

PROGRAMA DE  
ESPECIALIZACIÓN ANALÍTICA

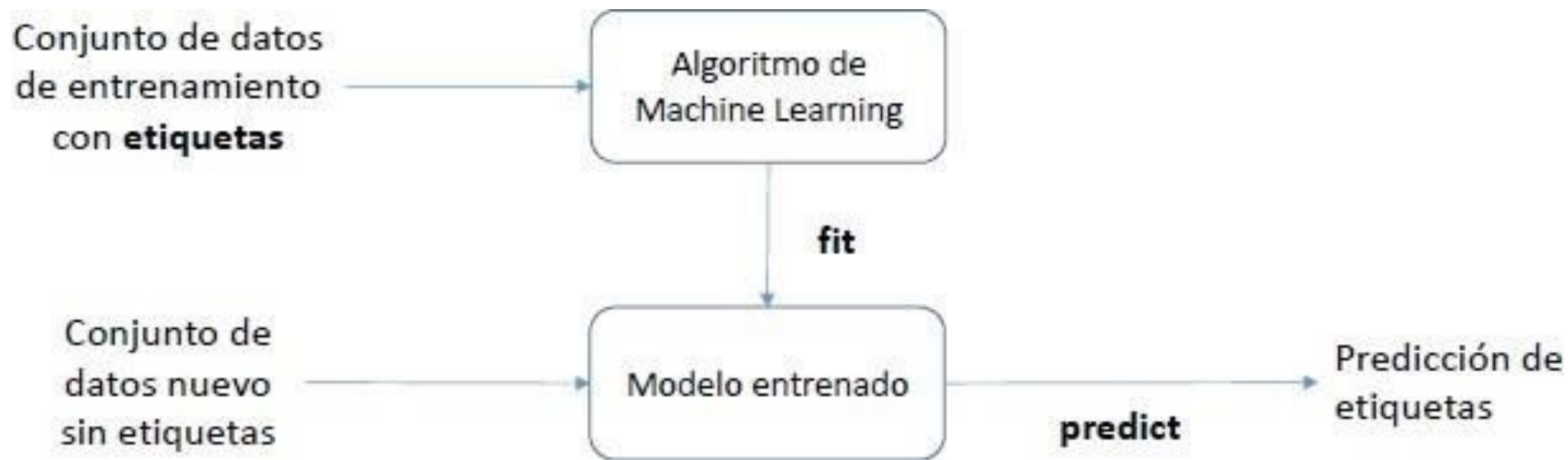
# ADVANCED DATA SCIENCE

## Métodos Supervisados

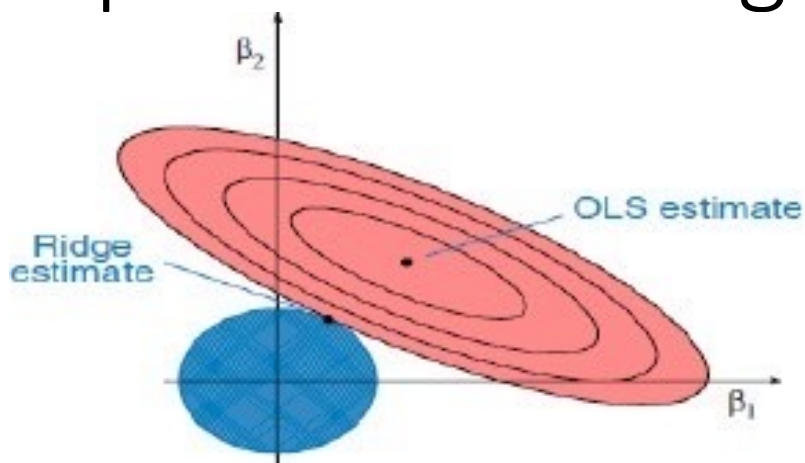
### SESIÓN 3

Docente: Brian Alarcon

## Entendiendo los Métodos Supervisados



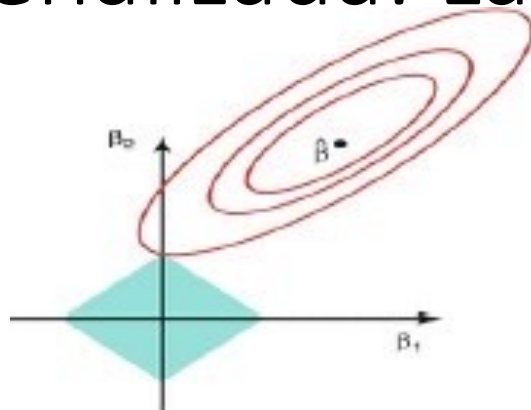
# Modelo de regresión penalizada: Ridge



$$\underbrace{\sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij})^2}_{\text{Mínimos cuadrados}} + \underbrace{\lambda \sum_{j=1}^p \beta_j^2}_{\text{Penalización ridge}}$$

- Es útil cuando existe colinearidad entre las variables utilizadas para el entrenamiento.
- El factor  $\lambda$  (lambda) sirve para controlar la intensidad de la regularización. Se utiliza la norma de regularización L2.
- La elección de este parámetro involucra un balance entre los componentes de sesgo y varianza del error cuadrático medio al estimar  $\beta$ .

# Modelo de regresión penalizada: Lasso



$$\underbrace{\sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij})^2}_{\text{Mínimos cuadrados}} + \lambda \underbrace{\sum_{j=1}^p |\beta_j|}_{\text{Penalización lasso}}$$

- Permite reducir a valores cercanos a cero los coeficientes de las variables menos relevantes.
- Se utiliza para la reducción de la dimensionalidad.
- El factor  $\lambda$  (lambda) controla el nivel de regularización.
- Lasso es una técnica de regresión lineal regularizada, como Ridge, con la leve diferencia en la penalización. (Norma L1 en lugar de L2)

# Modelo de regresión penalizada: Lasso

- Para valores crecientes de  $\lambda$ , los coeficientes  $\beta_j$  se contraen hacia cero como en Ridge (shrinkage), con la diferencia de que algunos de ellos se anulan.
- Esto es, Lasso produce estimación y selección de variables en forma continua y simultánea, siendo especialmente útil en el caso  $p \geq n$ .
- En los últimos años se han presentado algunas generalizaciones y extensiones de las técnicas presentadas anteriormente, especialmente diseñadas para ciertas situaciones particulares.

# Modelo de regresión penalizada: elastic net

Es una combinación de Ridge y Lasso. Se decide, que peso se le da a cada método de penalización y se implementa la regresión.

$$\underbrace{\sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij})^2}_{\text{Mínimos cuadrados}} + \underbrace{\lambda_1 \sum_{j=1}^p \beta_j^2}_{\text{Ridge}} + \underbrace{\lambda_2 \sum_{j=1}^p |\beta_j|}_{\text{Lasso}}$$

donde  $\lambda$  es un parámetro de precisión

$$\hat{\beta}^e = \arg \min_{\beta} \sum_{i=1}^n (y_i - x_i^T \beta)^2 + \frac{1-\alpha}{2} \lambda \sum_{i=1}^p \beta_i^2 + \alpha \lambda \sum_{i=1}^p |\beta_i|$$

- Si  $\lambda = 0$ , regresión lineal tradicional ( $\hat{\beta}^e = \hat{\beta}$ ).
- Si  $\lambda = \infty$ ,  $\hat{\beta}^e = 0$
- Si  $\alpha = 0$ , entonces  $\hat{\beta}^e = \hat{\beta}^{ridge}$
- Si  $\alpha = 1$ , entonces  $\hat{\beta}^e = \hat{\beta}^{lasso}$



# Tratamiento de los Datos

## Tratamiento de Datos

## Solución

Missings o Datos Nulos

### Imputación Paramétrica

- Variables Cuantitativas: Mediana

- Variables Cualitativas: Moda o Criterio Experto

### Imputación No Paramétrica

- Imputación por K-nn y RandomForest

Tipos de Datos

Label Encoder - One Hot Encoder

Outliers o Valores

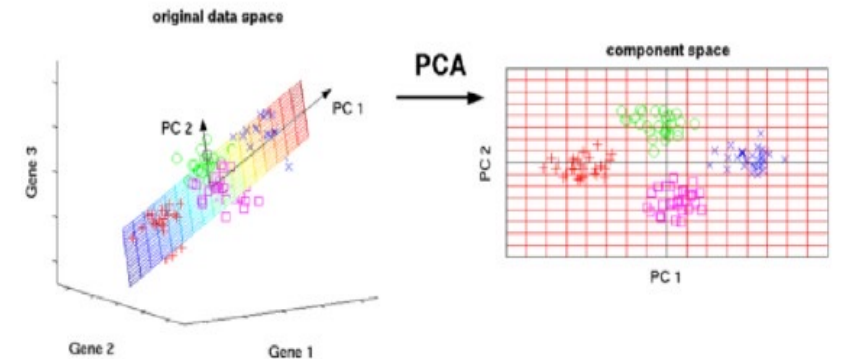
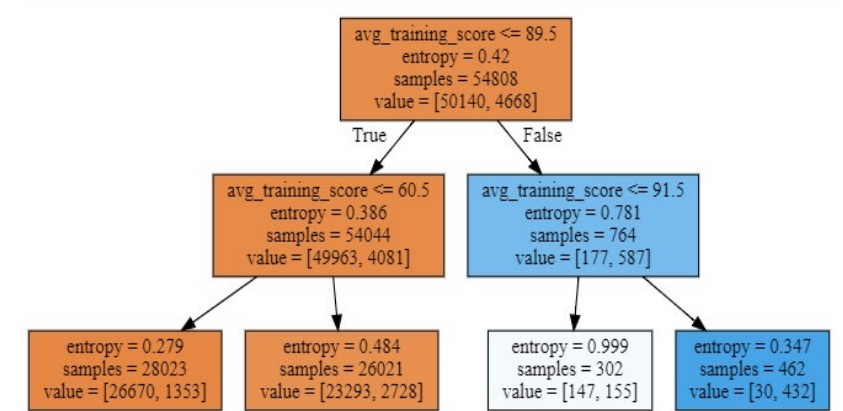
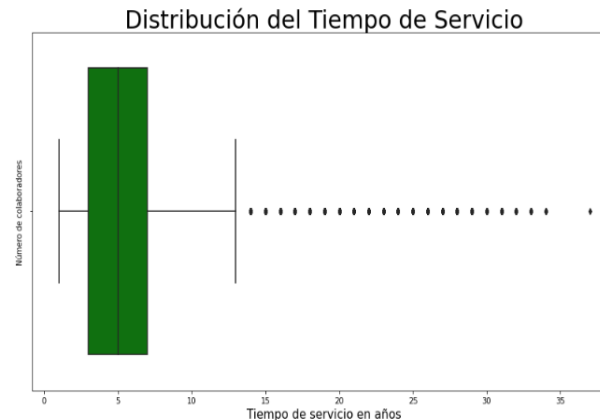
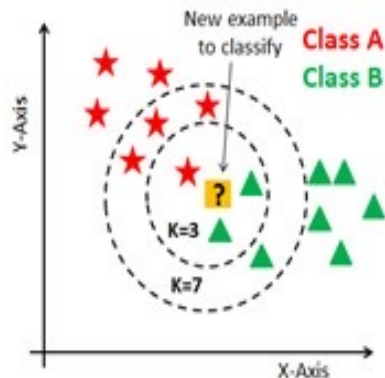
ACP - Análisis Componentes Principales

Discordantes

Recodificación por Árboles de Decisión

Estandarización de Variables

Topear o Recorte por Percentiles



# Missing y representatividad de variables

*Datos de*

			entradas				target

**Problema 1:** En muchos modelos, no se usan los casos con valores faltantes en las variables de entrada (eliminación casewise o por casos)

**Problema 2:** Las fórmulas de predicción no pueden asignar un puntaje a los casos con valores faltantes.



# Imputación de valores faltantes

Missing values

PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	male	22	1	0	A/5 21171	7.25		S
2	1	1	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	female	35	1	0	113803	53.1	C123	S
5	0	3	male	35	0	0	373450	8.05		S
6	0	3	male		0	0	330877	8.4583		Q

Variables Cuantitativas

\* Media o mediana.

Variables Cualitativas

\* Moda



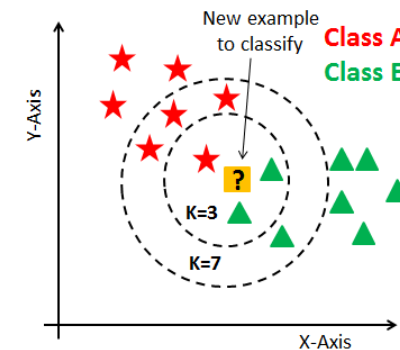
# Imputación de valores faltantes

Missing values

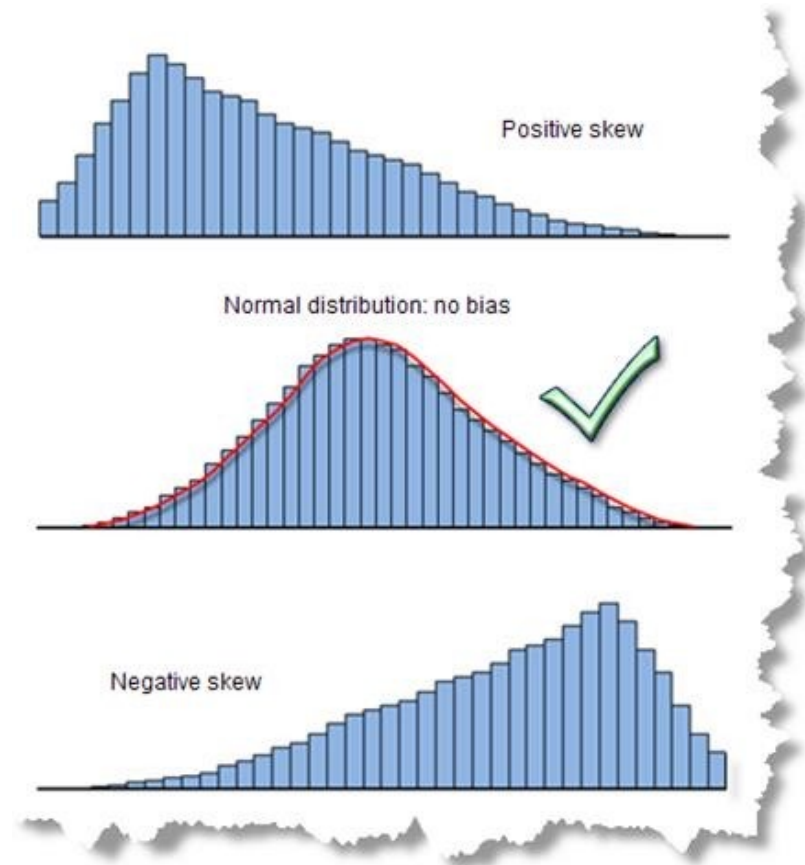
PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	male	22	1	0	A/5 21171	7.25		S
2	1	1	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	female	35	1	0	113803	53.1	C123	S
5	0	3	male	35	0	0	373450	8.05		S
6	0	3	male		0	0	330877	8.4583		Q

Las técnicas más utilizadas para la Imputación de datos son:

- KNN
- RandomForest
- Técnicas de clustering.



# Transformaciones de variables



## Transformaciones Lineales

Valores entre 0 y 1:

$$Z_{if} = \frac{X_{if} - \min(X_f)}{\max(X_f) - \min(X_f)}$$

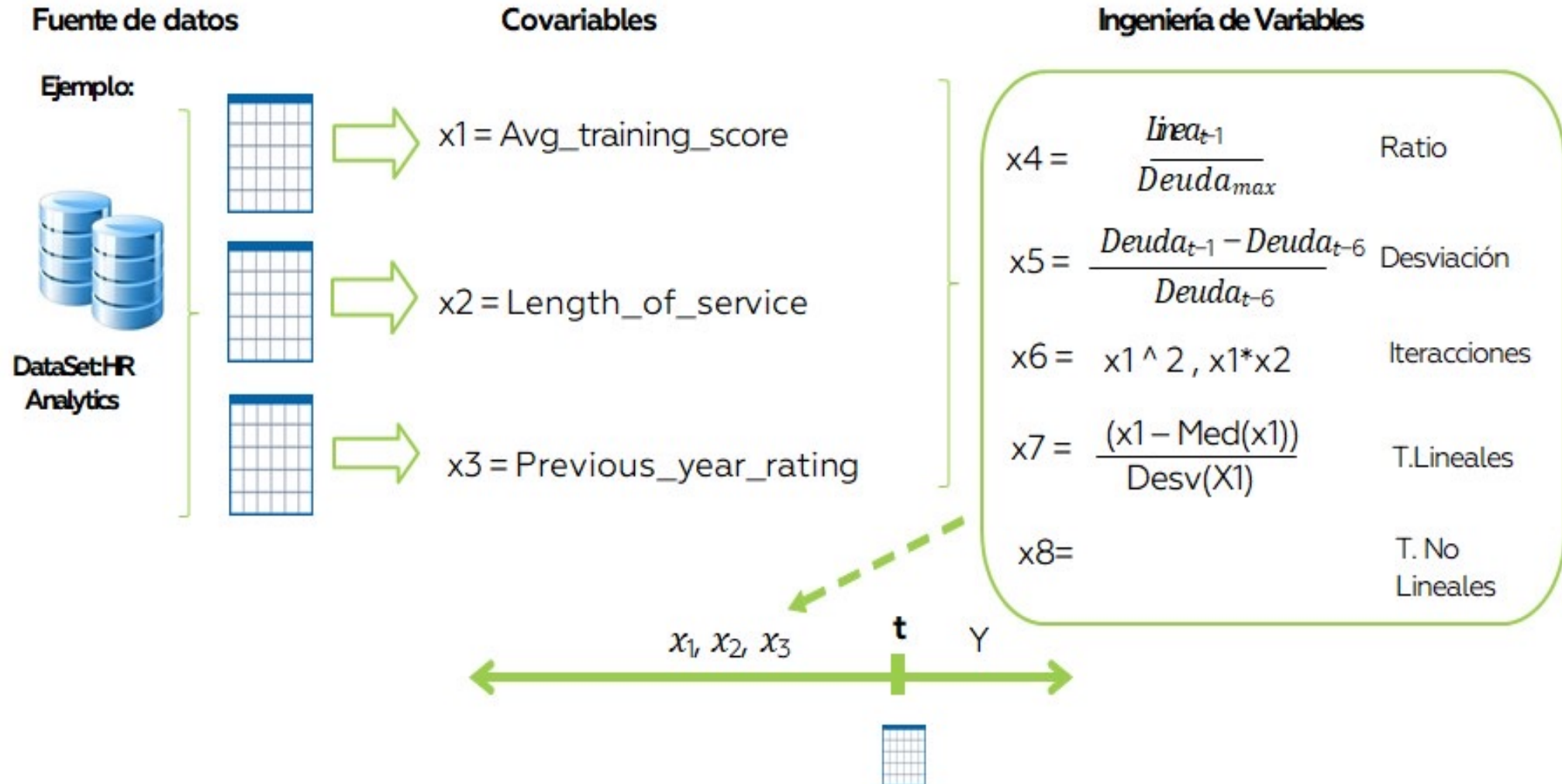
Normalización:

$$Z_{if} = \frac{X_{if} - \mu_f}{\sigma_f}$$

## Transformaciones No Lineales

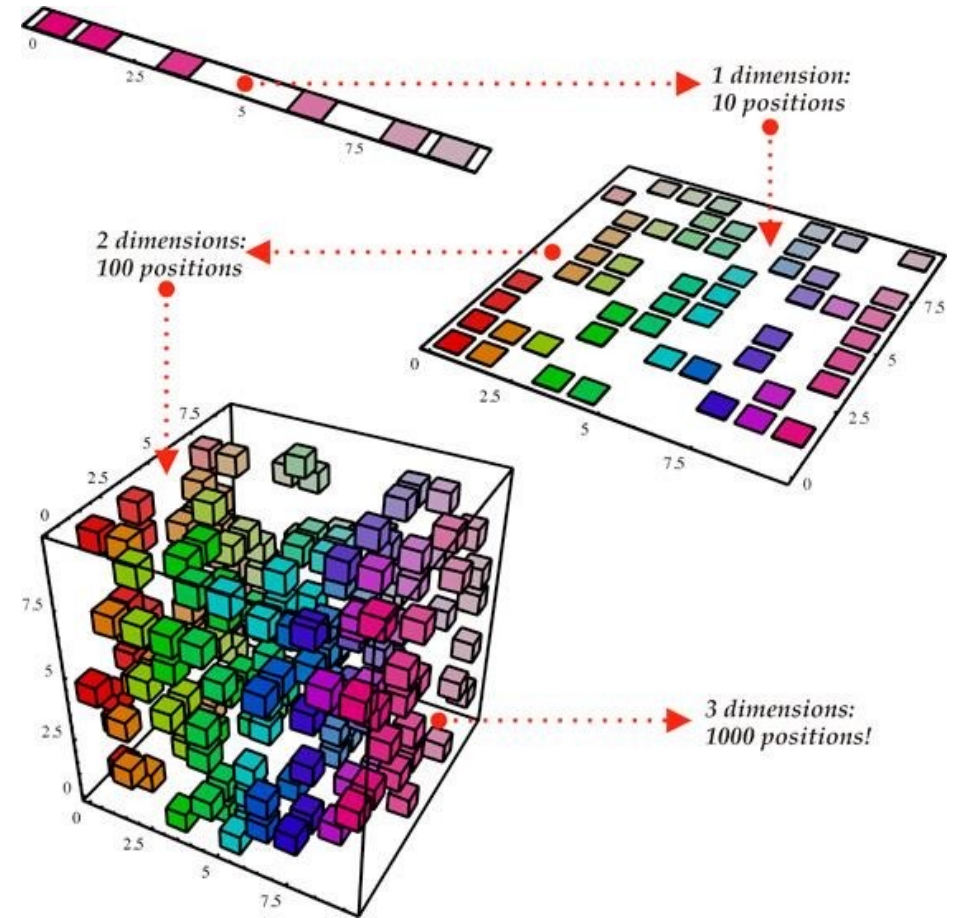
Potencia	Transformación	Descripción
$\lambda_1 = 2$	$Y' = Y^2$	Cuadrado
$\lambda_1 = 1$	$Y' = Y$	Datos sin Transformar
$\lambda_1 = 0.5$	$Y' = \sqrt{Y}$	Raíz Cuadrada
$\lambda_1 = 0.333$	$Y' = \sqrt[3]{Y}$	Raíz Cúbica
$\lambda_1 = 0$	$Y' = \ln(Y)$	Logaritmo
$\lambda_1 = -0.5$	$Y' = \frac{1}{\sqrt{Y}}$	Raíz Cuadrada Inversa
$\lambda_1 = -1$	$Y' = \frac{1}{Y}$	Reciproco

# Feature Engineering



# Selección de Variables

- La “Maldición de la Dimensionalidad”: Si tenemos más dimensiones, las combinaciones de los niveles aumentan mucho y se necesitan más observaciones para encontrar todas las combinaciones posibles.
- Mientras más variables se consideren, hay más posibilidades que las observaciones tengan diferencias entre sí.
- De la misma manera, a más variables, el tiempo de procesamiento de los algoritmos y la complejidad de los modelos tienden a aumentar.



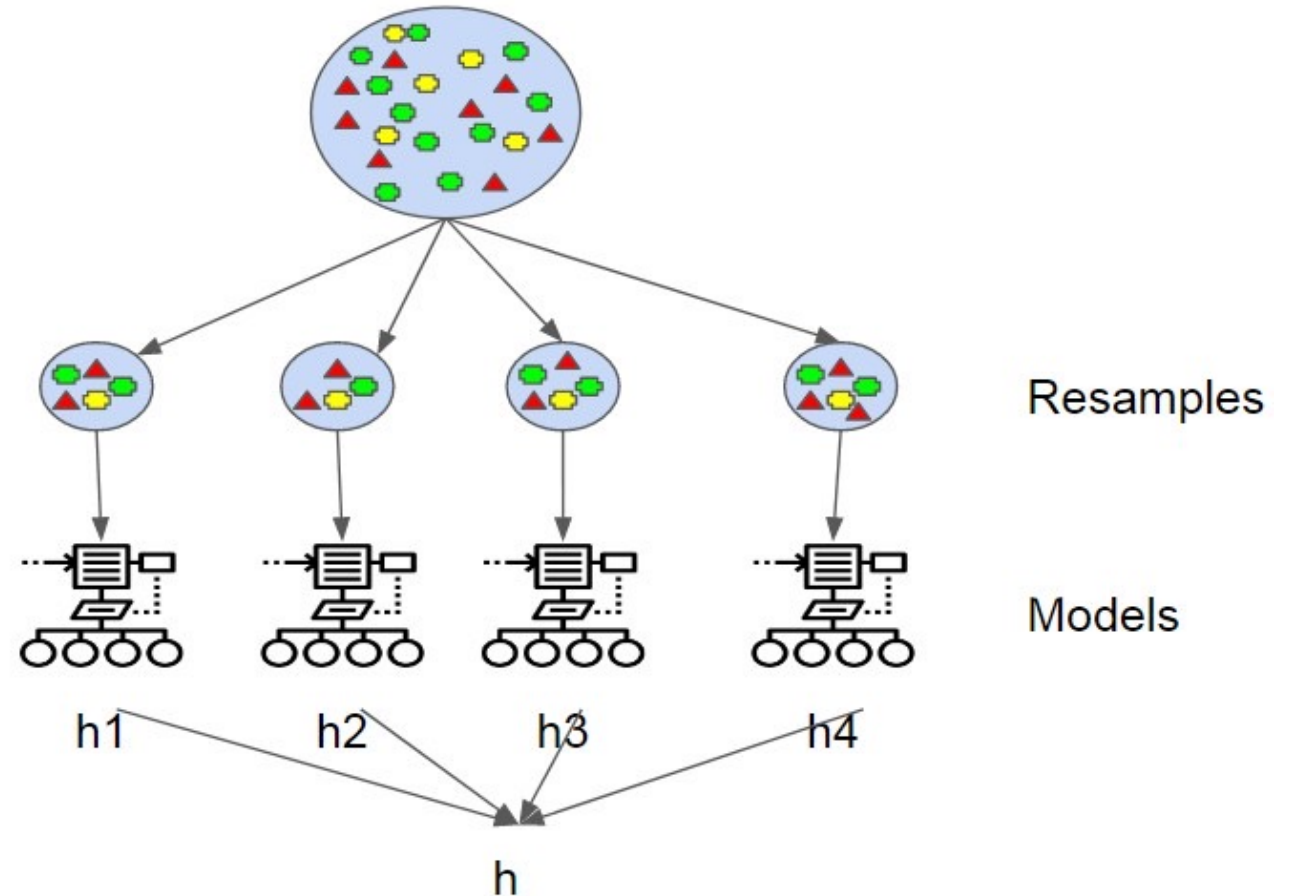
# Métodos de Selección

- Existen varios métodos para filtrar variables:
  - Eliminar las que sean una combinación lineal de otras.
  - Eliminar las que tienen correlaciones muy altas con otras.
  - Eliminar las que tienen una moda con +95% de frecuencia relativa.
  - Eliminar las que tienen muchos valores faltantes (missings).
  - Eliminar las categóricas que tienen una excesiva cantidad de valores diferentes, posiblemente no aporten.
- Y otros para seleccionar variables relevantes:
  - Variables con pocos missings o ninguno.
  - Variables categóricas con pocos niveles.
  - Variables cuantitativas con pocos valores extremos.
  - Variables con correlación o asociación alta con el Target
  - Variables con un alto indicador de Importancia
  - Variables con un WOE o IV altos



# Feature Importance - Random Forest

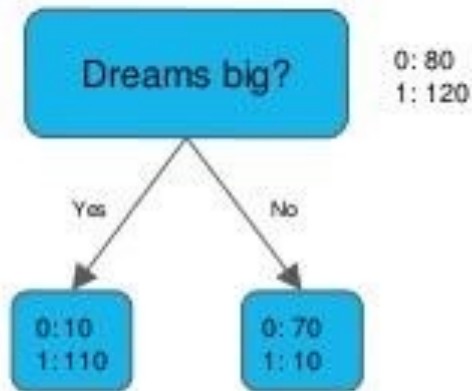
- El Modelo Random Forest es un tipo de modelo construido a partir de varios modelos de Árbol que se ensamblan a través de sumarizar sus predicciones.
- Este algoritmo nos ofrece la posibilidad de identificar el nivel de importancia de cada variable considerada en el entrenamiento del modelo, de manera que podamos quedarnos sólo con aquellas que sean más relevantes.





## Feature Importance - Random Forest

### Selección de variables con Random Forest: Gini



$$GINI(n) = \frac{120}{200} \cdot \frac{80}{200} + \frac{80}{200} \cdot \frac{120}{200} = 0.48$$

$$I(V; n) = 0.48 - 0.22 - 0.15 = 0.11$$

Criterio de split => métrica (Gini, Información, Varianza...)

Gini:  $GINI(n) = \sum_{i=1}^k p_i(1 - p_i)$  donde  $p_i$  es la proporción de la clase  $i$

Importancia de una variable en un split:

$$I(V; n) = GINI(n_{padre}) - GINI(n_{hijo_1}) - GINI(n_{hijo_2})$$

Importancia en el árbol:

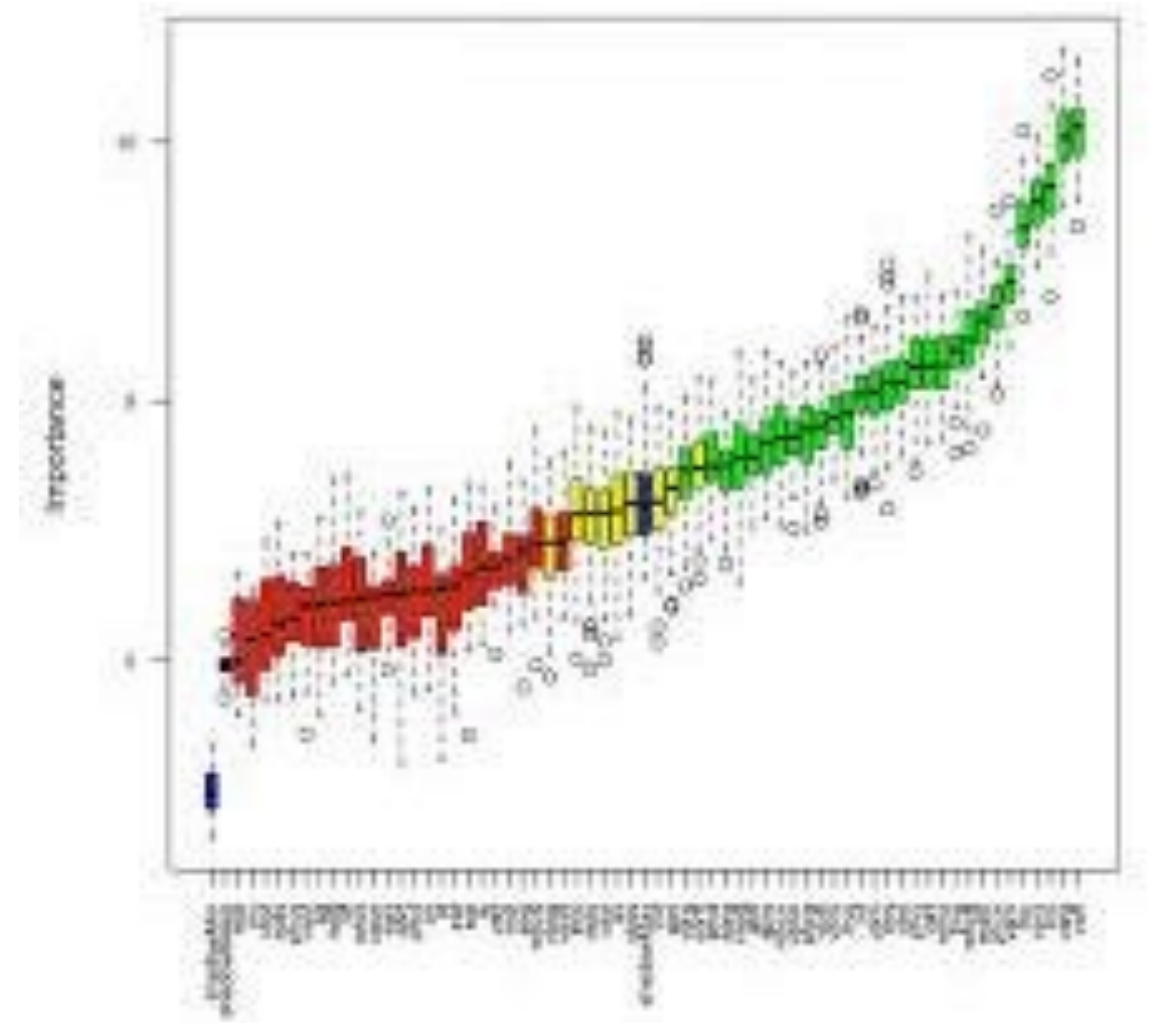
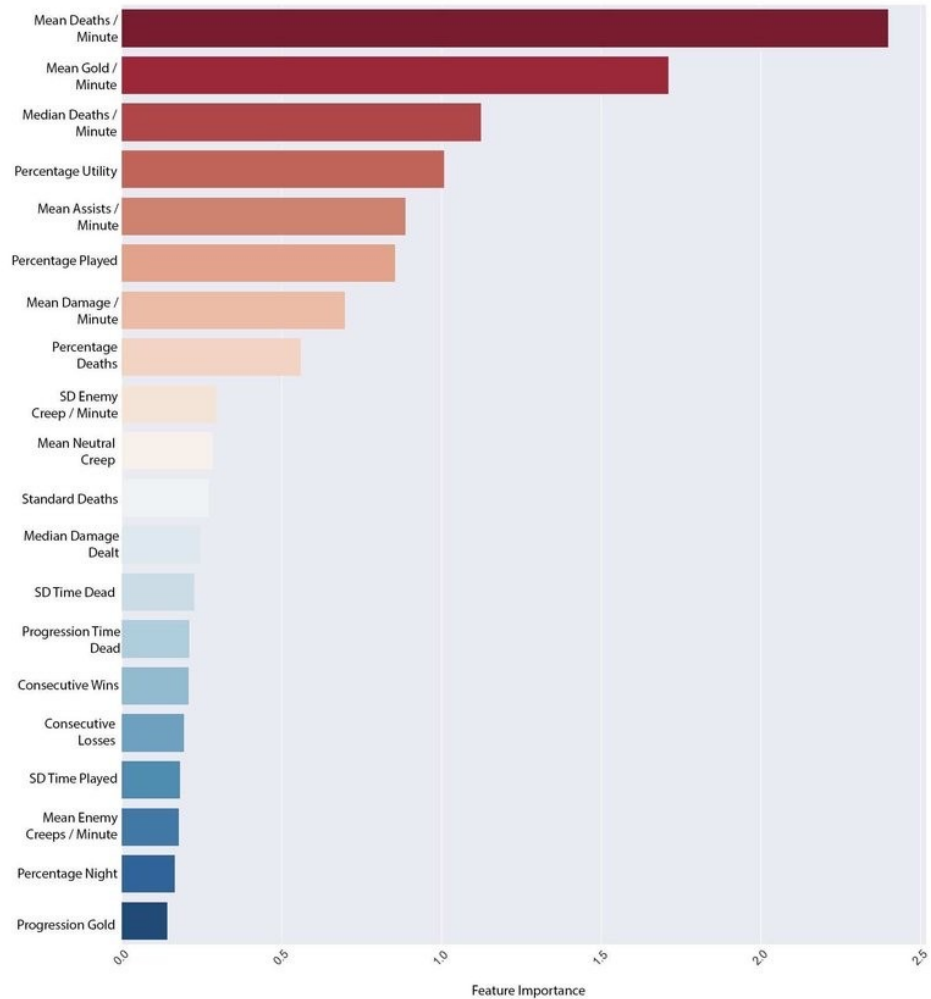
$$I_f(V) = \sum_{n \in N(f)} I(V; n)$$

Importancia en el modelo:

$$I(V) = \frac{1}{\#\mathfrak{F}} \sum_{f \in \mathfrak{F}} I_f(V)$$

Posteriormente, podemos hacer un ranking de variables por importancia para hacer la feature selection

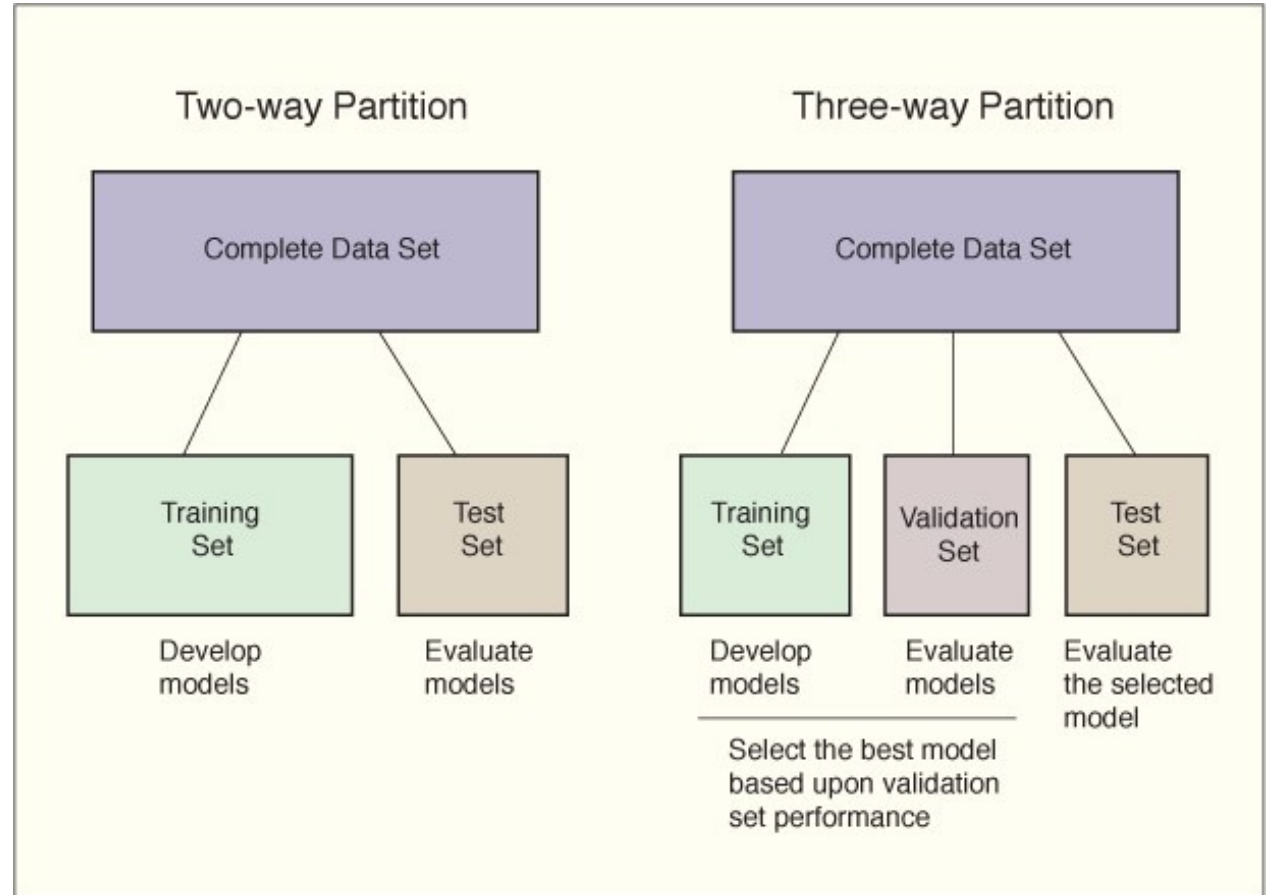
## Feature Importance - Random Forest



# Partición de la Muestra

Es necesario trabajar con muestras diferentes para evitar problemas de sobreajuste.

El método más común tiene que ver con una partición simple, en muestras de Entrenamiento, Validación y Test



# Tipos de Modelos según la Respuesta

Target



Database marketing

=> Compra/No compra ✓



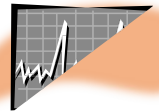
Estimación del Precio

=> ¿Cuánto vale una propiedad?



Detección de fraude

=> Fraude/No fraude ✓



Pronóstico de Ventas

=> ¿Cuánto serán las ganancias?



Detección de patrones

=> ¿Qué letra es?

# Regresión Logística

## DEFINICIÓN:

- Es un modelo **predictivo supervisado**.
- La regresión logística es un modelo de elección discreta en el que la variable **dependiente es cualitativa**. Para los ejemplos, nos vamos a concentrar en casos donde es binaria, pero también existe la Regresión Logística Multinomial, para target con varias categorías.
- Es flexible en cuanto a la naturaleza de las **variables explicativas**, pues éstas pueden ser de **cuantitativas y categóricas**.
- Permite estudiar el **impacto que tiene cada una de las variables independientes** en la probabilidad de que ocurra el suceso de estudio.

# Regresión Logística

Y : La variable Morosidad (Target o Variable objetivo) toma los siguientes valores:

Y : {  
    “1” si el cliente es **moroso**.  
    “0” si el cliente es **no moroso**.

¿Es dicotómica?

¿Es cualitativa?

¿Es mutuamente excluyente?

# Regresión Logística

- 1° Información histórica con la que contamos.



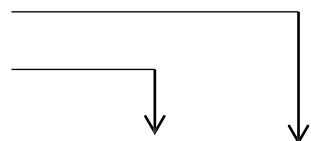
$$p_i = f(\beta_0 + \beta_1 X_1)$$

Se calcula la función logit.

- 2° Hallamos los **valores de los coeficientes** y construimos el **modelo de clasificación**.

$$p_i = \frac{e^{(\beta_0 + \beta_1 X_1)}}{1 + e^{(\beta_0 + \beta_1 X_1)}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1)}}$$

- 3° Con el modelo de clasificación **predecimos una probabilidad y/o clase** y los **comparamos** contra los **valores** que teníamos.



$Y = \text{Yest}$   
 $Y \neq \text{Yest}$



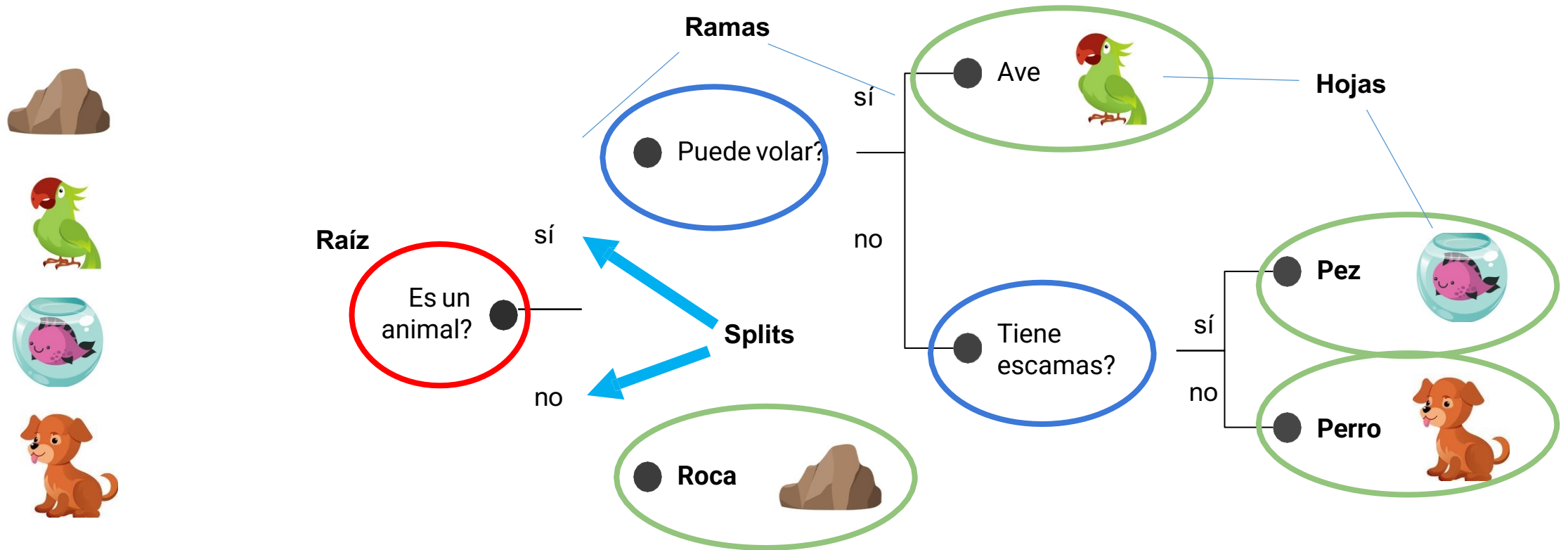
**Acierto**  
Error





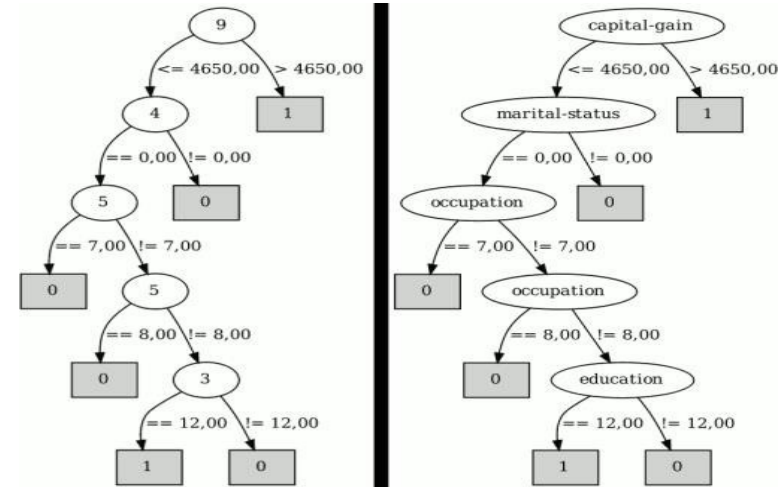
# Árboles de Decisión

- Nacen de una idea sencilla



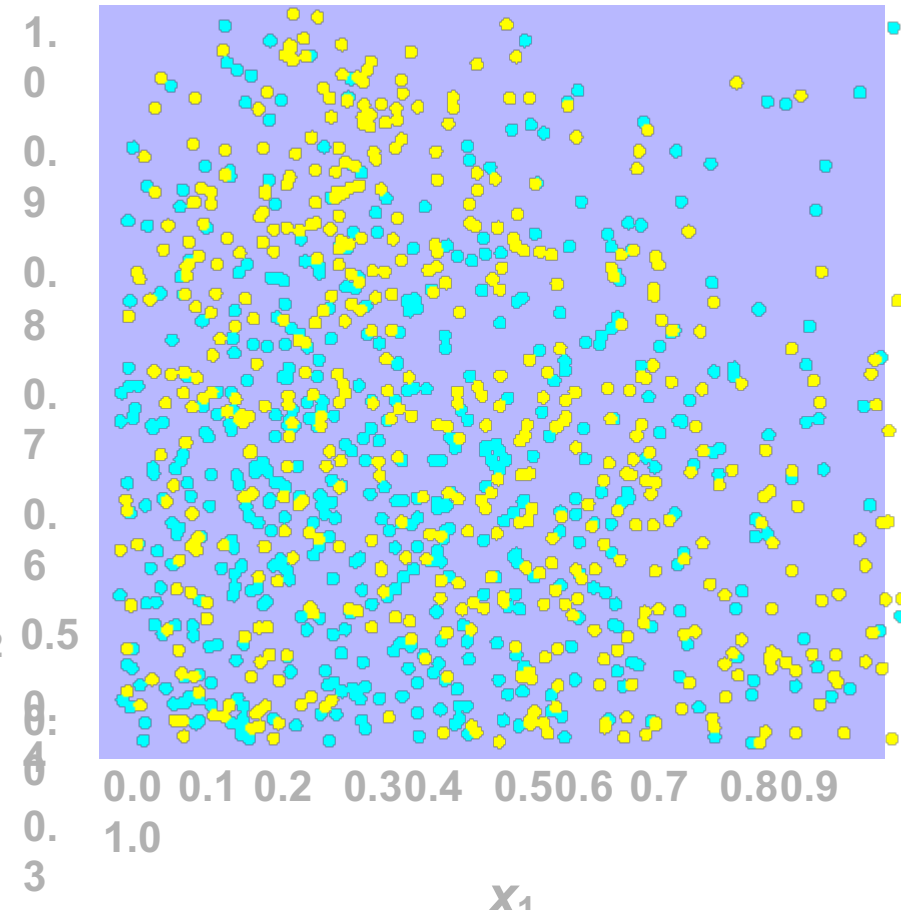
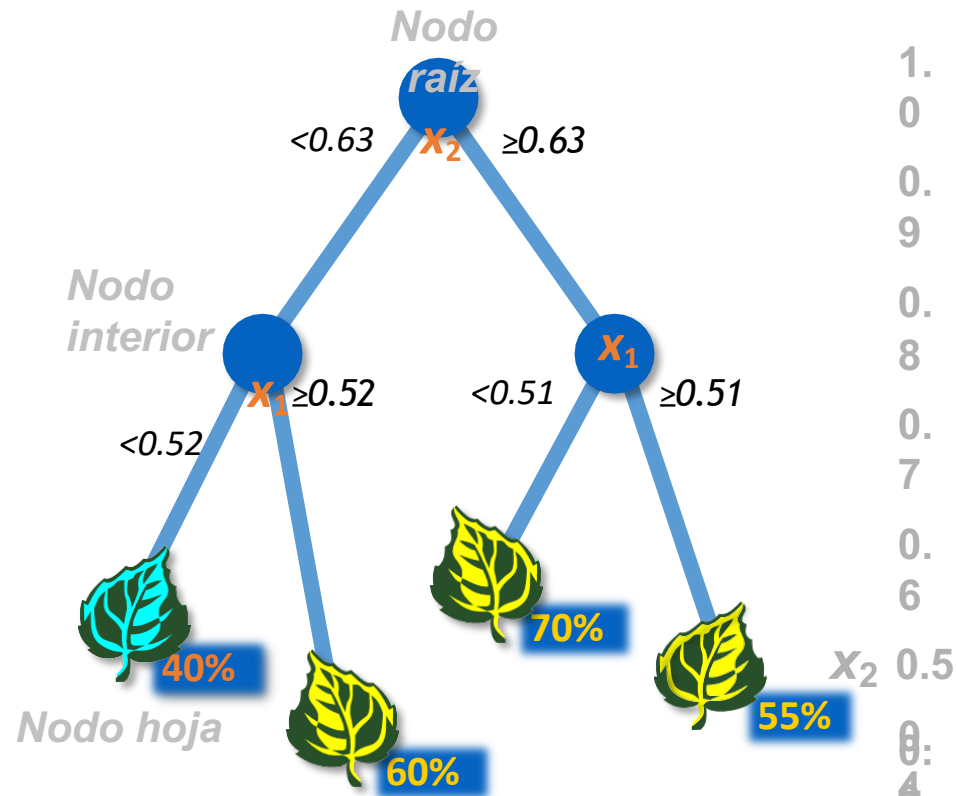
# Árboles de Decisión

- ❑ Árboles de **clasificación**: predicen categorías de objetos.
- ❑ Árboles de **regresión**: predicen valores continuos.
- ❑ Partición binaria recursiva.
- ❑ En cada iteración se selecciona la variable predictiva y el punto de separación que mejor reduzca la 'impureza'.



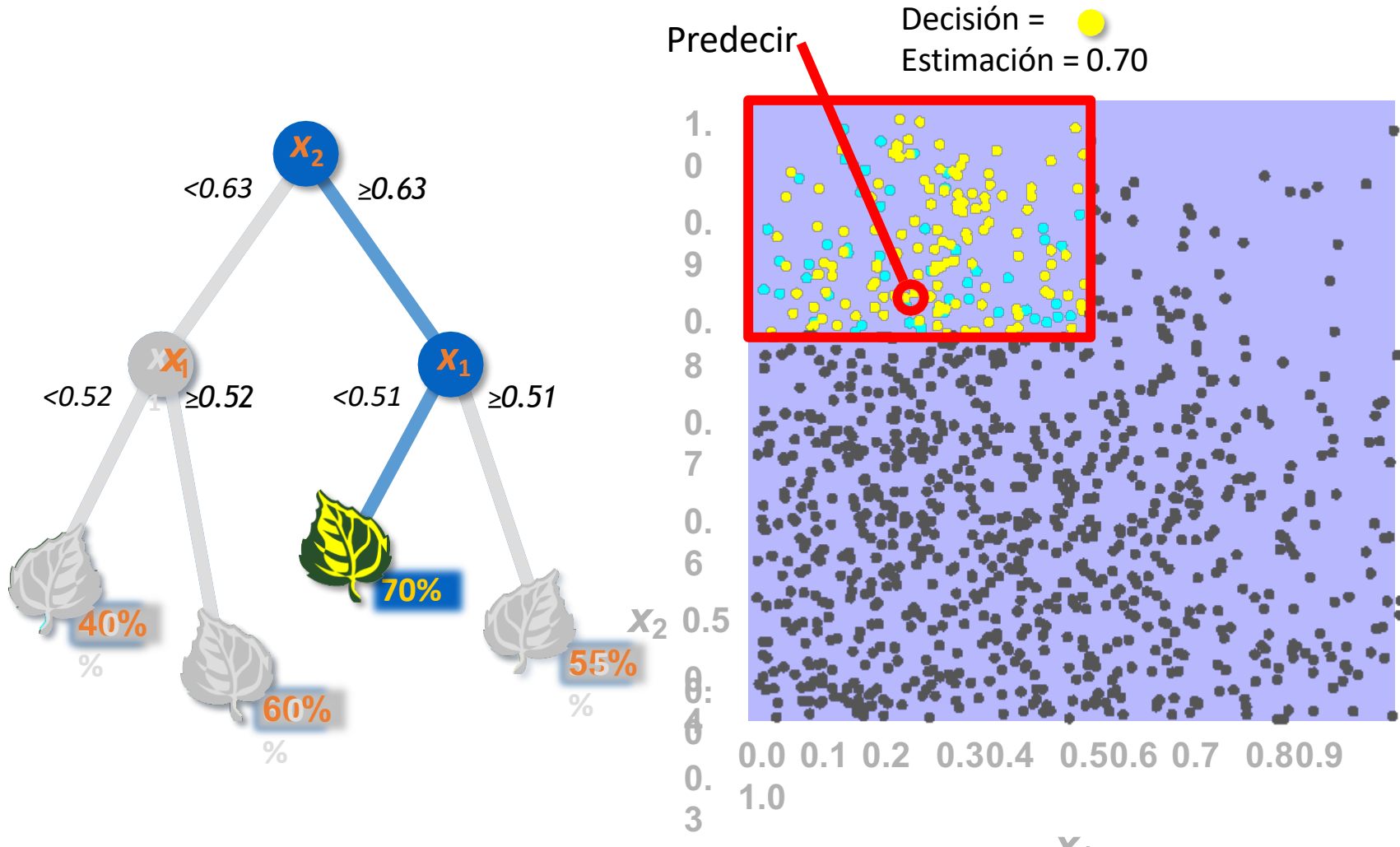
# Árboles de Decisión

Ejemplo: Predicción de una clase (color) utilizando dos variables ( $x_1$  y  $x_2$ )



# Árboles de Decisión

Ejemplo: Predicción de una clase (color) utilizando dos variables ( $x_1$  y  $x_2$ )



# CRITERIOS DE PARTICIÓN

- Cada partición tiene asociada una medida de pureza.
- Se trata de incrementar la homogeneidad de los subconjuntos resultantes de la partición.
- Que sean más puros que el conjunto originario. Existen criterios de impureza tales como :

**Medida de Entropía**

**Índice de Gini**

¿Cómo hallar las mejores separaciones?

$$I_G(p) = 1 - \sum_i p_i^2$$

**Entropy impurity**

$$I_E(p) = \sum_i -p_i \log p_i$$

**Misclasification Error impurity**

$$I_{MC}(p) = 1 - \max_i (p_i)$$

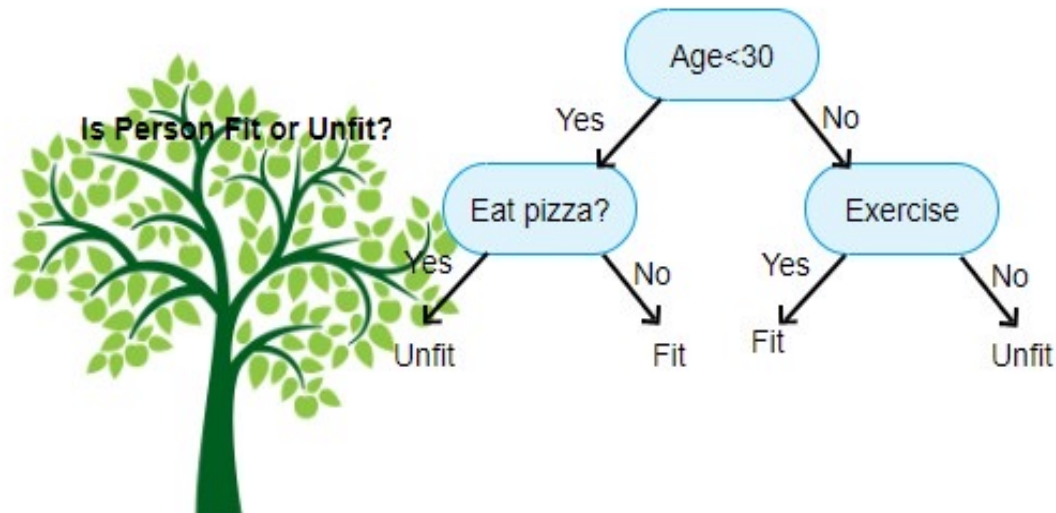
## CRITERIOS DE PARADA

- ❑ Un nodo se declarará terminal si el nodo es puro.
- ❑ Un nodo se declarará terminal si el nodo parental no tiene el mínimo establecido.
- ❑ Un nodo se declarará terminal si cualquier otra subdivisión no da una mejora mayor que la obtenida en el nodo padre.
- ❑ La división del nodo tiene como resultado un nodo hijo cuyo número de casos es menor que el tamaño mínimo preestablecido para un nodo hijo.
- ❑ La profundidad del árbol ha alcanzado su valor máximo preestablecido.

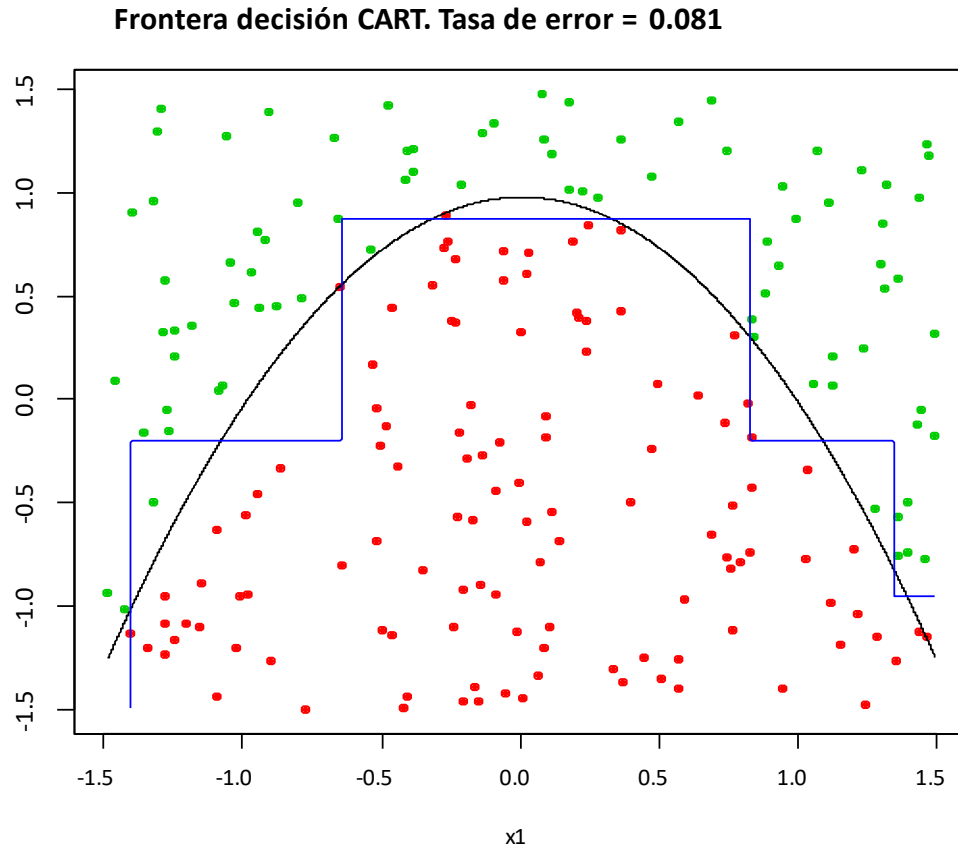


## PODA DE UN ÁRBOL

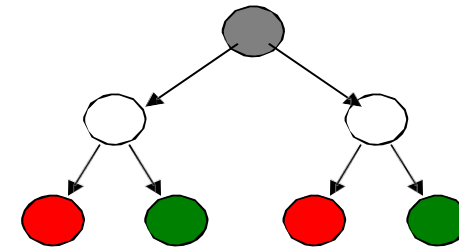
- ❑ En la primera fase, se construye un árbol que tenga muchos nodos.
- ❑ En la segunda fase, el árbol es podado, eliminando las ramas innecesarias hasta dar con el árbol adecuado.
- ❑ Este proceso compara simultáneamente todos los posibles subárboles resultado de podar en diferente grado el árbol original.



# Solución Árbol de Decisión



Solución obtenida  
mediante un Árbol  
**CART** sin podar.



# Pros y Contras

## Ventajas

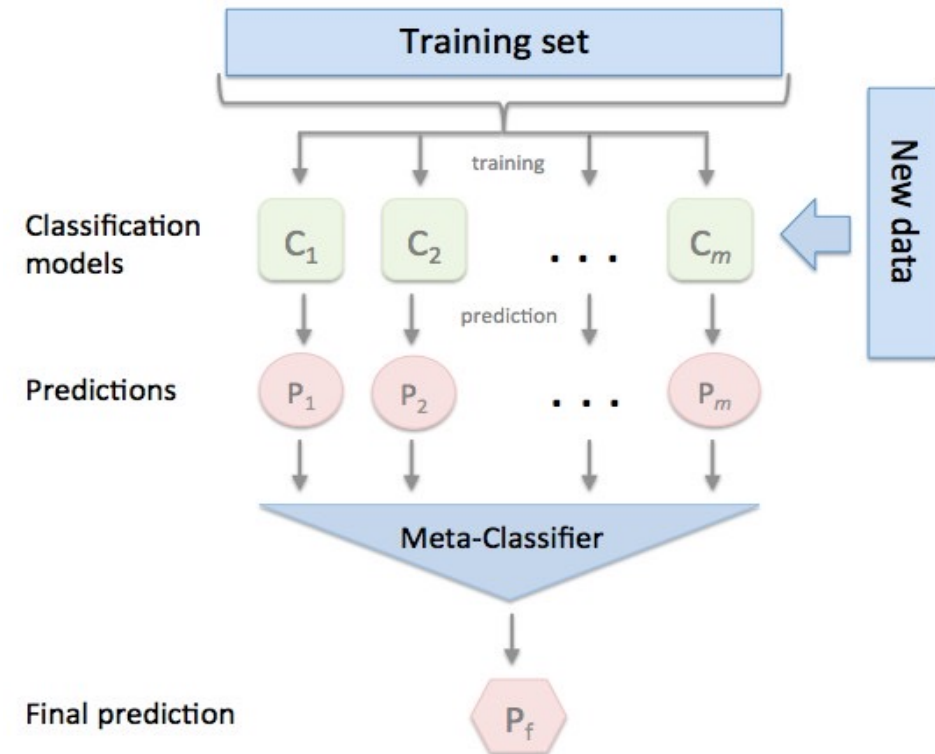
- Se desarrolla muy rápidamente
- No importa si hay valores faltantes en el conjunto de datos
- Descarta variables no importantes
- Fácil de interpretar e implementar

## Desventajas

- Los cortes se suelen hacer en base a variables que tienen muchos niveles
- Se puede caer en el sobreajuste
- Sensibles a cambios en los datos de entrenamiento
- Pueden haber variables que se repiten muchas veces

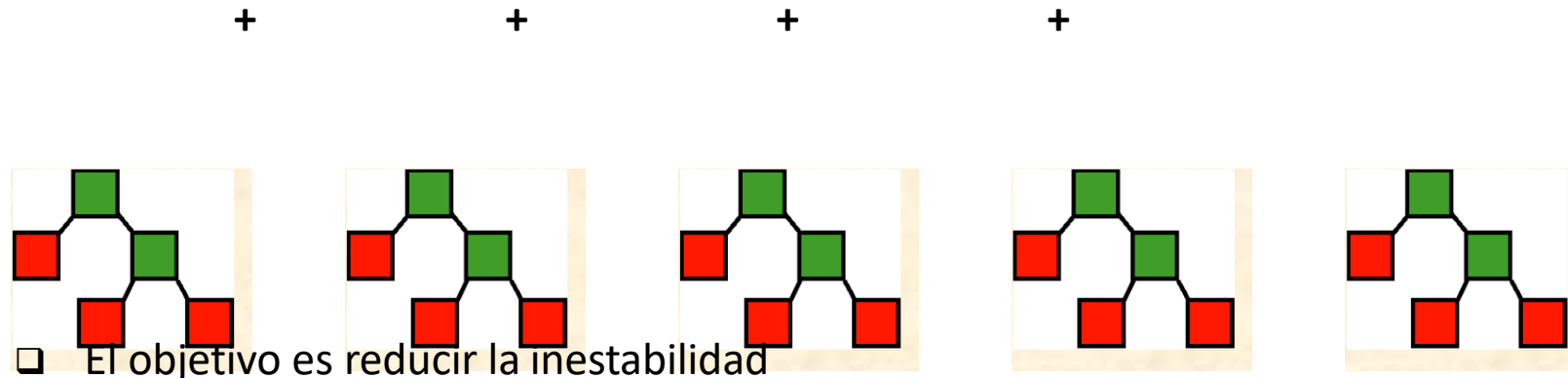
# Métodos de Ensemble

- ❑ La clave para que los métodos de **ensamble** consigan mejores resultados que cualquiera de sus modelos individuales es que, los modelos que los forman, sean lo más diversos posibles.
- ❑ El Ensemble se puede llevar a cabo de distintas maneras:
  - ❑ Voto por mayoría
  - ❑ Promedio
  - ❑ Media ponderada
  - ❑ Etc.



# ¿Qué es bagging?

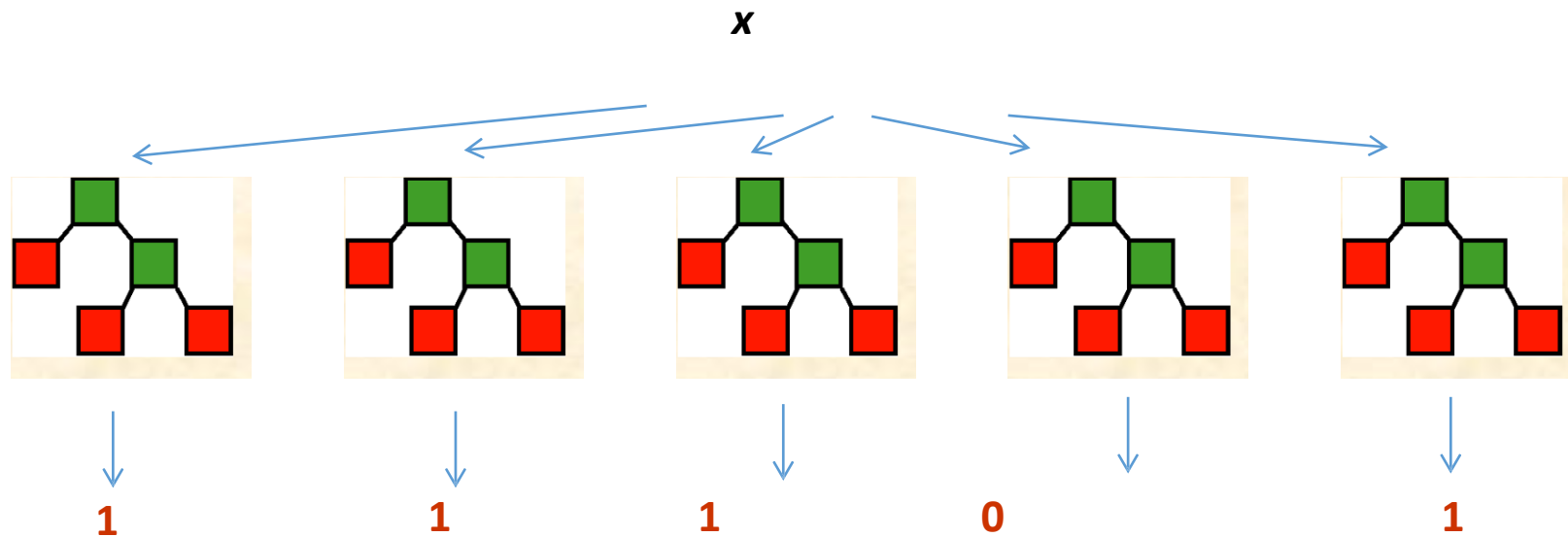
- ❑ Bagging quiere decir bootstrap aggregation. Introducido por Leo Breiman (Berkeley) en 1996
- ❑ La idea es simple. Si tienes las opiniones de un comité de expertos, considéralas todas para tomar una decisión
- ❑ Se extraen muestras bootstrap del conjunto de datos. Para cada muestra, se obtiene un modelo de predicción. El nuevo predictor “bagging” se construye mediante agregación



# Bagging para Clasificación

- Si el problema es de **clasificación** bagging clasificará cada nueva observación por mayoría.

Por ejemplo:

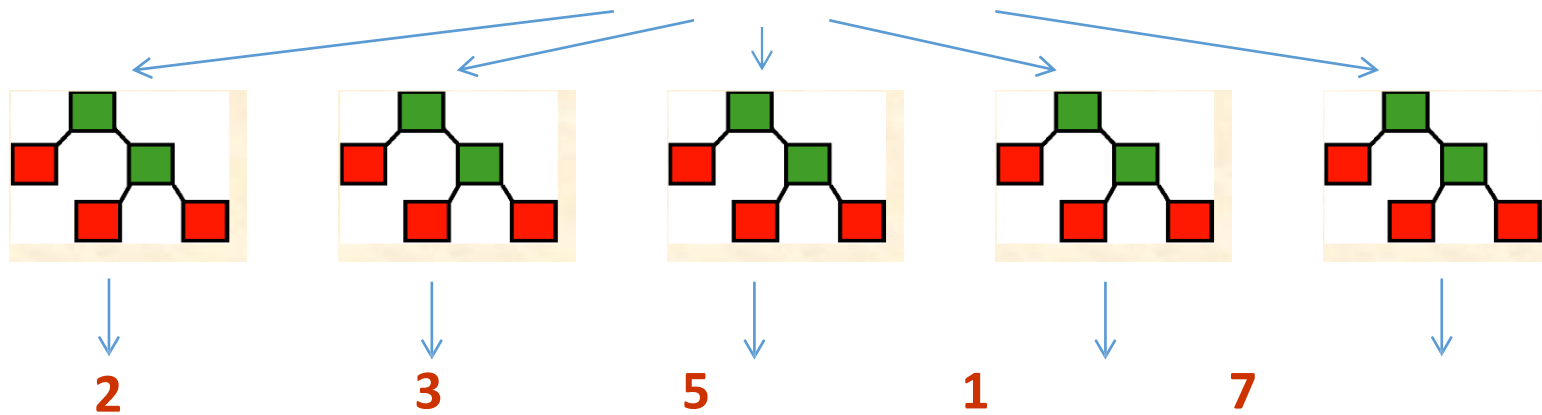


La clase 1 recibió cuatro votos. La clase 0 un voto.  
El predictor bagging clasificará  $x$  en la clase 1.

# Bagging para Regresión

- Si el problema es de **regresión** la predicción bagging se obtiene promediando las predicciones de todos los modelos. Por ejemplo:

$x$



La predicción bagging será:  $(2+3+5+1+7)/5 = 3.6$

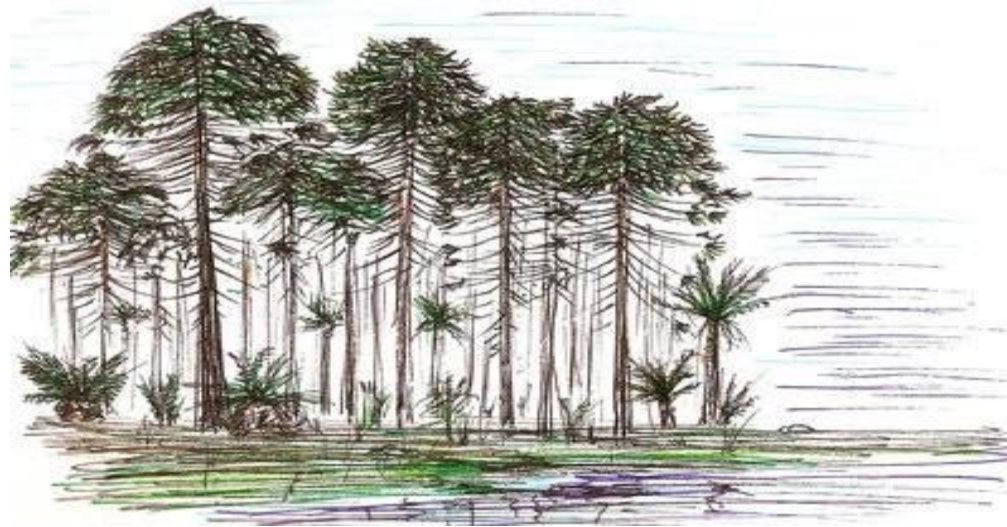
Cuando la variable respuesta es binaria 0/1, el bagging para clasificar por mayoría.

regresión se reduce al criterio de



# Bosques Aleatorios (Random Forest)

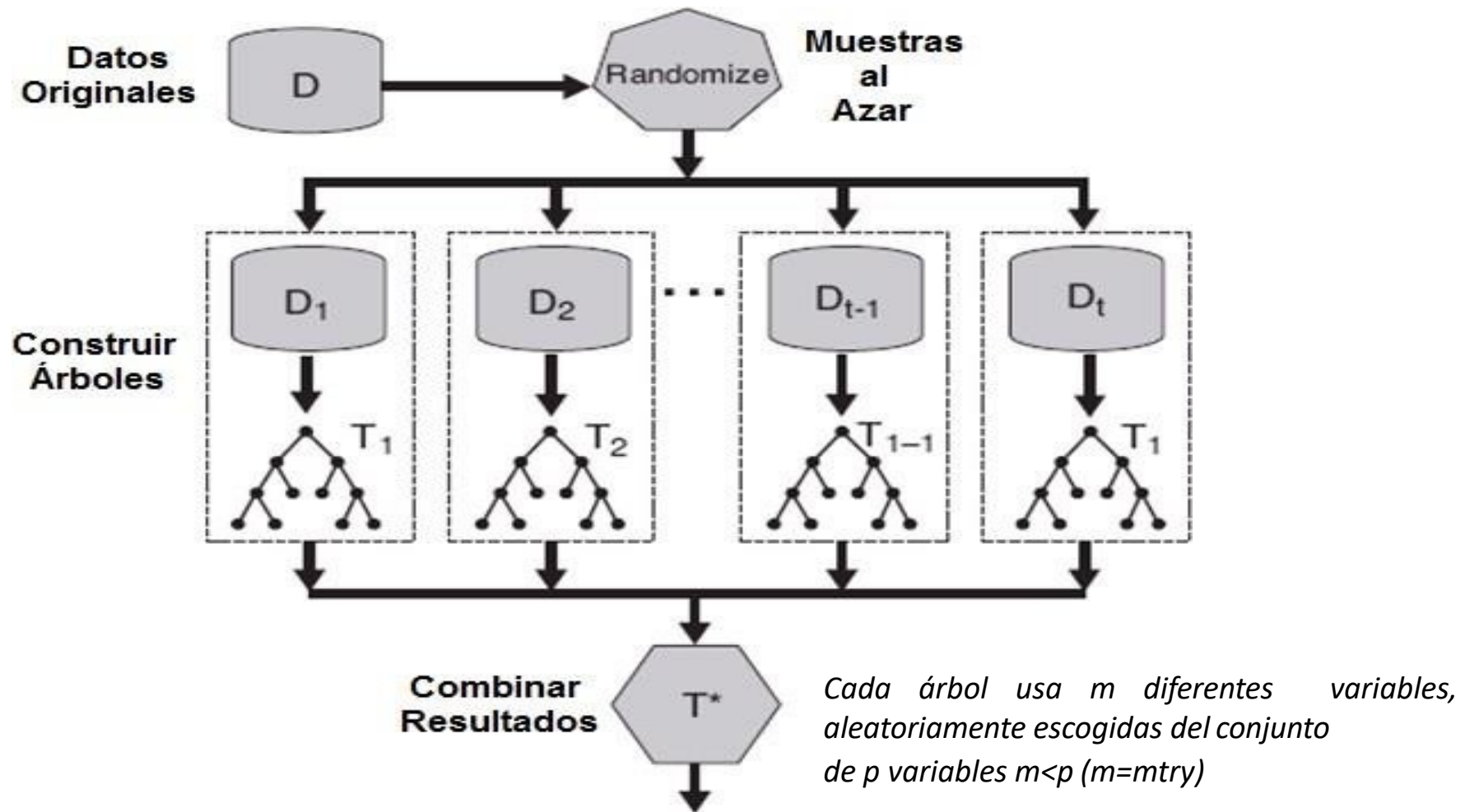
- ❑ El caso en el que todos los clasificadores del Método de Consenso son Árboles dicho método se denomina Bosques Aleatorios (Random Forest).



# Bosques Aleatorios (Random Forest)

- ❑ Desarrollado por Leo Breiman (Berkeley) en 2001
- ❑ Tiene su base en
  - ✓ La predicción con CART
  - ✓ La agregación de modelos de árbol
  - ✓ Bootstrap Aggregation (Bagging)
- ❑ Comercializado por Salford Systems en la herramienta **RandomForests™**.
- ❑ Implementado por Andy Liaw y Matthew Wiener en la librería **randomForest** del entorno R de programación.

# Bosques Aleatorios (Random Forest)

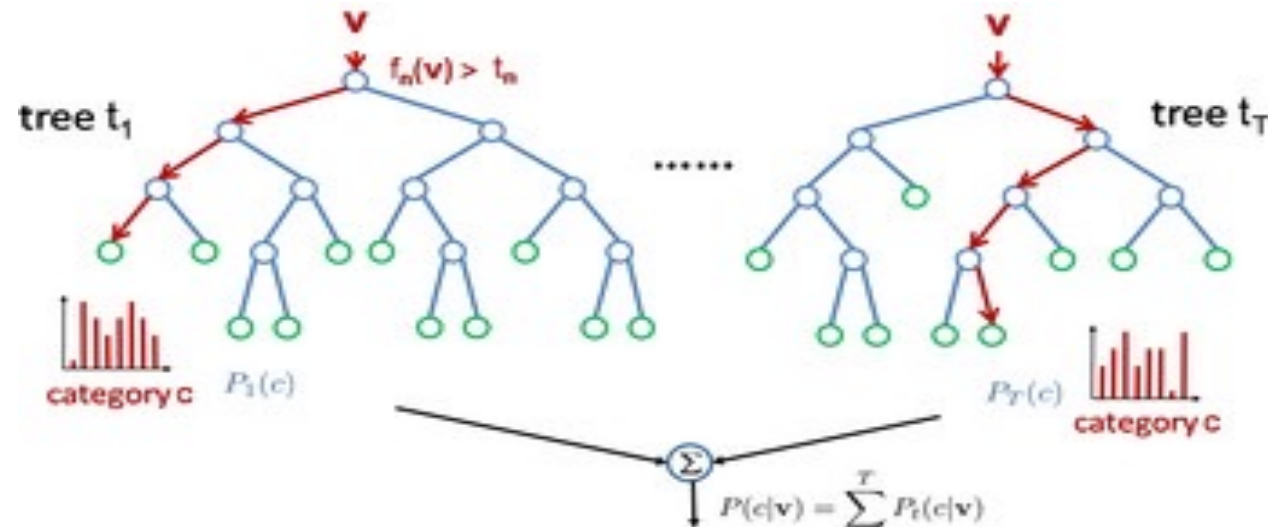


# Métodos de Ensamble: Boosting

- ❑ Consiste en ajustar secuencialmente múltiples modelos sencillos, llamados **weak learners**, de forma que cada modelo aprende de los errores del anterior.
- ❑ Como valor final, al igual que en **bagging**, se toma la media de todas las predicciones (variables continuas) o la clase más frecuente (variables cualitativas).
- ❑ Tres de los métodos de **boosting** más empleados son **AdaBoost**, **Gradient Boosting** y **Extreme Gradient Boosting** (XGBoost).

# Xtreme Gradient Boosting (XGBoost)

- ❑ Tiempo después de la publicación de algoritmo de Gradient Boosting, se le incorporó muchas propiedades avanzadas y de optimización.
- ❑ Como resultado de éstas propiedades nació XGBOOST, uno de los algoritmos más usados en competiciones, en la industria o en problemas específicos de rápida solución.



# Ventajas de XGBoost vs GBM

## ❑REGULARIZACIÓN:

- ✓ La implementación estándar de GBM no tiene regularización como XGBoost, por lo tanto, también ayuda a
  - reducir el sobreajuste.
- ✓ De hecho, XGBoost también se conoce como técnica de "refuerzo **regularizado** ".

## ❑PROCESAMIENTO EN PARALELO:

- ✓ XGBoost implementa el procesamiento paralelo y es **sorprendentemente más rápido** en comparación con GBM.
- ✓ XGBoost también admite la implementación en Hadoop.

## ❑ALTA FLEXIBILIDAD

- ✓ XGBoost permite a los usuarios definir **objetivos de optimización personalizados y criterios de evaluación** .
- ✓ Esto agrega una nueva dimensión al modelo y no hay límite para lo que podemos hacer.

# Ventajas de XGBoost vs GBM

## ❑ MANEJO DE VALORES PERDIDOS

- ✓ XGBoost tiene una rutina incorporada para manejar los valores perdidos.

## ❑ PODA DE ARBOLES:

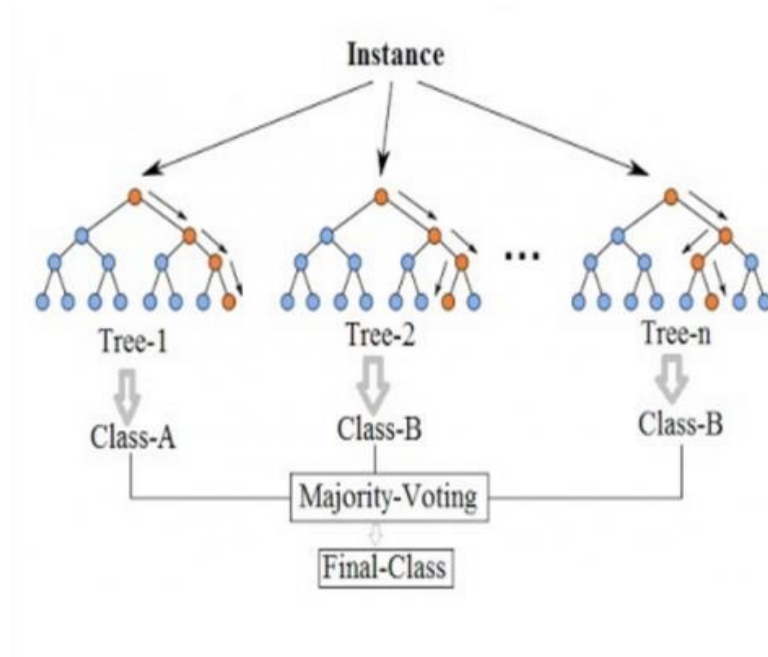
- ✓ Un GBM dejaría de dividir un nodo cuando encuentre una pérdida negativa en la división. Por lo tanto, es más un **algoritmo codicioso**.
- ✓ XGBoost por otro lado hace **divisiones hasta la max\_depth** especificada y luego comienza a **podar** el árbol hacia atrás y eliminar divisiones más allá de las cuales no hay ganancia positiva.

## ❑ CROSS-VALIDATION INCORPORADO

- ✓ XGBoost permite al usuario ejecutar una **validación cruzada en cada iteración** del proceso de refuerzo y, por lo tanto, es fácil obtener el número óptimo exacto de iteraciones de refuerzo en una sola ejecución.

# Parámetros de XGBoost

- ❑ **PARÁMETROS GENERALES:** Afectan a cada árbol individual en el modelo.
- ❑ **PARÁMETROS DE REFUERZO O BOOSTING:** Afectan la operación de **refuerzo** en el modelo.
- ❑ **PARÁMETROS DE APRENDIZAJE DE TAREAS :** Otros parámetros para el funcionamiento general.





# Parámetros de XGBoost

1. **Booster** .- Tipo de modelo a elegir. Árbol o Lineal.
2. **Silent** .- Mensajes o avisos mientras el modelo se ajusta.
3. **Nthread** .-Número de núcleos a usar del sistema.
4. **Eta** .- Ratio de aprendizaje o contribución de cada árbol.
5. **Min\_child\_weight**.- Suma mínima de pesos de las observaciones. Similar a min\_child\_leaf en GBM.
6. **Max\_depth** .- Profundidad máxima en niveles
7. **Max\_leaf\_nodes**.- **Cantidad de observaciones (hojas)**
8. **Gamma**.- Mínima reducción del error requerida para una división o corte de nodo.
9. **Max\_delta\_step**.- Actualización de pesos. Estilo conservador.
10. **Subsample**.- Fracción de observaciones en cada árbol.
11. **Colsample\_bytree**.- Equivalente a max\_features.
12. **Colsample\_bylevel**.- Proporción de variables para cada corte.
13. **Lambda**.- Regularización L2. (Ridge)
14. **Alpha**.- Regularización L1. (Lasso)
15. **Scale\_pos\_weight**.- En caso de desequilibrio ayuda a la convergencia.

# Parámetros de XGBoost

1. **Objective.-** Función objetivo (depende del tipo de variable respuesta)
  - ✓ **reg:linear**
  - ✓ **binary:logistic**
  - ✓ **multi:softmax**
  - ✓ **multi:softprob**
2. **Eval\_metric.-** Métrica usada en la validación de la data.
  - ✓ **Rmse** – root mean square error
  - ✓ **Mae** – mean absolute error
  - ✓ **Logloss** – negative log-likelihood
  - ✓ **Error** – Binary classification error rate (0.5 threshold)
  - ✓ **Merror** – Multiclass classification error rate
  - ✓ **Mlogloss** – Multiclass logloss
  - ✓ **Auc**: Area under the curve
3. **Seed.-** Semilla aleatoria.

# La Matriz de Confusión

		Clase predicha		
		0	1	
Clase real	0	<b>Verdadero Negativo</b>	<b>Falso Positivo</b>	Negativo Real
	1	<b>Falso Negativo</b>	<b>Verdadero Positivo</b>	Positivo Real
		Negativo Predicho	Positivo Predicho	

- Exactitud =  $(VP + VN)/(VP+FP+VN+FN)$
- Sensibilidad =  $VP/(VP+FN)$
- Especificidad =  $VN/(VN+FP)$
- Precisión =  $VP/(VP+FP)$
- F1-Score =  $2*Sens*Prec/(Sens+Prec)$

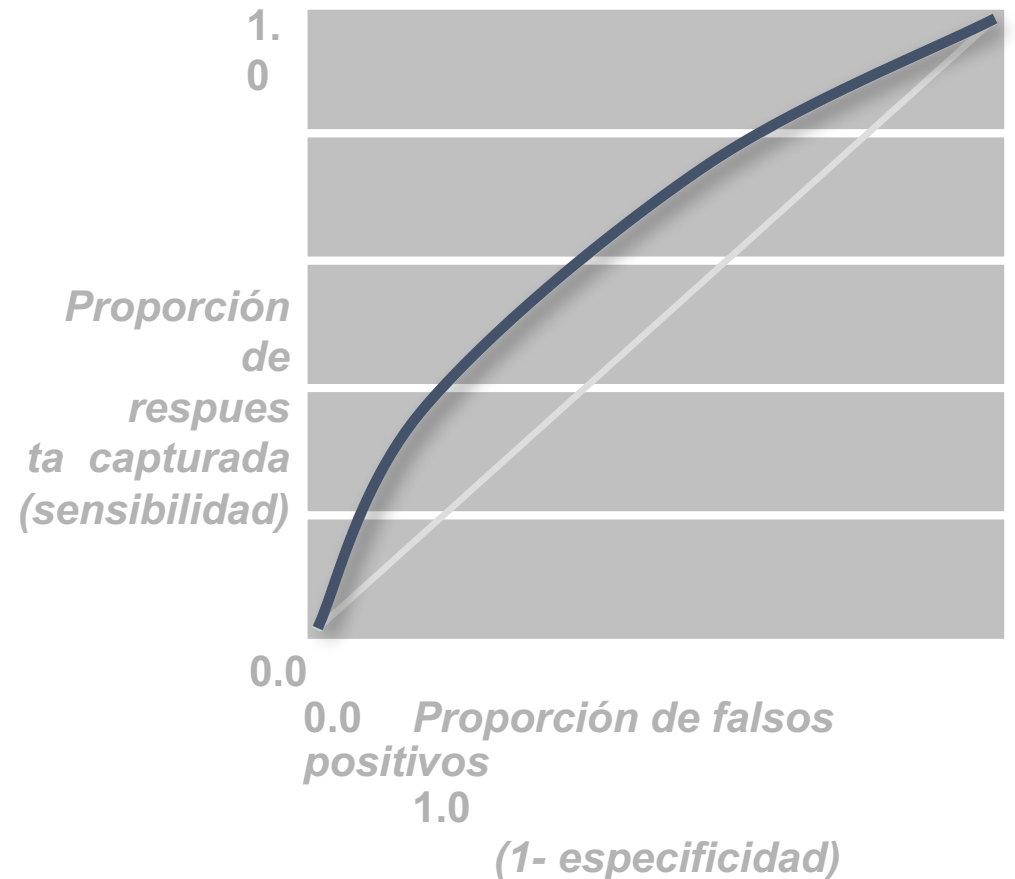
En todos los casos, el modelo es mejor si el valor está más cerca de 1.

## La Matriz de Confusión



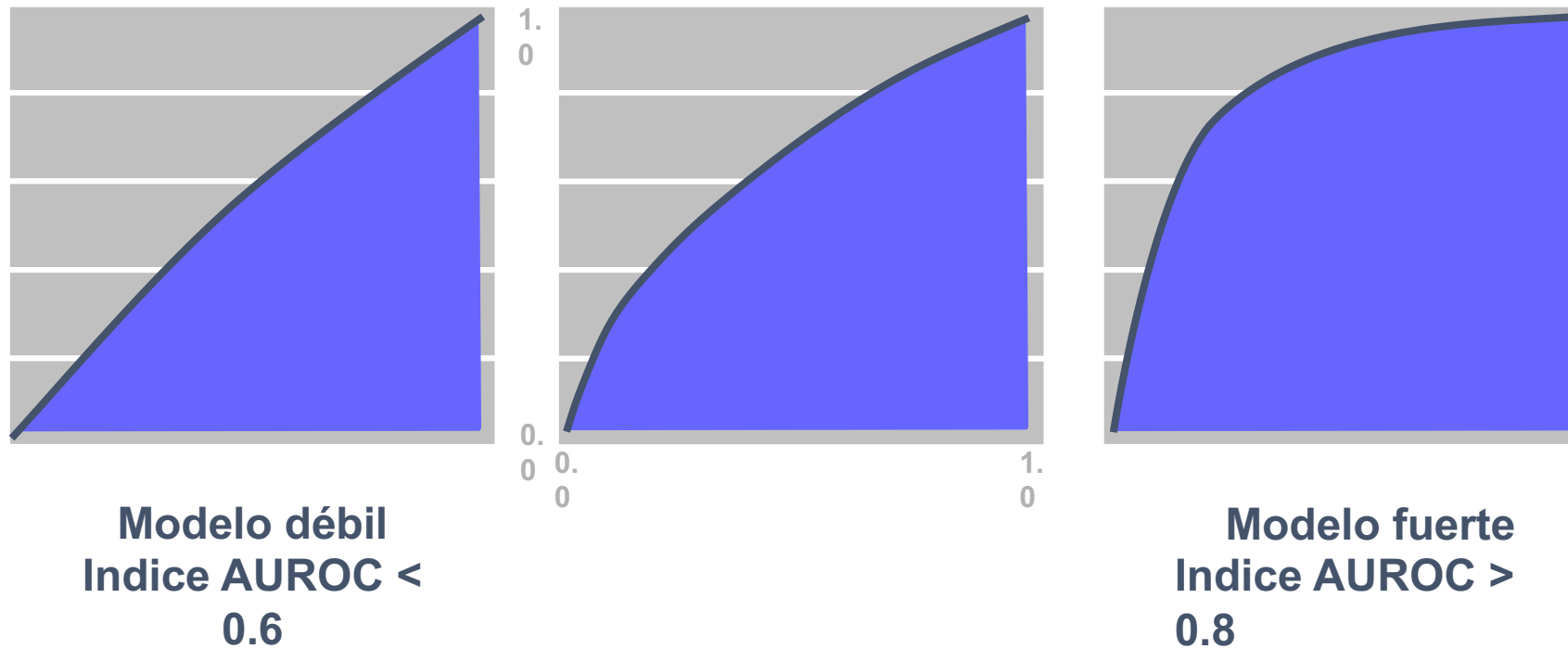
# La Curva de Operación Característica (ROC)

- El gráfico ROC exhibe la solución de compromiso entre la proporción de respuesta capturada (Sensibilidad) y la proporción de falsos positivos.



# Área bajo la Curva ROC (AUC)

- Un modelo tiene un mejor nivel de predictibilidad si el Área bajo la Curva ROC (AUC) es mayor. El máximo valor es 1.



Además:

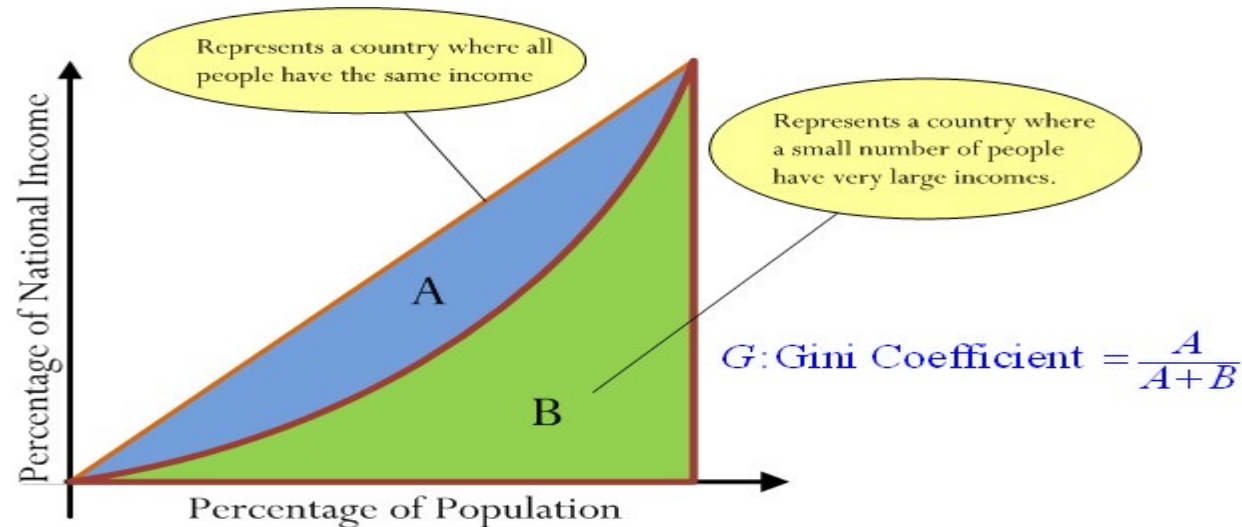
$$\text{Gini} = 2 * \text{AUC} - 1$$

A mayor Gini, el modelo es más preciso

# Coeficiente de Gini

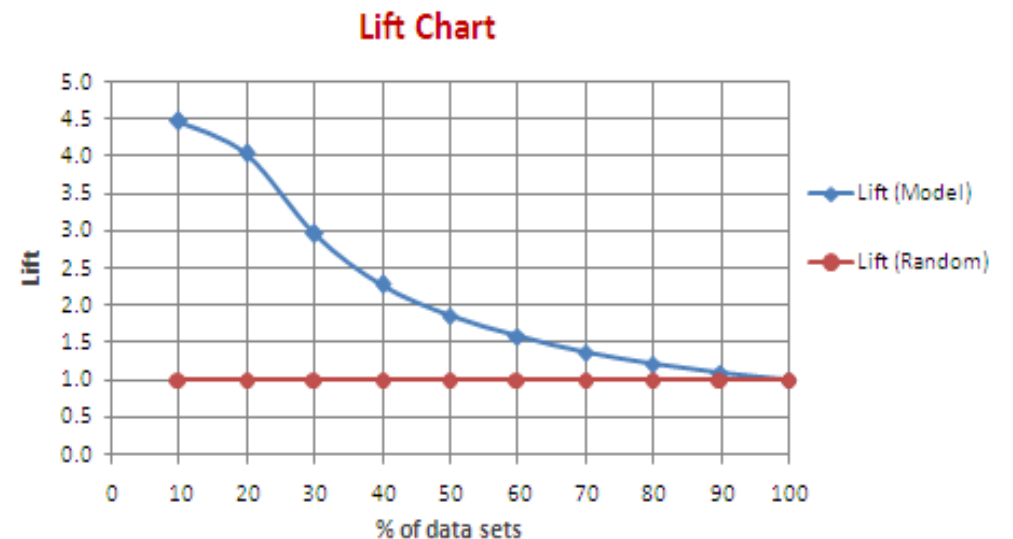
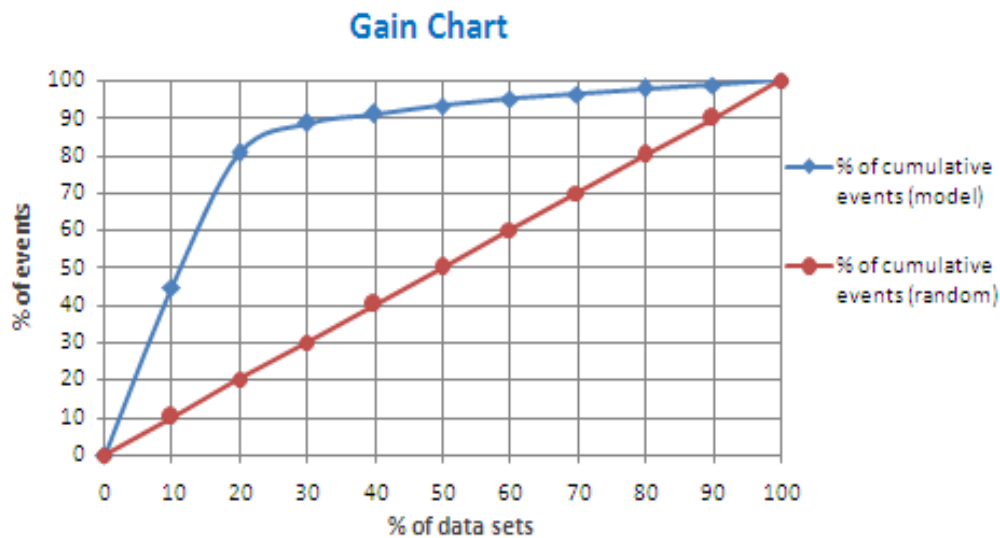
$$\text{Gini} = 2 * (\text{AUC} - 0.5)$$

Si el valor de Gini se encuentra entre 0 y 0.25, decimos que el modelo predictivo tiene una clasificación “Baja”; si el valor del Gini se encuentra entre 0.25 y 0.45, tiene una clasificación “Aceptable”; si el valor del Gini se encuentra entre 0.45 y 0.6, tiene una clasificación “Buena”, y finalmente, si el valor del Gini es mayor a 0.6, el modelo tiene una clasificación de “Muy buena”.



# Curvas de Ganancia y Lift

- **Gain:** Identificar las respuestas capturadas en cada decil.
- **Lift:** Cuántas veces más probable es encontrar el evento de interés por decil.
- En algún modelo puede haber interés en tomar sólo aquellos con mayores probabilidades.





# Curvas de Ganancia y Lift

- ¿Cómo se elaboran? Se van contabilizando la proporción de eventos capturados en cada decil (Gain) y se divide entre el porcentaje base acumulado (Lift)

**TABLA DE RESPUESTA - GANANCIAS**

Decile	Input Values		Cumulative Responses	% of events	Gain	Cumulative Lift
	Number of Cases	Number of Responses				
1	2500	2179	2179	44.71	44.71	4.47
2	2500	1753	3932	35.97	80.67	4.03
3	2500	396	4328	8.12	88.80	2.96
4	2500	111	4439	2.28	91.08	2.28
5	2500	110	4549	2.26	93.33	1.87
6	2500	85	4634	1.74	95.08	1.58
7	2500	67	4701	1.37	96.45	1.38
8	2500	69	4770	1.42	97.87	1.22
9	2500	49	4819	1.01	98.87	1.10
10	2500	55	4874	1.13	100.00	1.00
	25000	4874				

— PROGRAMA DE —  
ESPECIALIZACIÓN ANALÍTICA

---

# ADVANCED DATA SCIENCE

---