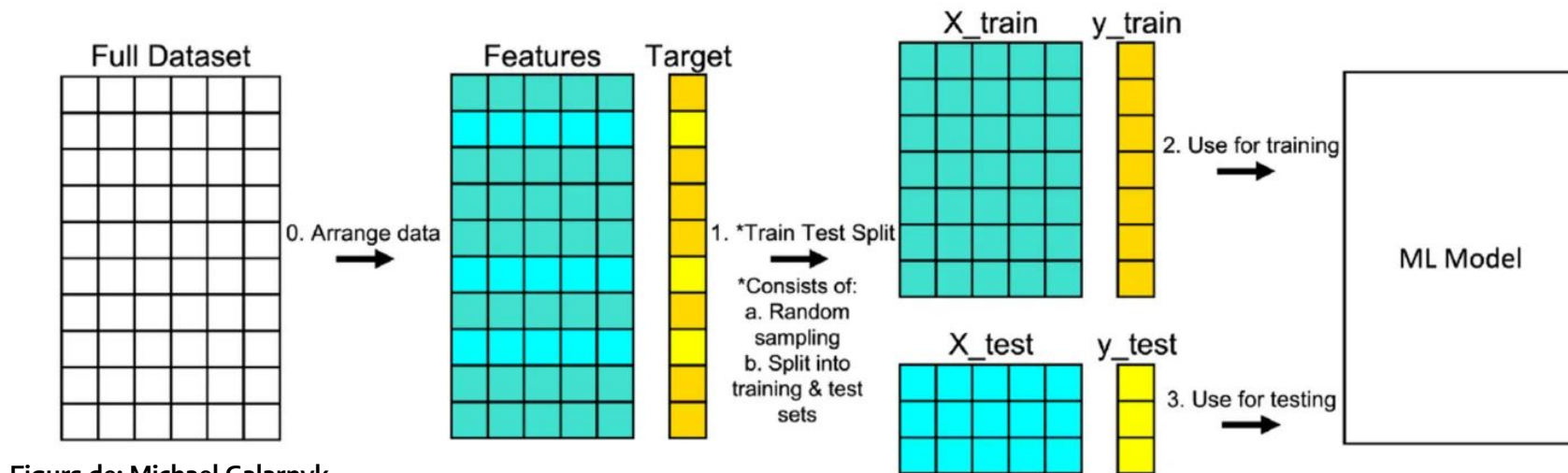


Figuras de: Michael Galarnyk

# Sin reemplazo

¿Cuándo se usan submuestras generadas por sampleo sin reemplazo?

Cuando queremos que las muestras sean totalmente independientes: e.g. separación entre entrenamiento y validación, cross validation

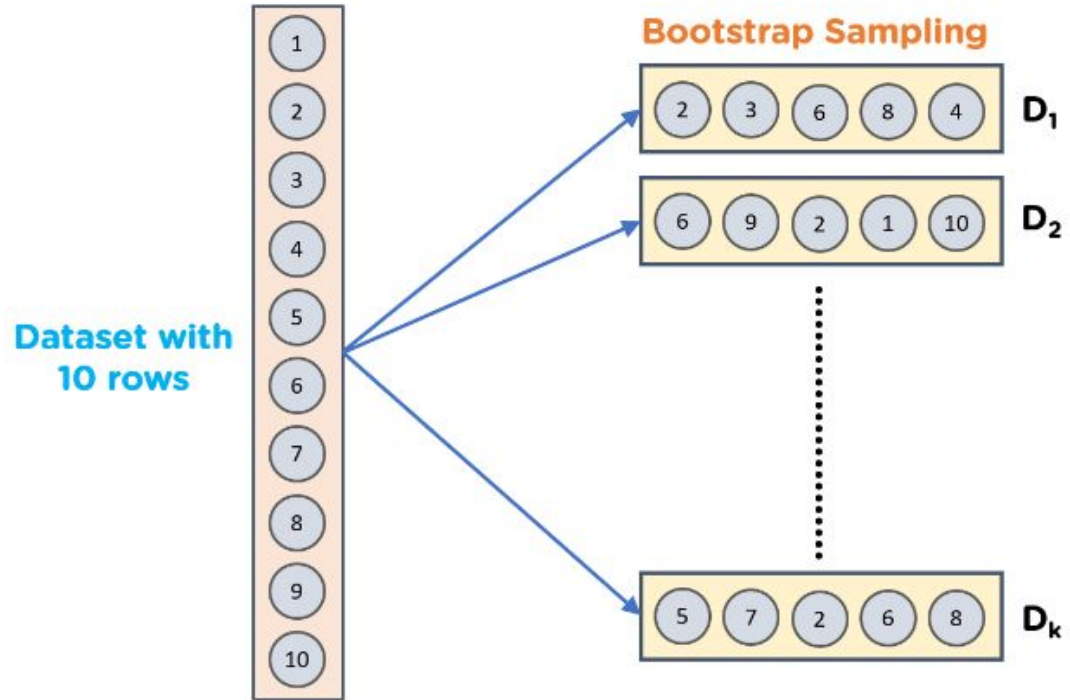


Figurs de: Michael Galarnyk



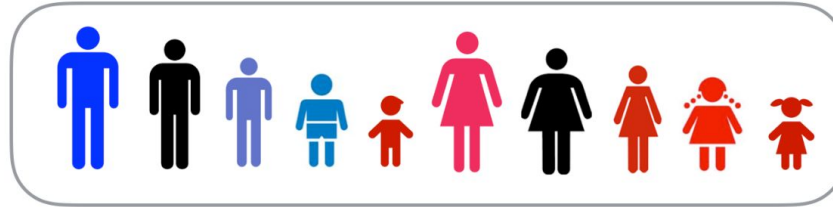
# Bootstrapping

- Generar aleatoriamente sub-muestras a partir de una sola muestra
- *Bagging*: cada línea/elemento se sortea independientemente para cada elemento de la nueva (sub-)muestra
- La sub-muestra puede tener el número de elementos que se desee



# Bootstrapping

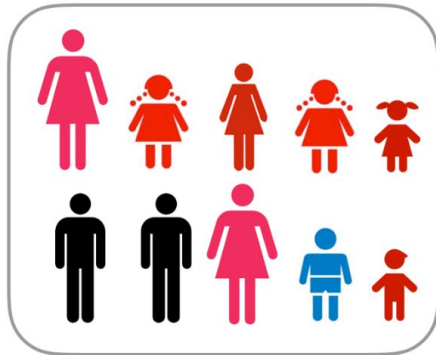
Observed Sample = #N persons of different heights



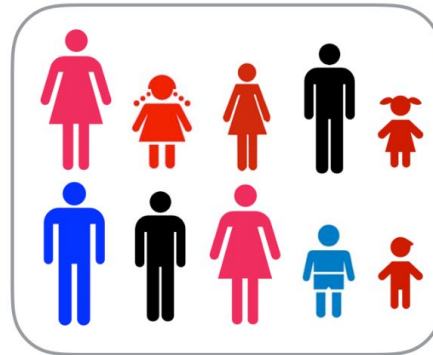
Average height?

Bootstrapping  
(e.g. sampling with replacement)

#M Bootstrap Samples

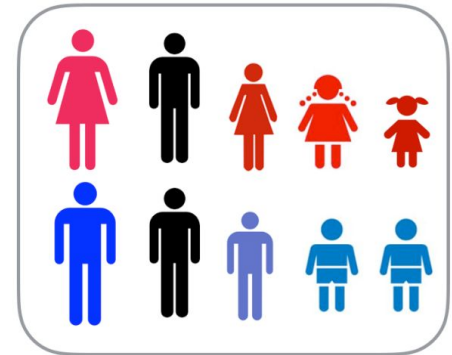


1<sup>st</sup> Avg. Height



2<sup>nd</sup> Avg. Height

.....  
.....  
.....

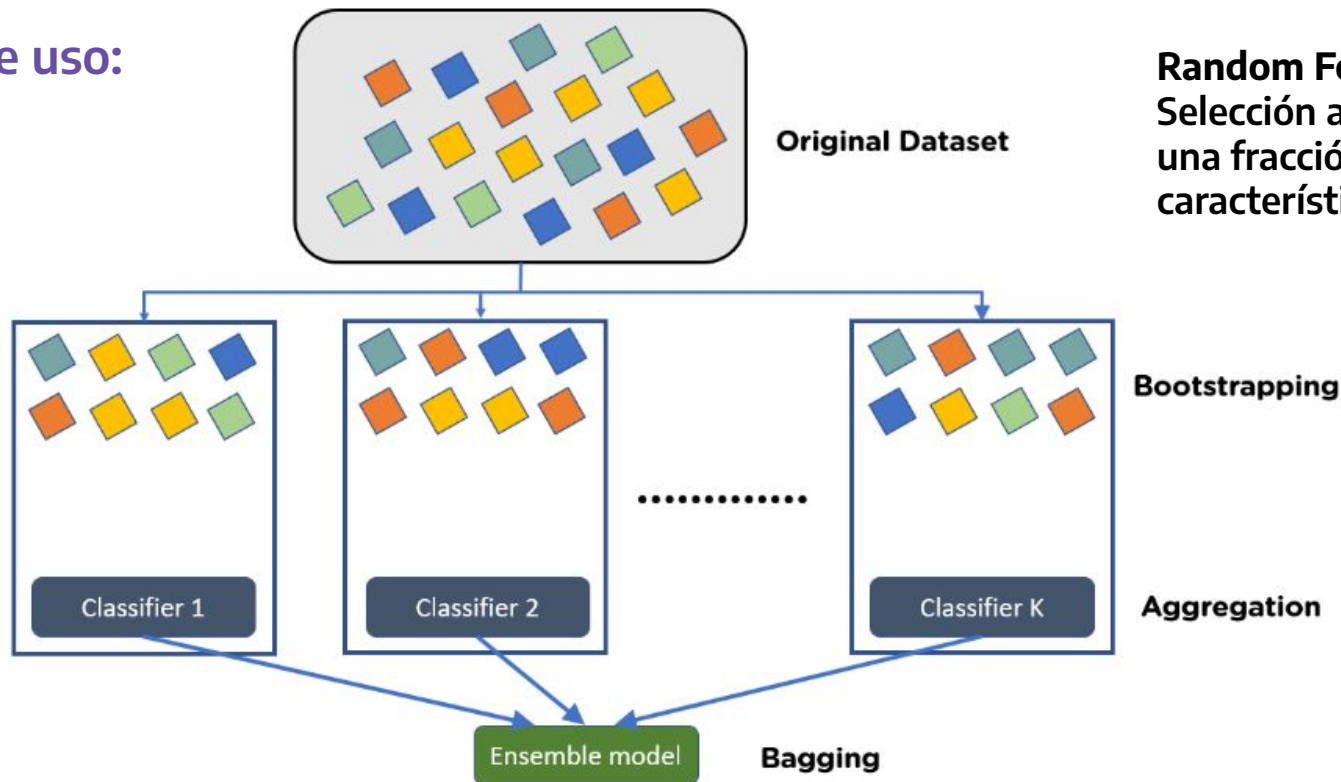


M<sup>th</sup> Avg. Height

# Bagging: bootstrap aggregation

Ejemplo de uso:

*bagging*



**Random Forest:**  
Selección aleatoria de  
una fracción de las  
características

# Bootstrap aggregating - Bagging

Error del comité

$$E_{\text{COM}} = \frac{1}{M} E_{\text{AV}}.$$

Advertencia: solo válido si

$$\begin{aligned} \mathbb{E}_{\mathbf{x}} [\epsilon_m(\mathbf{x})] &= 0 \\ \mathbb{E}_{\mathbf{x}} [\epsilon_m(\mathbf{x})\epsilon_l(\mathbf{x})] &= 0, \quad m \neq l \end{aligned}$$

(eso nunca ocurre exactamente)

Nota: en general los errores de los comités no son independientes

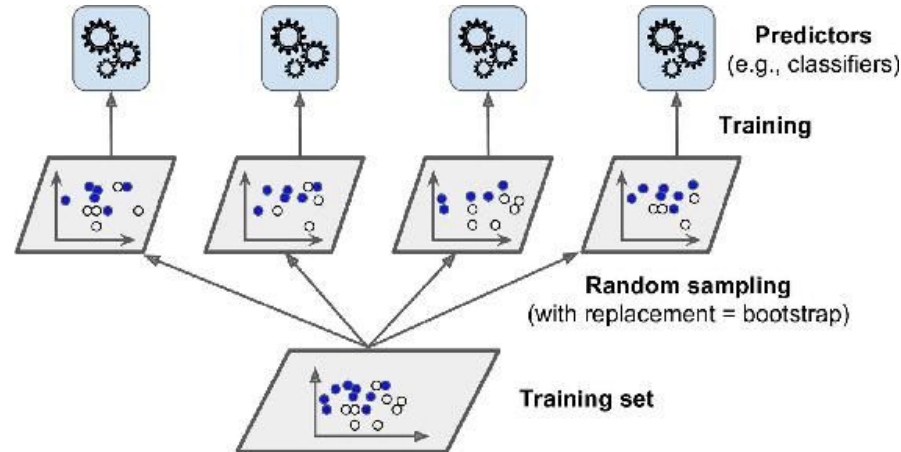
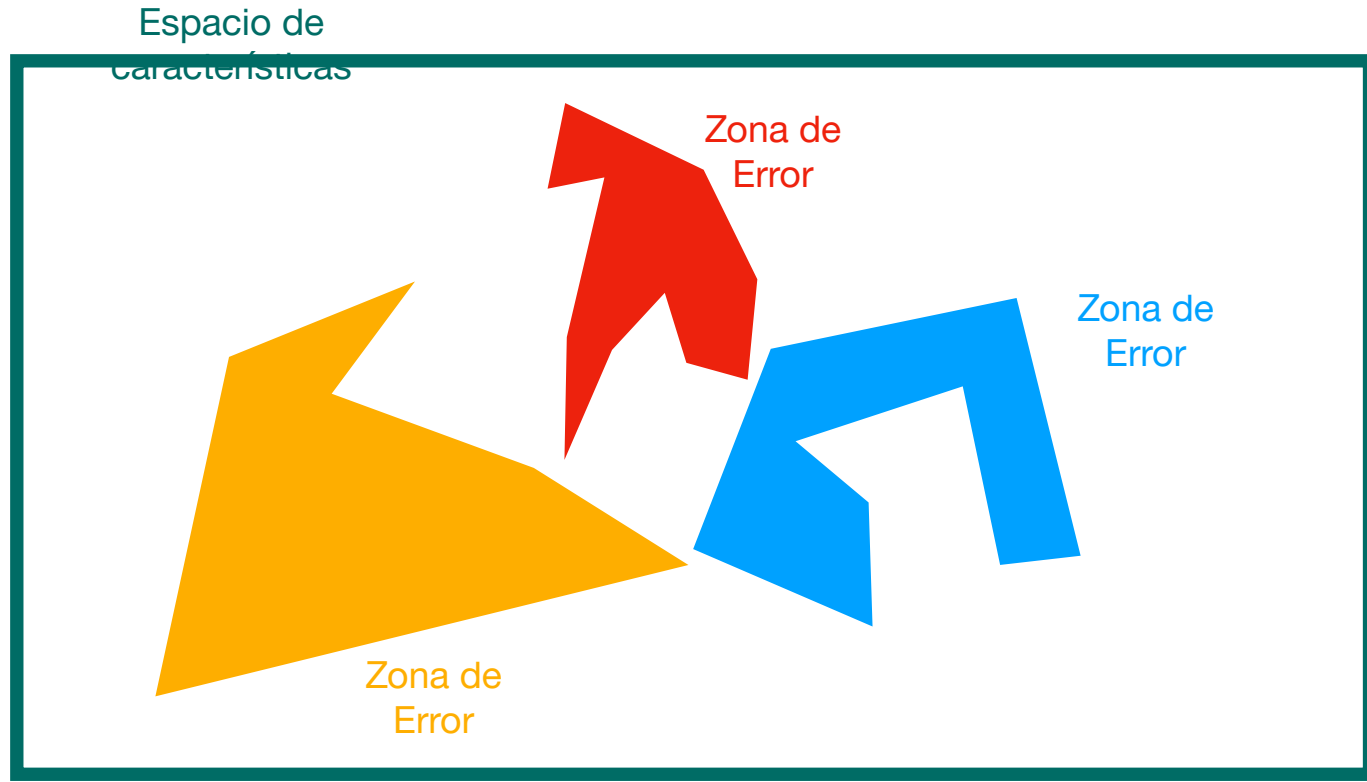


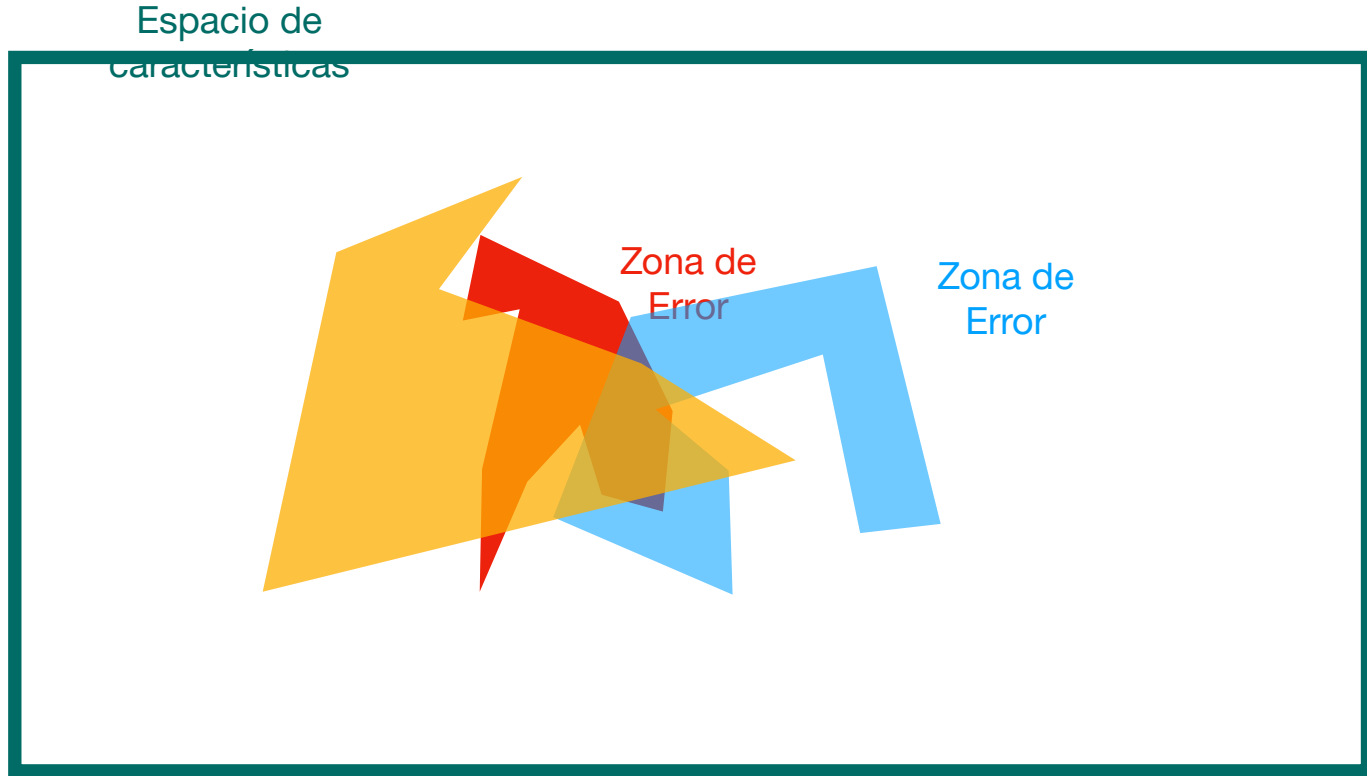
figura de Géron

# *Ensembles*: combinación de modelos para mejorar el desempeño



Inspirado en diapos de A. Farall

# *Ensembles*: combinación de modelos para mejorar el desempeño



Inspirado en diapos de A. Farall

# Bosques aleatorios / Random Forests: bagging de árboles de decisión

RandomForest  
Classifier

BaggingClassifier

DecisionTree  
Classifier

es un

usando

RandomForest  
Regressor

BaggingRegressor

DecisionTree  
Regressor

como estimador de base,  
pero

*bootstrap*  
`max_samples = 1`

y los árboles de decisión  
tienen `split =`  
`'random'`

# Bosques aleatorios / Random Forests: bagging de árboles de decisión

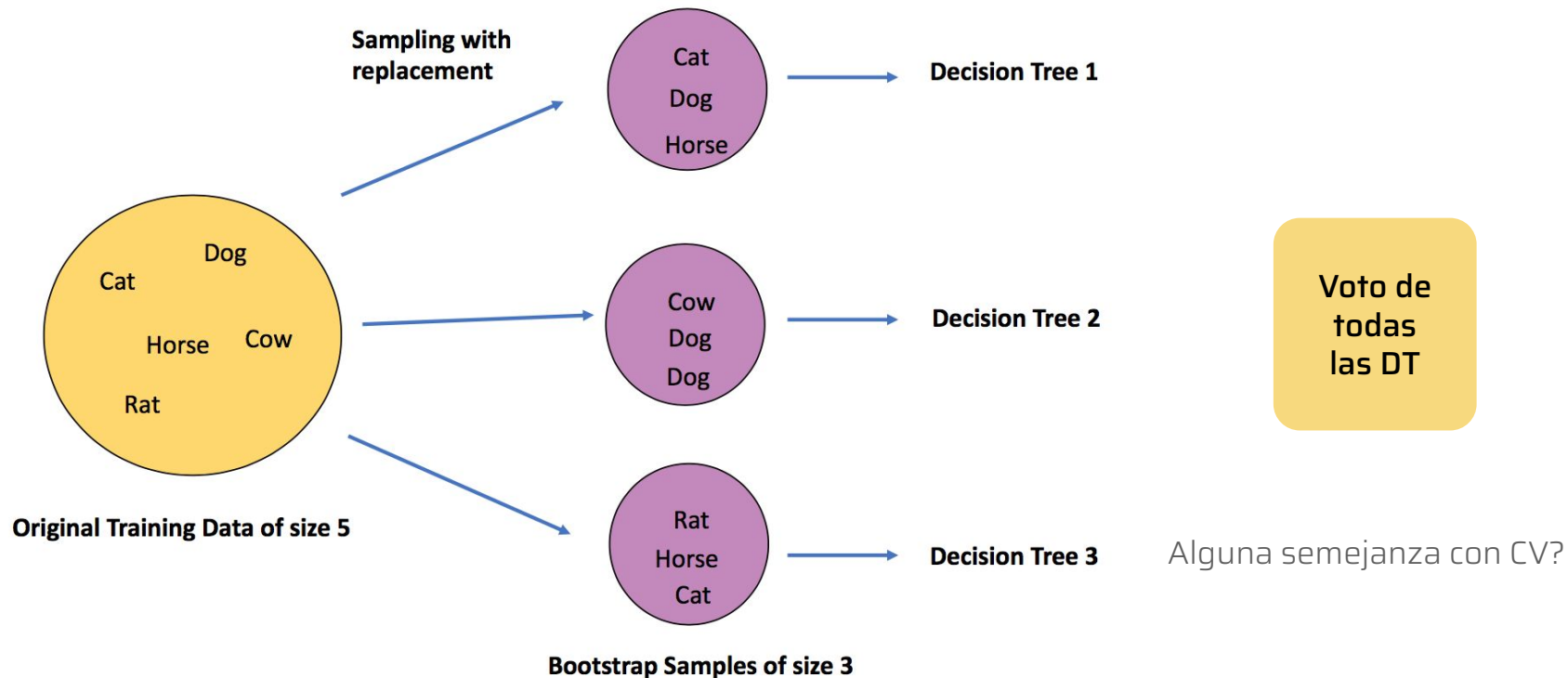


figura de Navnina Bhatia

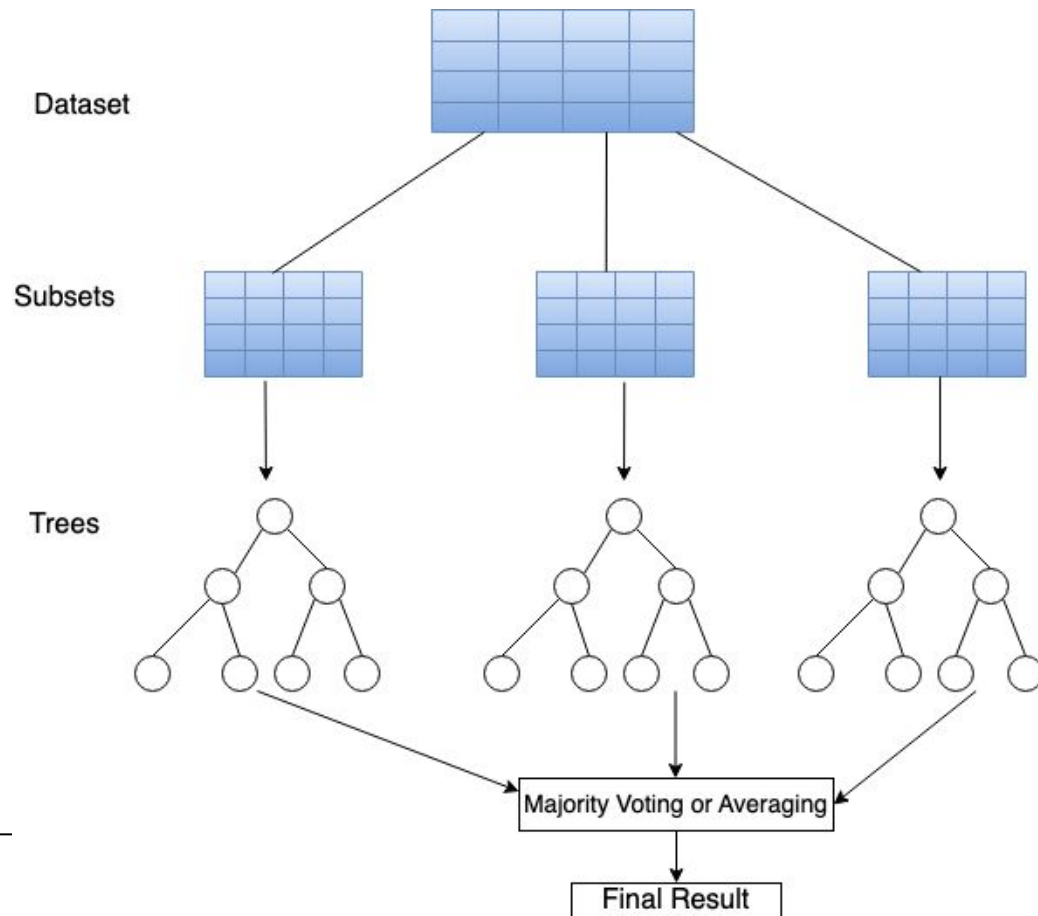


# ExtraTrees x Bosques Aleatorios

Los algoritmos Extra Trees construyen múltiples árboles de decisión sobre todo el conjunto de datos.

Los Bosques Aleatorios (Random Forests) construyen múltiples árboles de decisión sobre subconjuntos de datos con reemplazo

Además, en los Bosques Aleatorios se elige el mejor nodo para dividir mientras que en los Extra Trees se realiza una aleatorización en la división de nodos.



# Out-of-bag score (oob-score)

- Para cada DT, usar los datos que quedaron afuera de la muestra (out-of-the bag) como si fueran datos no vistos para hacer predicciones
- Para un dato, usar el voto de todas las DT que no lo tienen y hacer un promedio
- Se calcula el score de la clasificación final de toda la muestra de out-of-the bag

## Diferencia con score en el subconjunto de test

- Los datos de test nunca fueron usados en el entrenamiento del modelo x lo que es ***out-of-the bag*** para una DT no lo es, necesariamente, para otra
  - La evaluación con el test se hace usando el modelo completo, en este caso, todas las DT x el ***oob*** solo usa las DT que no vieron ese dato
  - Distinto también de CV (permite además, calcular un error)
  - De cualquier forma, los resultados son semejantes!
- (Curiosidad: 1/e de los datos está disponible como ***oob***)

Vamos al notebook!

**Notebook\_Semana\_2\_Ensembles.ipynb**


# Métricas de clasificación

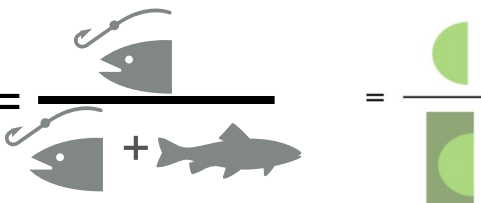
Matriz de  
confusión

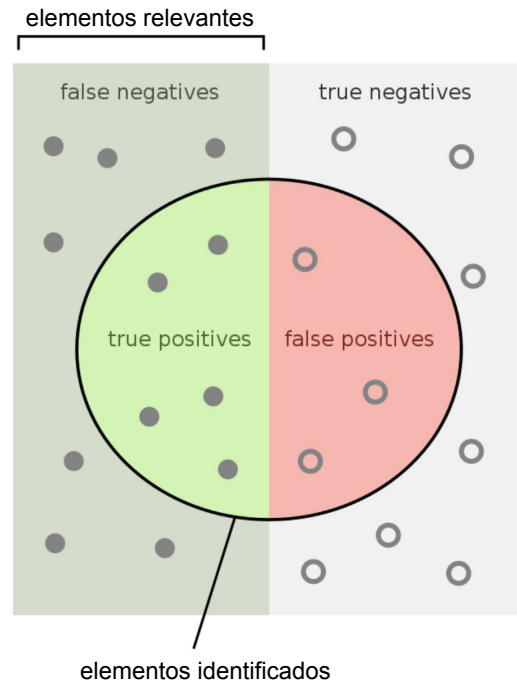
$$\begin{pmatrix} TN & FP \\ FN & TP \end{pmatrix}$$

elementos relevantes

Métricas

$$\text{precision} = \frac{TP}{TP + FP}$$


$$\text{recall} = \frac{TP}{TP + FN}$$




Taza de éxitos

$$\text{accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

Probemos eso:

**Notebook\_02\_RandomForests.ipynb**



**Argentina  
programa  
4.0**

---



**Universidad  
Nacional  
de San Martín**



**Escuela de  
Ciencia y Tecnología  
ECyT\_UNSAM**



**Secretaría de Economía  
del Conocimiento**