Escuela Técnica Superior de Ingeniería Informática

Máster Universitario en Ingeniería Informática

Tema 1.2
Repaso Aprendizaje Automático

Deep Learning

Máster de Ingeniería Informática

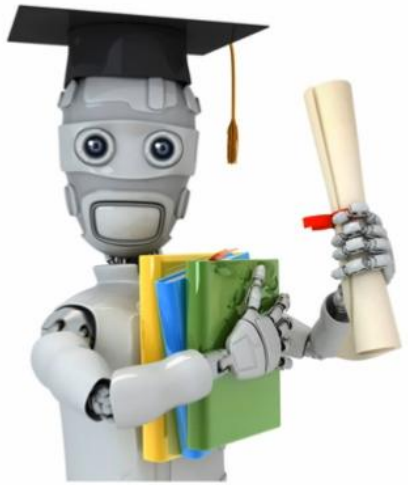
Universidad de Sevilla

Contenido

- ¿Qué es Machine Learning?
- Clasificaciones de ML
- Metodología creación de modelos

¿Qué es Machine Learning?

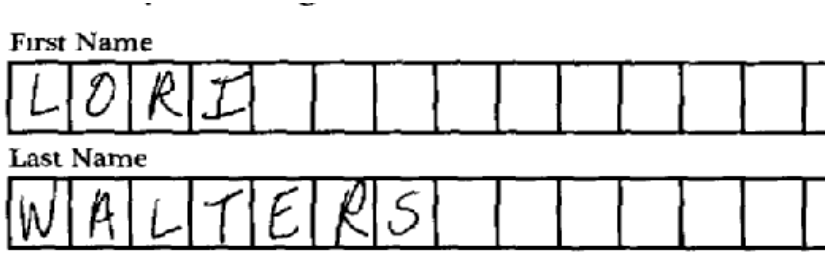
- En español: **Aprendizaje Automático** (**Aprendizaje de Máquina**)
- Rama de la **Inteligencia Artificial** cuyo objetivo es conseguir que las computadoras aprendan
- Concretamente, proceso de **inducción del conocimiento**:

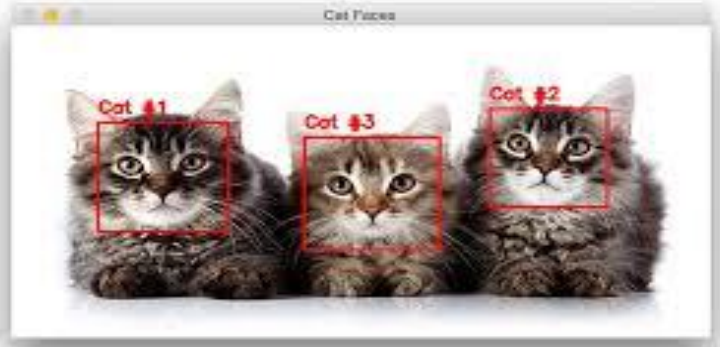


Crear algoritmos capaces de generalizar comportamientos y reconocer patrones a partir de una información suministrada como ejemplos

¿Qué es Machine Learning?

- **Learn by example**: se usan ejemplos para entrenar ordenadores a realizar tareas que serían difíciles de programar
- Algunos ejemplos de aplicación:
  - Reconocimiento de escritura manual
  - Traducción de lenguaje
  - Reconocimiento del habla
  - Clasificación de imágenes
  - Conducción autónoma

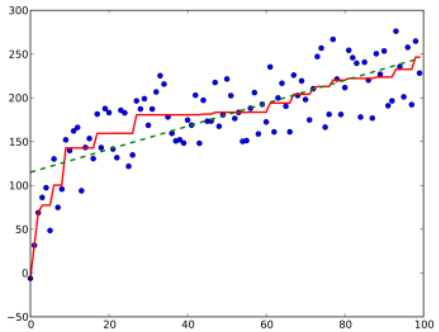


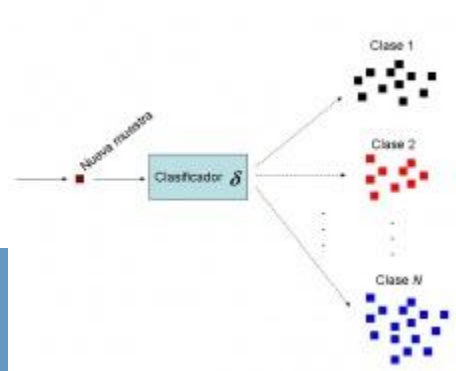


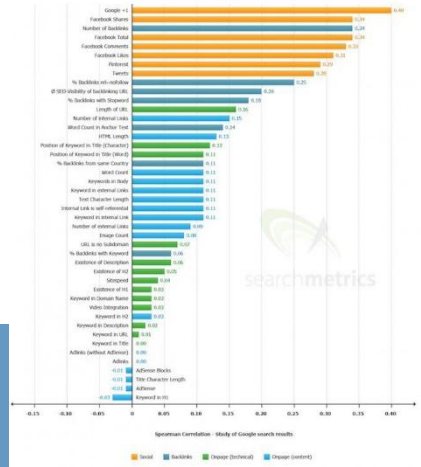
Clasificación de Machine Learning

Por tipo de objeto a predecir:

- **Regresión**: Predecir un valor continuo
- **Clasificación**: Predecir la clasificación sobre un conjunto de clases prefijadas
- **Ranking**: Predecir el orden óptimo de un conjunto de clases según un orden de relevancia prefijado



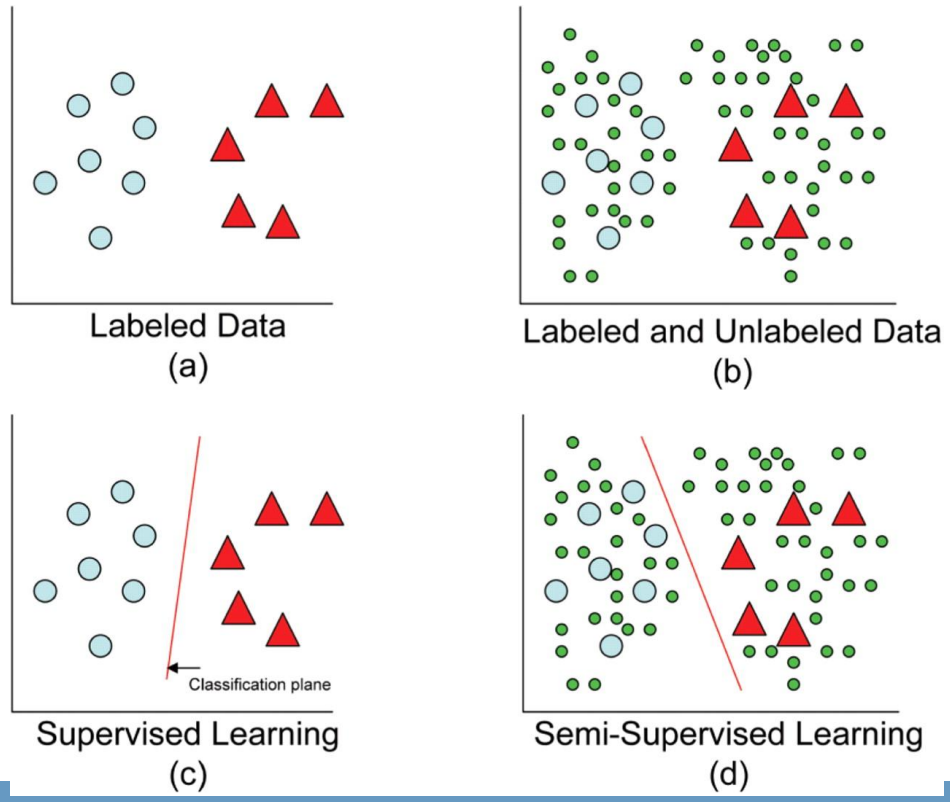


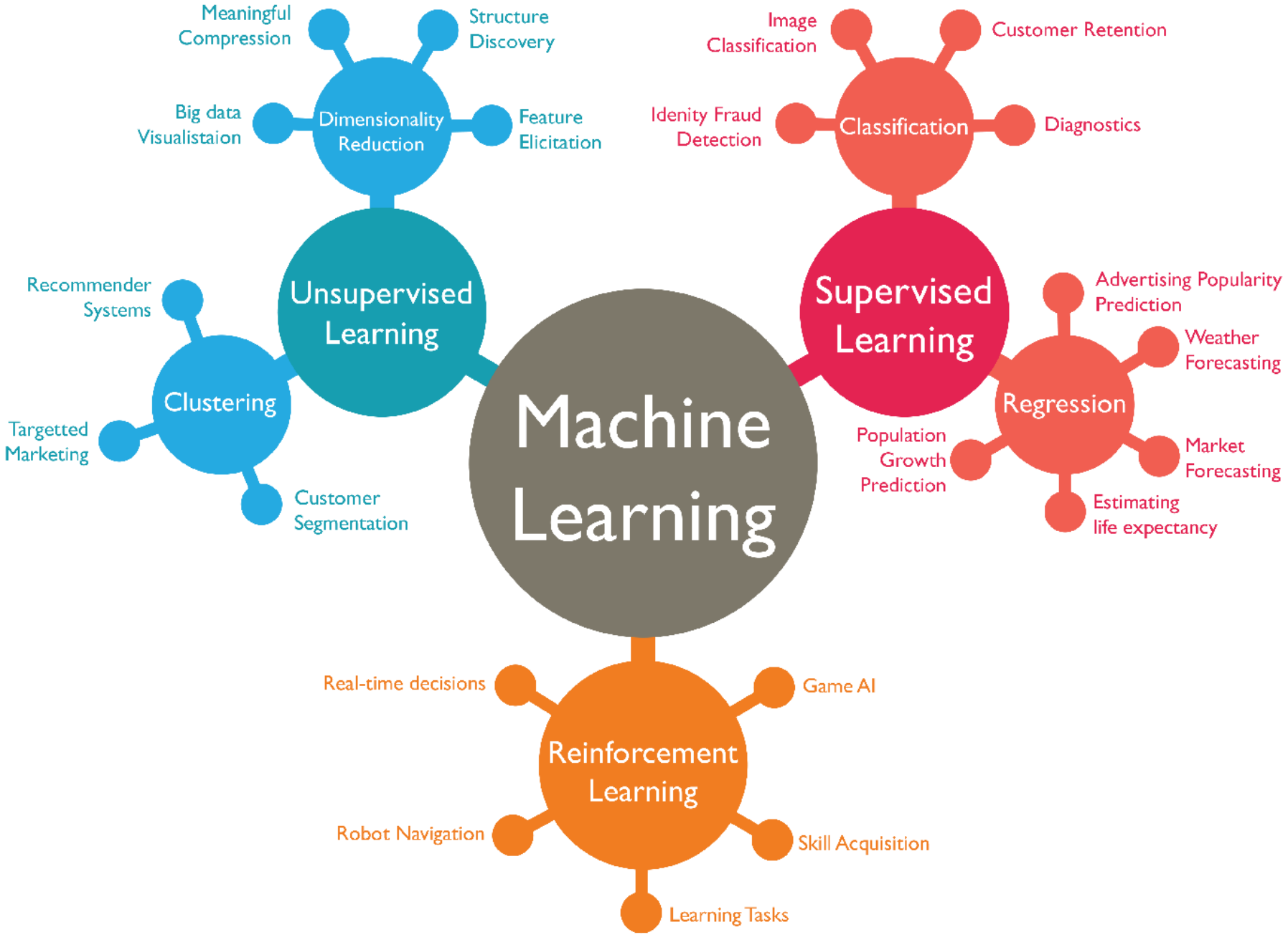


Clasificación de Machine Learning

Por cómo se usan los ejemplos:

- **Supervisado**: se conoce el valor esperado de los ejemplos (**ejemplos etiquetados**)
- **No supervisado:** solo se tiene información de los datos de entrada, no de la salida esperada
- **Semisupervisado**: Una mezcla
- **Por refuerzo**: el sistema recibe una compensación por sus acciones, e intenta tomar mejores acciones.



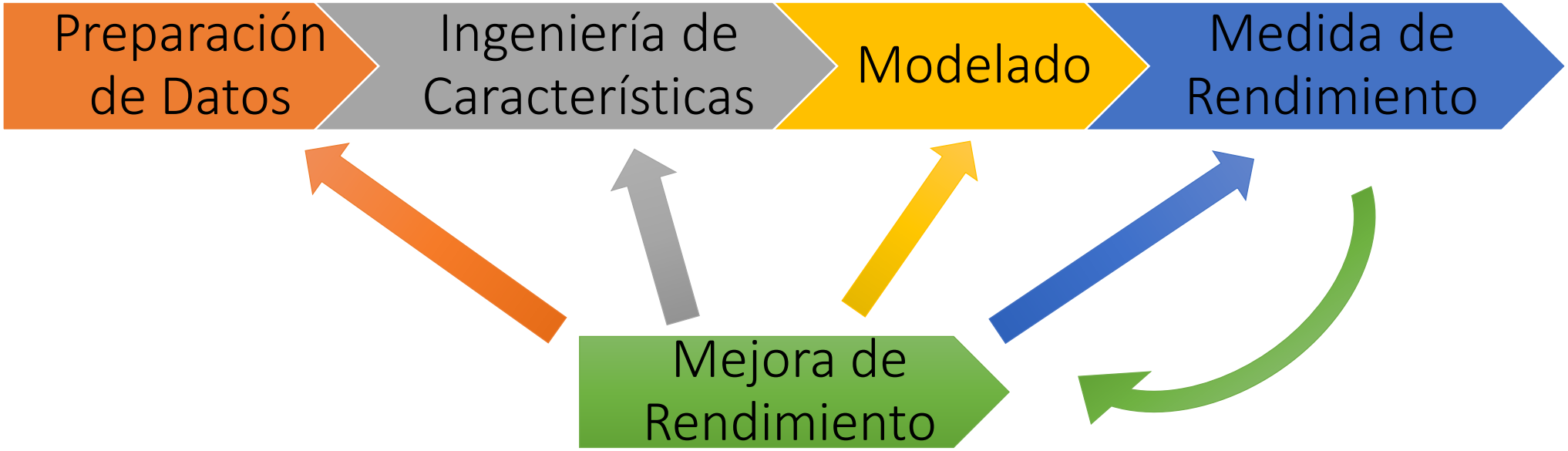

Machine Learning
Unsupervised Learning
Supervised Learning
Reinforcement Learning
Dimensionality Reduction
Clustering
Classification
Regression
Meaningful Compression
Structure Discovery
Big data Visualistaion
Feature Elicitation
Recommender Systems
Targetted Marketing
Customer Segmentation
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Advertising Popularity Prediction
Weather Forecasting
Population Growth Prediction
Market Forecasting
Estimating life expectancy
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

Objetivos de la creación de modelos ML

- Determinar la **estructura óptima** en un conjunto de datos para conseguir realizar una **tarea concreta**.
- Se obtiene aplicando **algoritmos de aprendizaje** sobre conjuntos de **datos de entrenamiento**.

Metodología por pasos

- Hay 5 pasos básicos para construir un modelo ML:



- Aunque es un proceso altamente iterativo, que debe repetirse hasta encontrar resultados satisfactorios…

Preparación de Datos	• Importación • Limpieza
Ingeniería de Características	• Datos Relevantes • Datos Útiles
Modelado	• Tipos de Algoritmos • Cómo se elige el adecuado
Medida de Rendimiento	• Métodos para medir el rendimiento • Qué indicador usar
Mejora de Rendimiento	• Porqué un modelo puede funcionar mal • Técnicas para mejorar un modelo

Preparación de Datos

- Importación
- Limpieza

Ingeniería de Características

- Datos Relevantes
- Datos Útiles

Modelado

- Tipos de Algoritmos
- Cómo se elige el adecuado

Medida de Rendimiento

- Métodos para medir el rendimiento
- Qué indicador usar

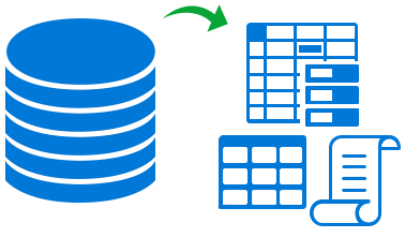
Mejora de Rendimiento

- Porqué un modelo puede funcionar mal
- Técnicas para mejorar un modelo

Preparación de Datos

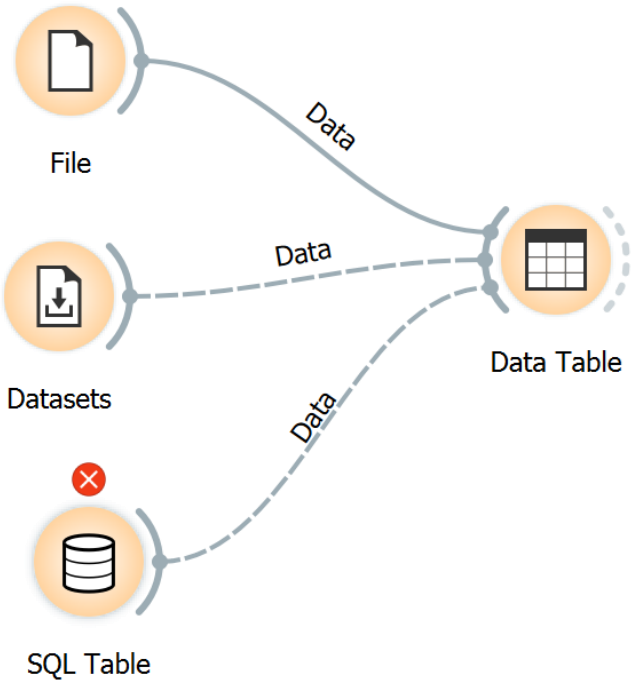
- Consta, esencialmente, de 3 pasos:

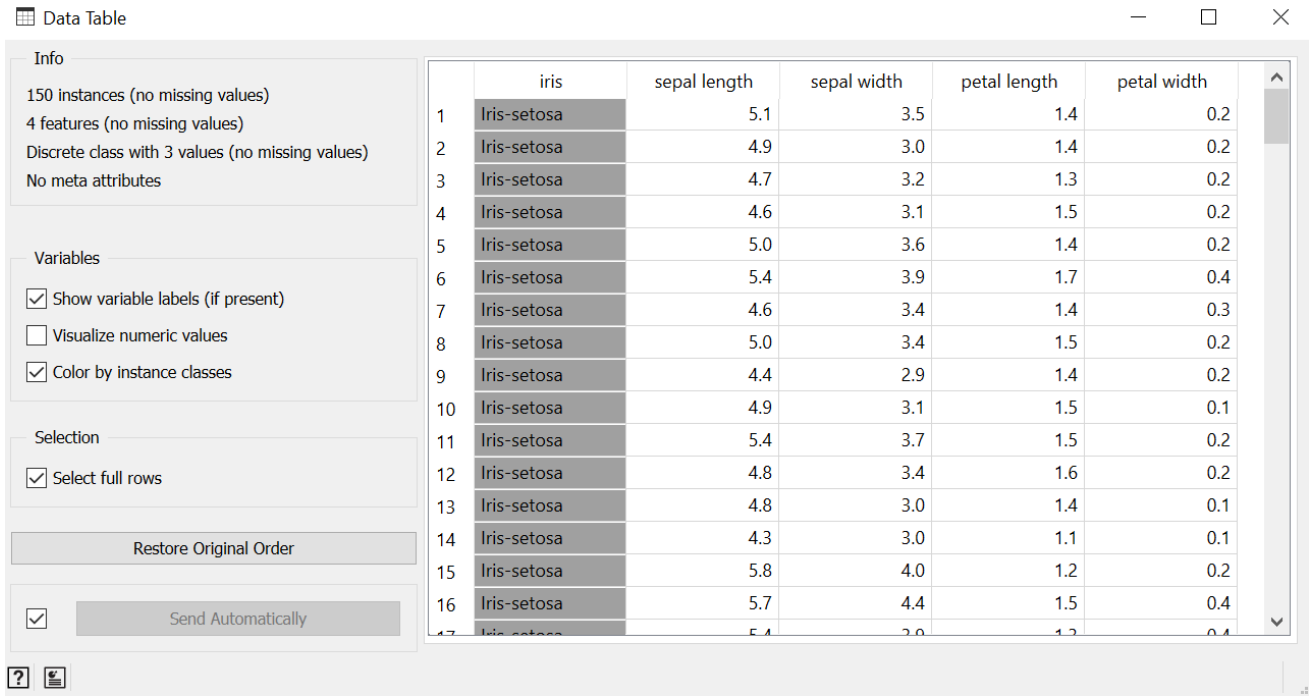
Obtención → Limpieza → Formateado



Obtención

- Por medio de consultas SQL, fichero CSV, … se obtiene una estructura regular en forma de tabla (dataframe)

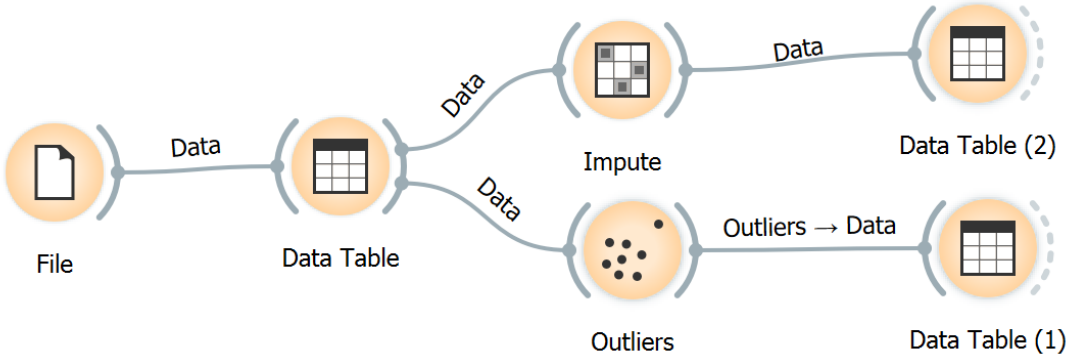


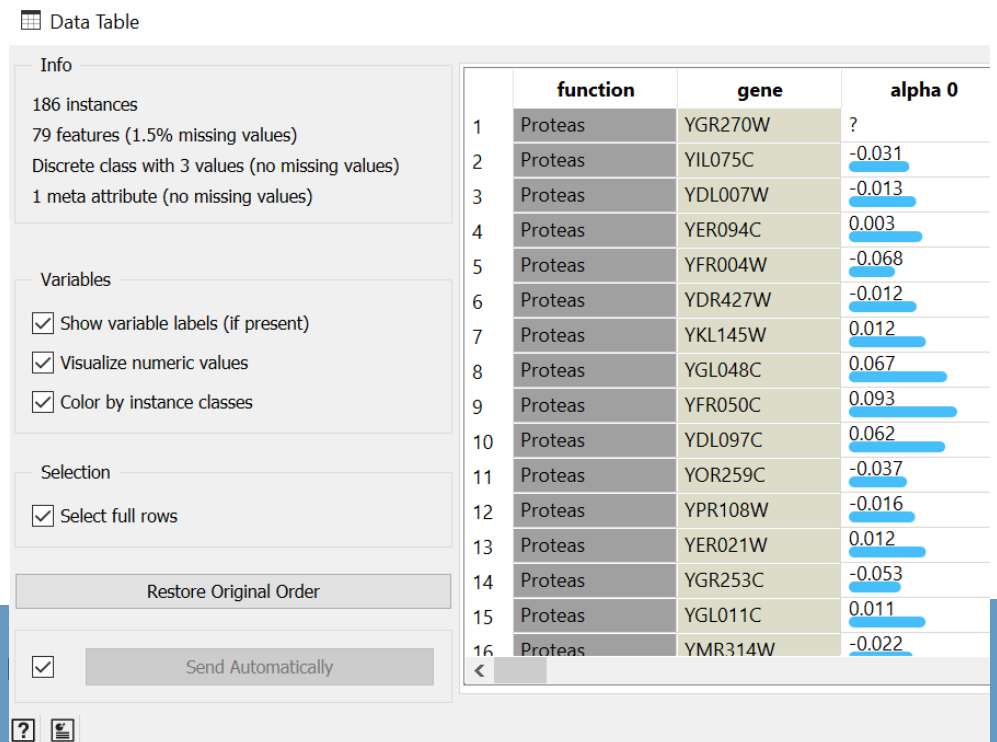


	iris	sepal length	sepal width	petal length	petal width
1	Iris-setosa	5.1	3.5	1.4	0.2
2	Iris-setosa	4.9	3.0	1.4	0.2
3	Iris-setosa	4.7	3.2	1.3	0.2
4	Iris-setosa	4.6	3.1	1.5	0.2
5	Iris-setosa	5.0	3.6	1.4	0.2
6	Iris-setosa	5.4	3.9	1.7	0.4
7	Iris-setosa	4.6	3.4	1.4	0.3
8	Iris-setosa	5.0	3.4	1.5	0.2
9	Iris-setosa	4.4	2.9	1.4	0.2
10	Iris-setosa	4.9	3.1	1.5	0.1
11	Iris-setosa	5.4	3.7	1.5	0.2
12	Iris-setosa	4.8	3.4	1.6	0.2
13	Iris-setosa	4.8	3.0	1.4	0.1
14	Iris-setosa	4.3	3.0	1.1	0.1
15	Iris-setosa	5.8	4.0	1.2	0.2
16	Iris-setosa	5.7	4.4	1.5	0.4

Limpieza

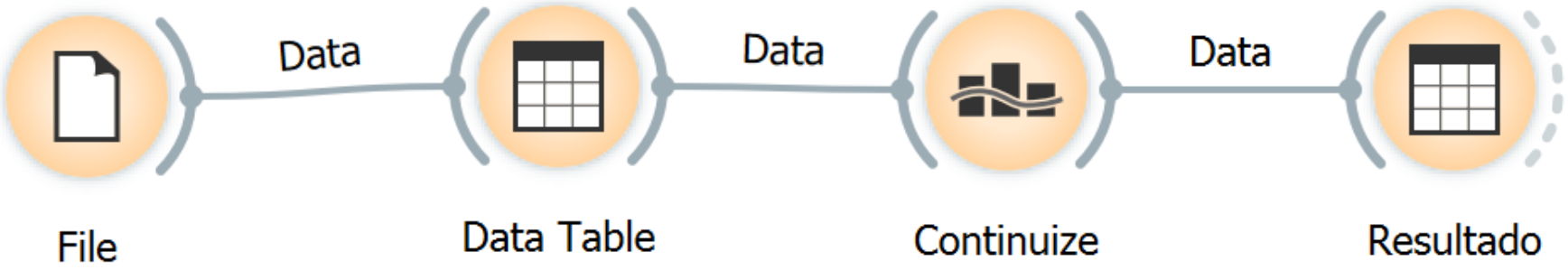
- **Valores faltantes**: Si el porcentaje de faltantes de una columna es alto, eliminarla. Si no, imputar los faltantes.
- **Outliers**: Valores que se salen de lo normal.
  - Umbral de confianza
  - Otros métodos robustos





Formateado

- Tienen el objetivo de ajustar los datos a las necesidades del algoritmo a usar.
- Por ejemplo, normalizar valores en columnas.



Formateado

- Variables categóricas (o nominales):
  - Se refiere a clases o categorías.
  - Color (rojo, azul, verde); Posición (primero, segundo, tercero)…
- Codificar variables categóricas.
  - Pasar a numéricas (perfecto para posición), o bien
  - Crear variables nuevas "dummy“ (one-hot-encoding)

Color
Rojo
Azul
Verde

Rojo	Azul	Verde
1	0	0
0	1	0
0	0	1

Preparación de Datos

- Importación
- Limpieza

Ingeniería de Características

- Datos Relevantes
- Datos Útiles

Modelado

- Tipos de Algoritmos
- Cómo se elige el adecuado

Medida de Rendimiento

- Métodos para medir el rendimiento
- Qué indicador usar

Mejora de Rendimiento

- Porqué un modelo puede funcionar mal
- Técnicas para mejorar un modelo

Ingeniería de Características

- Una **característica** (**feature**) es una propiedad individual medible del fenómeno/problema que está siendo analizado, y que será usado para formar predicciones.
  - imágenes: pixeles
  - coches autónomos: datos cámaras, sensores, GPS…
- El número de características se llama **dimensión**.

	iris	sepal length	sepal width	petal length	petal width
111	Iris-virginica	6.500	3.200	5.100	2.000
117	Iris-virginica	6.500	3.000	5.500	1.800
148	Iris-virginica	6.500	3.000	5.200	2.000
59	Iris-versicolor	6.600	2.900	4.600	1.300
76	Iris-versicolor	6.600	3.000	4.400	1.400
66	Iris-versicolor	6.700	3.100	4.400	1.400
78	Iris-versicolor	6.700	3.000	5.000	1.700
87	Iris-versicolor	6.700	3.100	4.700	1.500
109	Iris-virginica	6.700	2.500	5.800	1.800
125	Iris-virginica	6.700	3.300	5.700	2.100
141	Iris-virginica	6.700	3.100	5.600	2.400
145	Iris-virginica	6.700	3.300	5.700	2.500
146	Iris-virginica	6.700	3.000	5.200	2.300
77	Iris-versicolor	6.800	2.800	4.800	1.400
113	Iris-virginica	6.800	3.000	5.500	2.100
144	Iris-virginica	6.800	3.200	5.900	2.300
53	Iris-versicolor	6.900	3.100	4.900	1.500
121	Iris-virginica	6.900	3.200	5.700	2.300

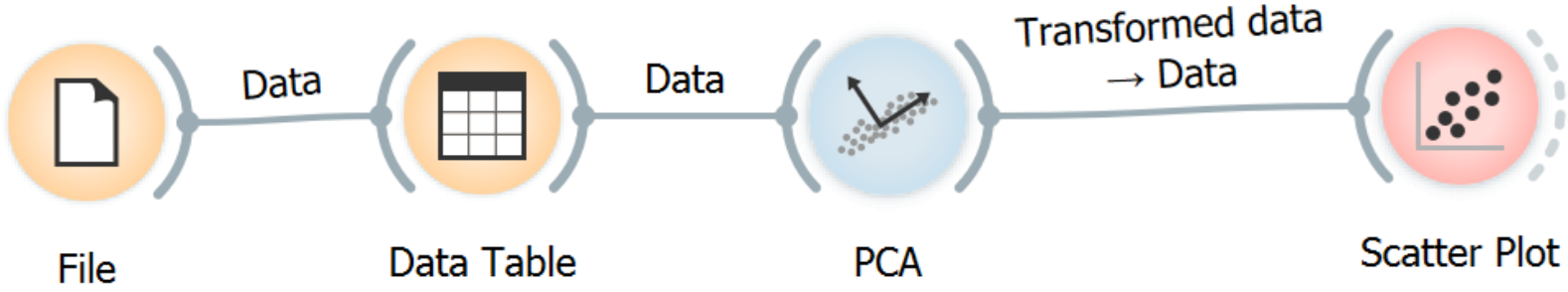
Ingeniería de Características

- **Ingeniería de Características** es el proceso de transformar datos en bruto en características relevantes. Se busca:
  - **Informativas**: Proporciona datos útiles para el modelo de aprendizaje con el fin de predecir correctamente la etiqueta.
  - **Discriminativa**: Ayuda al modelo a diferenciar entre los ejemplos de entrenamiento.
  - **No redundante**: No hay dos características que den la misma información.

Precio de venta	Sí	No
Informativa	Tamaño	Nombre del vecino
Discriminativa	Tamaño	Habitable
No redundante	Tamaño	Tamaño en hectáreas

Ingeniería de Características

- Algunos métodos:
  - **Selección de Características**: eliminando las no relevantes:
    - eliminación recursiva (eliminar características no informativas),
    - filtro umbral de varianza (eliminar carac. no discriminativas),
    - filtro de alta correlación (eliminar características redundantes)
  - **Extracción de Características**: Comienza con un conjunto de datos medibles y construye automáticamente características derivadas que son más relevantes. PCA, t-SNE,…



Preparación de Datos
- Importación
- Limpieza

Ingeniería de Características
- Datos Relevantes
- Datos Útiles

Modelado
- Tipos de Algoritmos
- Cómo se elige el adecuado

Medida de Rendimiento
- Métodos para medir el rendimiento
- Qué indicador usar

Mejora de Rendimiento
- Porqué un modelo puede funcionar mal
- Técnicas para mejorar un modelo

Modelado

Algoritmo	Pros 	Contras 
Paramétrico (p.ej. regresión lineal)	• **Más simple**: Fácil de entender e interpretar • **Más rápido**: Para ajustarse a los datos • **Menos datos**: Necesitan pocos datos para alcanzar un buen rendimiento	• **Complejidad limitada**: Más adecuados a casos en los que se intuye la estructura que deben tener los datos
No Paramétrico (p.ej. KNN, árboles de decisión)	• **Flexibilidad**: Se pueden ajustar a un gran número de formas funcionales, sin suposiciones previas • **Rendimiento**: ofrece mejor rendimiento sobre datos más complejos	• **Más lento** • **Más datos** • **Overfitting**

Modelado

- Además, los modelos suelen disponer de **hiperparámetros** que permiten ajustar su funcionamiento.
- Un **hiperparámetro** es un parámetro del algoritmo, que permite elegir características de cómo el algoritmo se va a aplicar.
- **No se deben confundir con los parámetros del algoritmo.**

Algoritmo	Hiperparámetros	Parámetros
Regresión Lineal	• **Fit_intercept**: Decide si el término β_0 se incluye en la ecuación	β
Random Forest	• **n_estimators**: Número de árboles a considerar • **criterion**: Indicador para determinar el atributo seleccionado para hacer la división en cada árbol	No paramétrico
K-medias	• **init**: Método de inicialización de los centroides.	No paramétrico

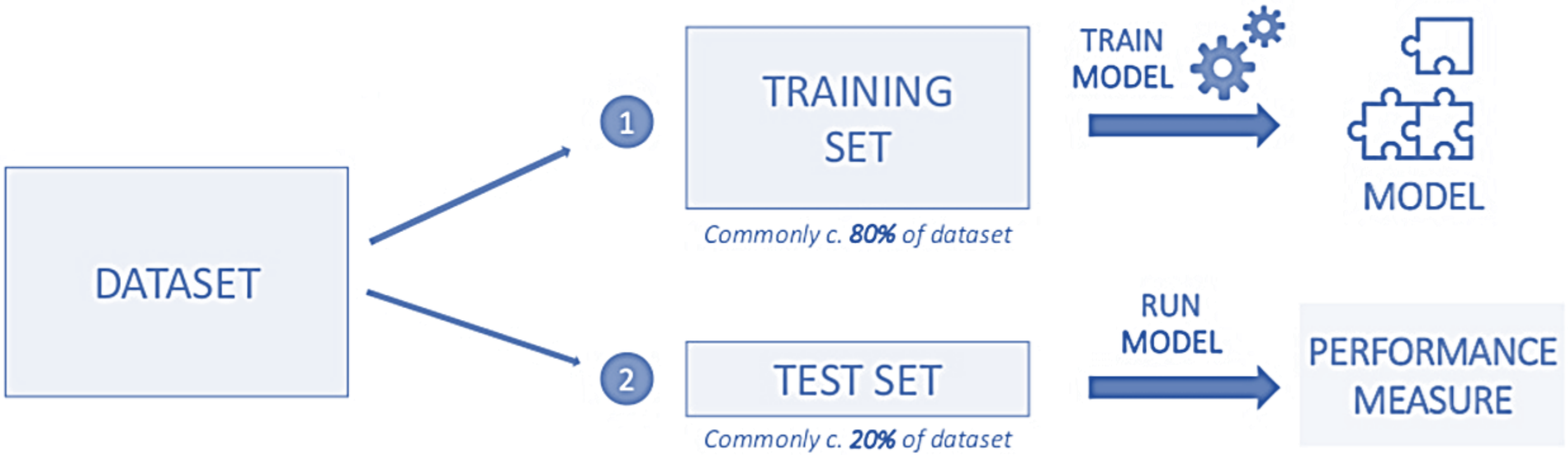
Modelado

- La mayoría de los algoritmos tienen la misma capacidad, son equivalentes... se diferencian por los datasets sobre los que operan.
- **Teorema "No Free Lunch":**
  - Cuando se promedian a lo largo de todas las posibles situaciones, todos los modelos funcionan igual de bien (o igual de mal)
  - Para cada par de modelos, podemos encontrar un par de datasets en los que cada uno se comporta mejor que el otro.

Preparación de Datos	• Importación • Limpieza
Ingeniería de Características	• Datos Relevantes • Datos Útiles
Modelado	• Tipos de Algoritmos • Cómo se elige el adecuado
Medida de Rendimiento	• Métodos para medir el rendimiento • Qué indicador usar
Mejora de Rendimiento	• Porqué un modelo puede funcionar mal • Técnicas para mejorar un modelo

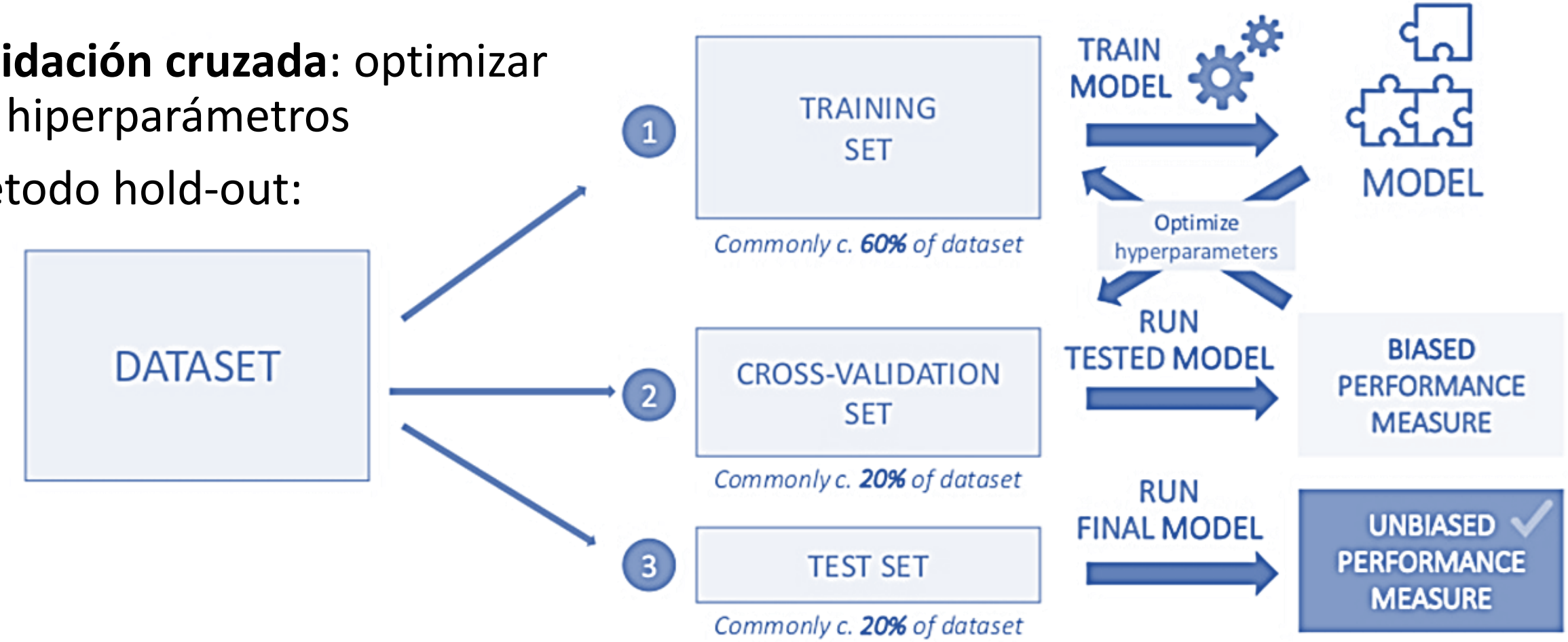
Medida del Rendimiento

- Partir el dataset en training y test



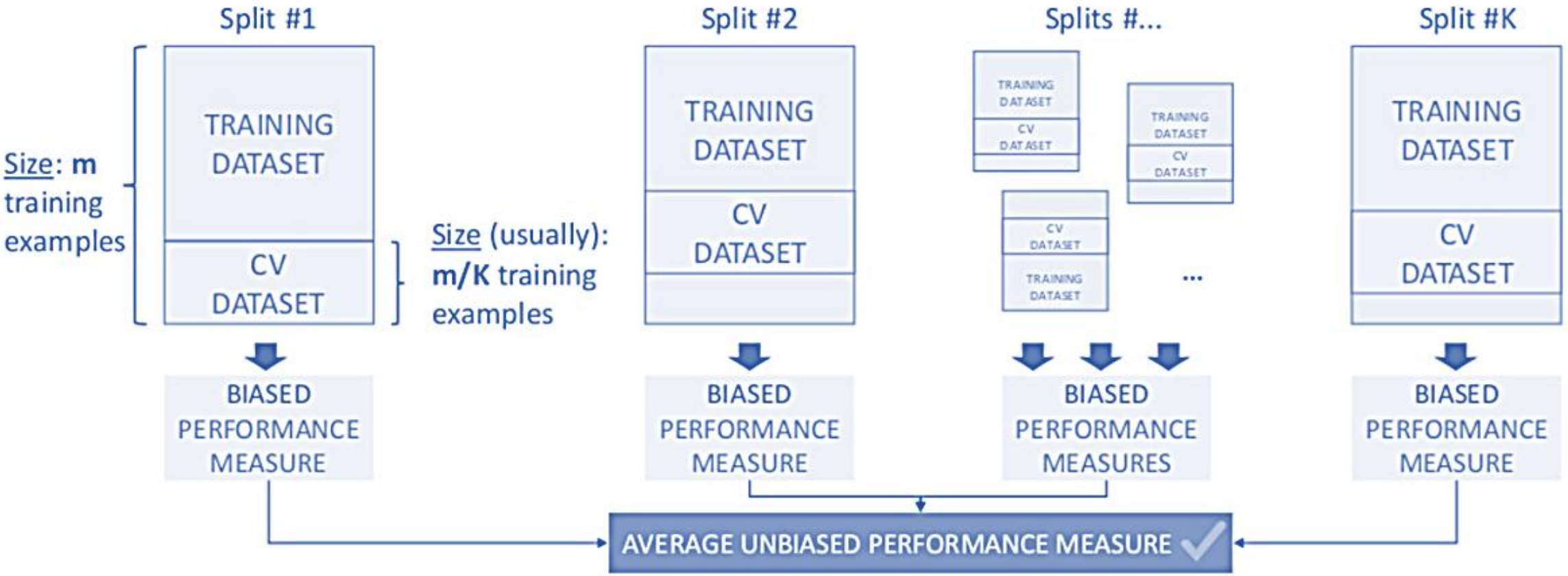
Medida del Rendimiento

- **Validación cruzada**: optimizar los hiperparámetros
- Método hold-out:



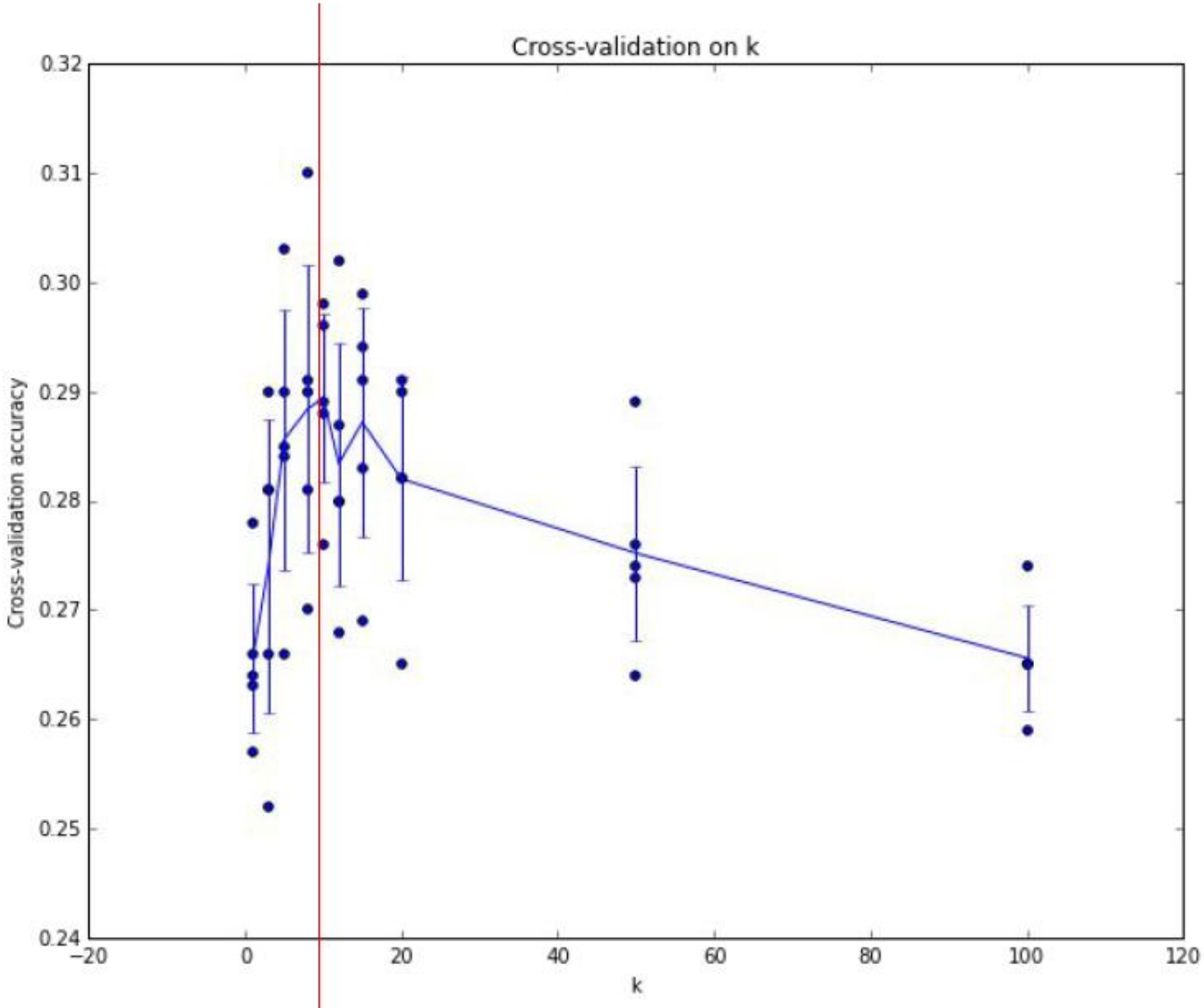
Medida del Rendimiento

- **Validación cruzada**: optimizar los hiperparámetros
- Método K-validación cruzada:



Medida del Rendimiento

- Un ejemplo de K-validación con KNN



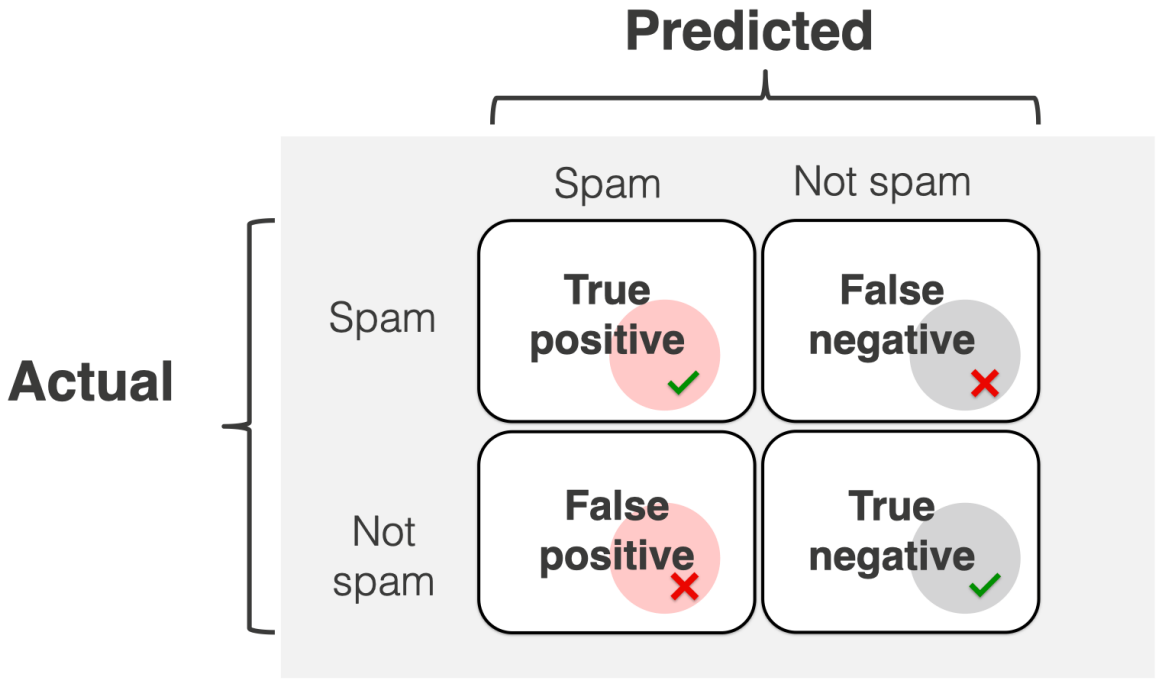
Medida del Rendimiento

- ¿Cómo medimos el rendimiento?
- **Matriz de confusión** (para clasificación)

Predicted \ Truth	Asphalt	Concrete	Grass	Tree	Building	Total
Asphalt	2385	4	0	1	4	2394
Concrete	0	332	0	0	1	333
Grass	0	1	908	8	0	917
Tree	0	0	0	1084	9	1093
Building	12	0	0	6	2053	2071
Total	2397	337	908	1099	2067	6808

Medida del Rendimiento

- Rendimiento en Clasificación



https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall

Medida del Rendimiento

- Rendimiento en Clasificación

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}}$$

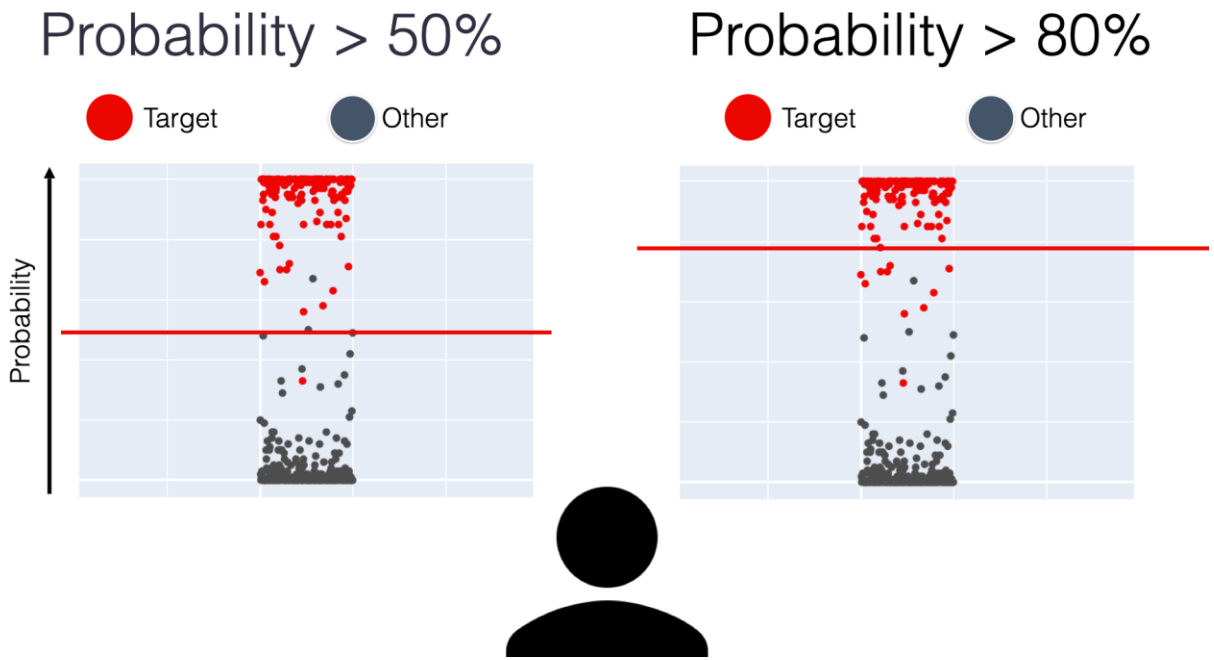
$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall

Medida del Rendimiento

- Rendimiento en Clasificación



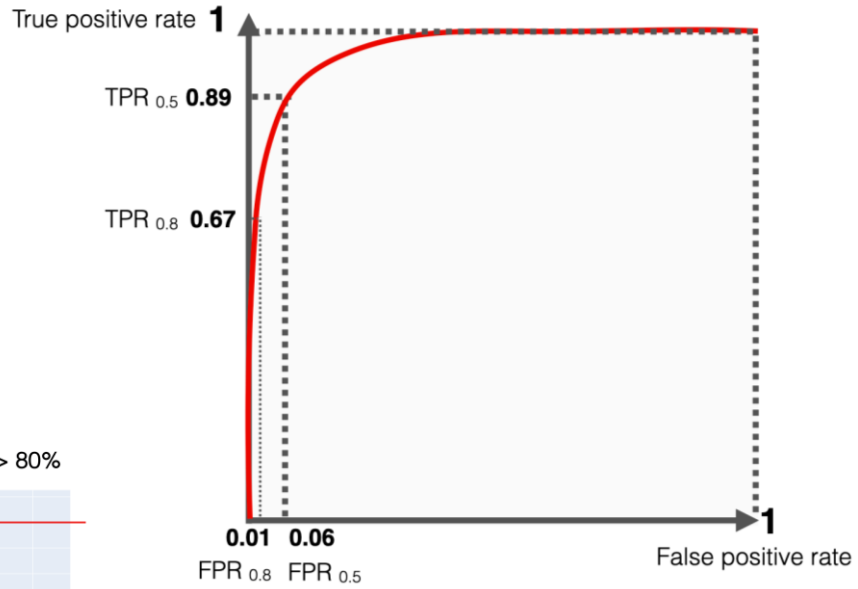
$$TPR = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

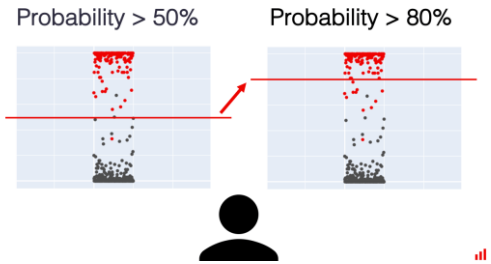
$$FPR = \frac{\text{False Positives}}{\text{False Positives + True Negatives}}$$

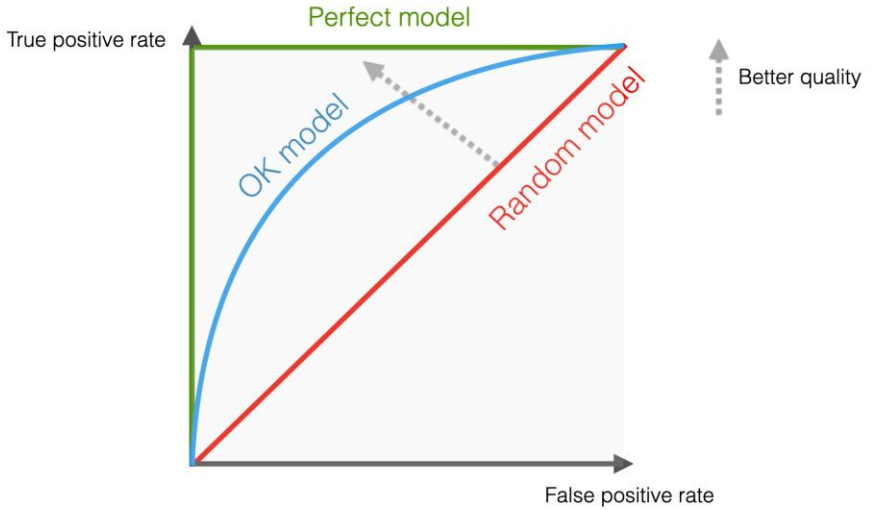
https://www.evidentlyai.com/classification-metrics/explain-roc-curve

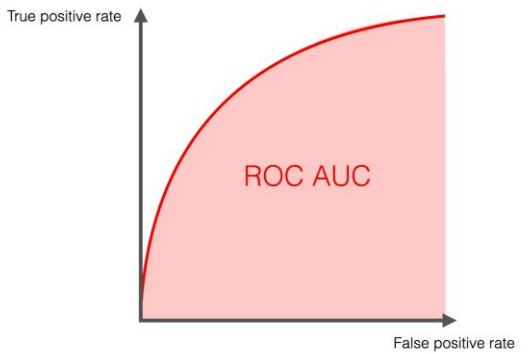
Medida del Rendimiento

- Rendimiento en Clasificación









https://www.evidentlyai.com/classification-metrics/explain-roc-curve

Medida del Rendimiento

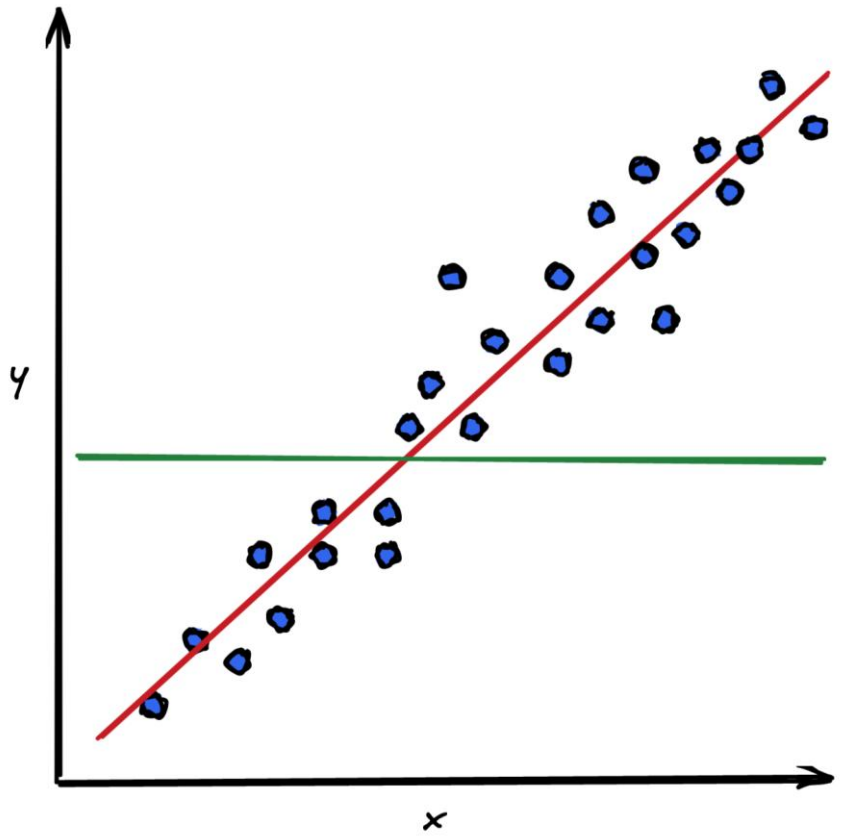
- Rendimiento en Regresión

$$MSE = \frac{1}{N}\Sigma_{i=1}^{N}(y_i - \hat{y}_i)^2$$

$$RMSE = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

$$MAE = \frac{1}{N}\Sigma_{i=1}^{N}|y_i - \hat{y}_i|$$

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\Sigma_{i=1}^{N}(y_i - \hat{y}_i)^2}{\Sigma_{i=1}^{N}(y_i - \bar{y}_i)^2}$$



R-squared: Comparing the fit of a linear model to a simple mean benchmark

Medida del Rendimiento

- Importante: Crea un modelo lo antes posible y mánchate las manos...
- Un modelo fallido muchas veces da tanta información sobre el proceso real como uno válido

Preparación de Datos	• Importación • Limpieza
Ingeniería de Características	• Datos Relevantes • Datos Útiles
Modelado	• Tipos de Algoritmos • Cómo se elige el adecuado
Medida de Rendimiento	• Métodos para medir el rendimiento • Qué indicador usar
Mejora de Rendimiento	• Porqué un modelo puede funcionar mal • Técnicas para mejorar un modelo

¿Porqué puede funcionar mal un modelo?

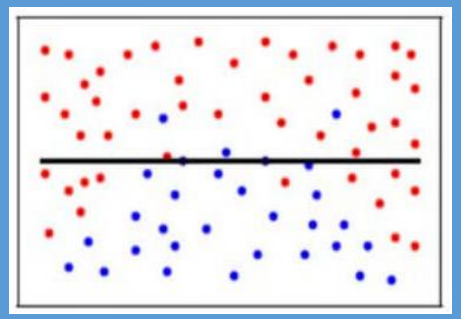
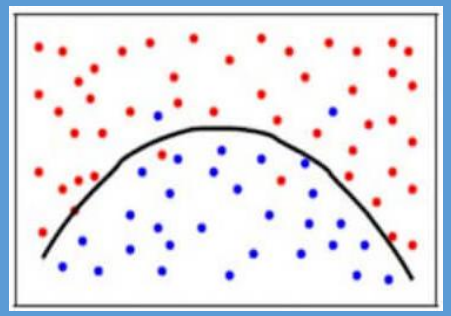
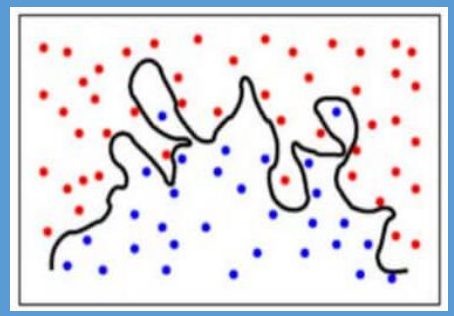
Un modelo debe **generalizar bien sobre datos no conocidos**, reproduciendo la estructura subyacente de los mismos, pero obviando el ruido.

Los principales problemas que puede presentar un modelo son:

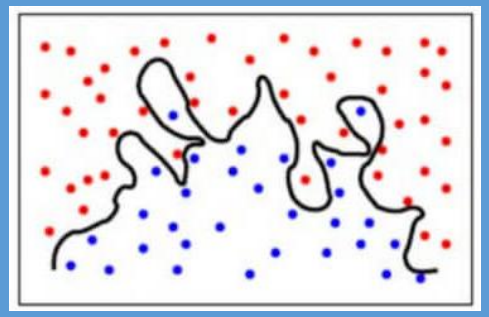
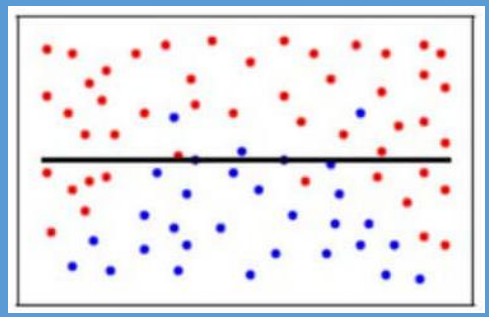
- **Underfitting**: El modelo es demasiado simple para reproducir la estructura de los datos. Se dice que tiene un alto **bias**.
- **Overfitting**: El modelo es demasiado complejo para reproducir la estructura de los datos. Captura el ruido de los datos de entrenamiento. Se dice que tiene una alta **varianza**.

¿Porqué puede funcionar mal un modelo?

Una técnica sencilla para saber si el modelo sufre de **Underfitting** o de **Overfitting** se obtiene de la comparación entre su comportamiento con los datos de entrenamiento y los datos de test:

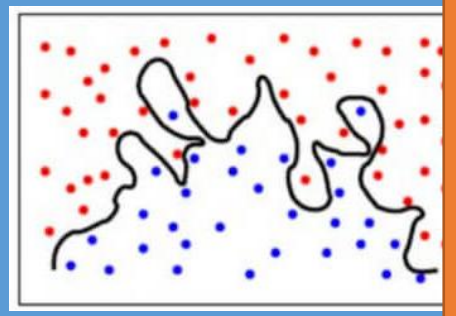
	 Underfitting	 Correcto	 Overfitting
Entrenamiento	Malo	Bueno	Muy bueno
Test	Malo	Bueno	Malo

¿Cómo se soluciona?

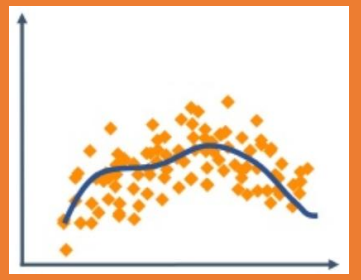
		 Overfitting	 Underfitting
Posibles Soluciones	**Datos**	Más datos de entrenamiento	
		Menos características	Más características
	Algoritmos	Algoritmos más simples	Algoritmos más complejos
		Regularización	
		Bagging	Boosting

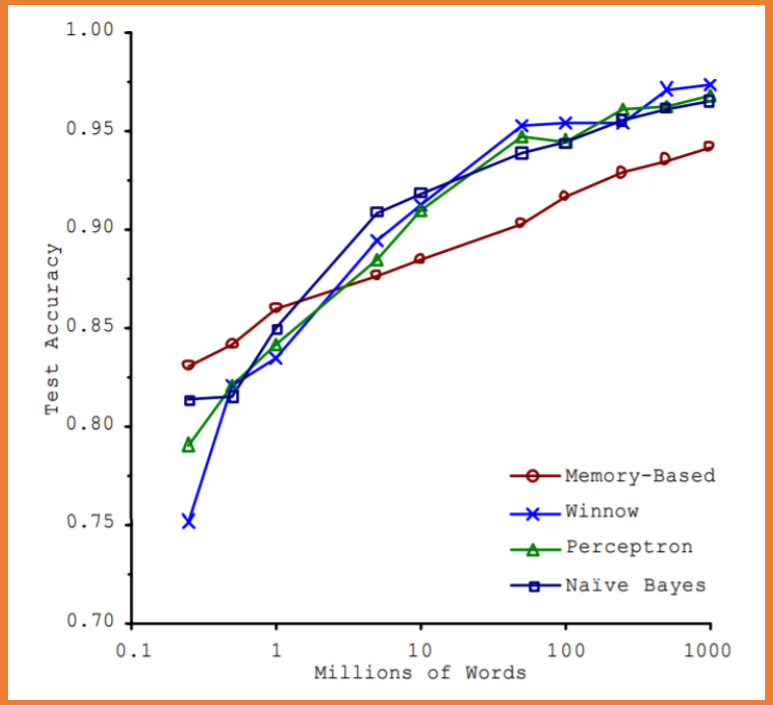
¿Cómo se soluciona?

		Overfitting
Posibles Soluciones	Datos	Más datos d
		Menos carac
	Algoritmos	Algoritmos más s
		Regularización
		Bagging

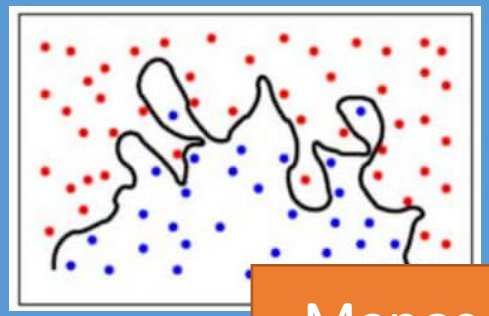
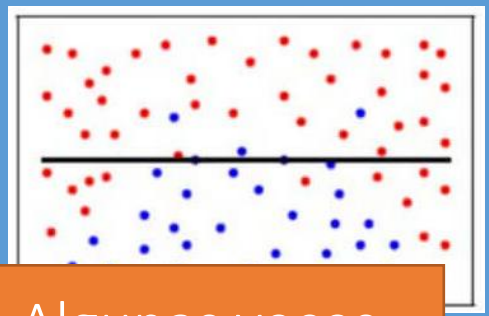


Más datos de entrenamiento: Cuantos más datos haya, menos sensible es el modelo a los datos particulares, y en consecuencia generalizará mejor. Reduce la varianza.





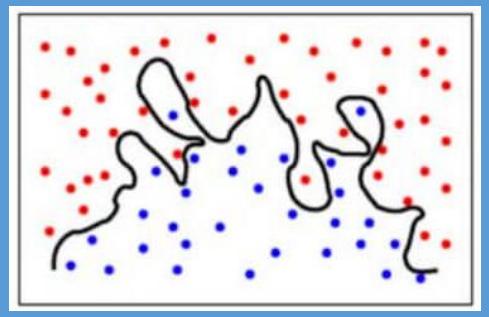
¿Cómo se soluciona?

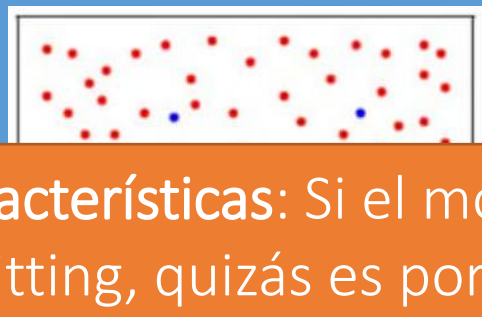
		 Overf	
Posibles Soluciones	**Datos**	Más datos de e	
		Menos características	Más características
	Algoritmos	Algoritmos más simples	Algoritmos más complejos
		Regularización	
		Bagging	Boosting

Menos características: Algunas veces, algunas características añaden más ruido que información.

¿Cómo se soluciona?

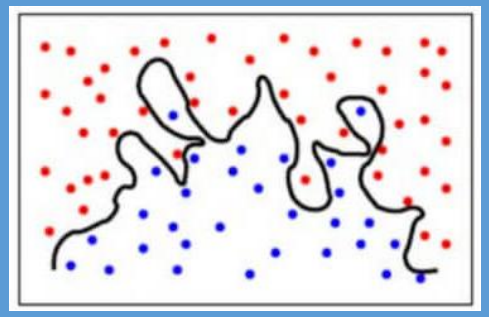
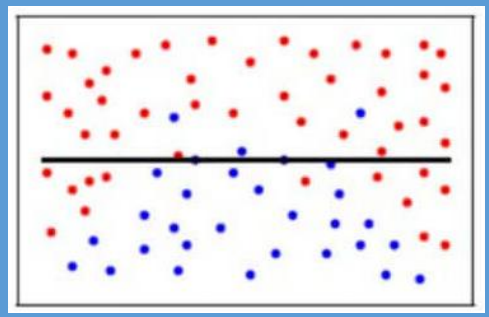
		Overfitting	
Posibles Soluciones	**Datos**	Más datos de entrenamiento	
		Menos características	Más características
	Algoritmos	Algoritmos más simples	Algoritmos más complejos
		Regularización	
		Bagging	Boosting





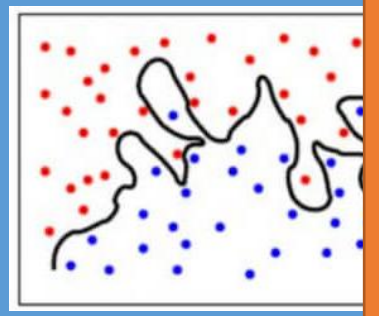
Más características: Si el modelo es underfitting, quizás es porque no tenemos suficiente información... no es un proceso adivinativo. Reduce el bias.

¿Cómo se soluciona?

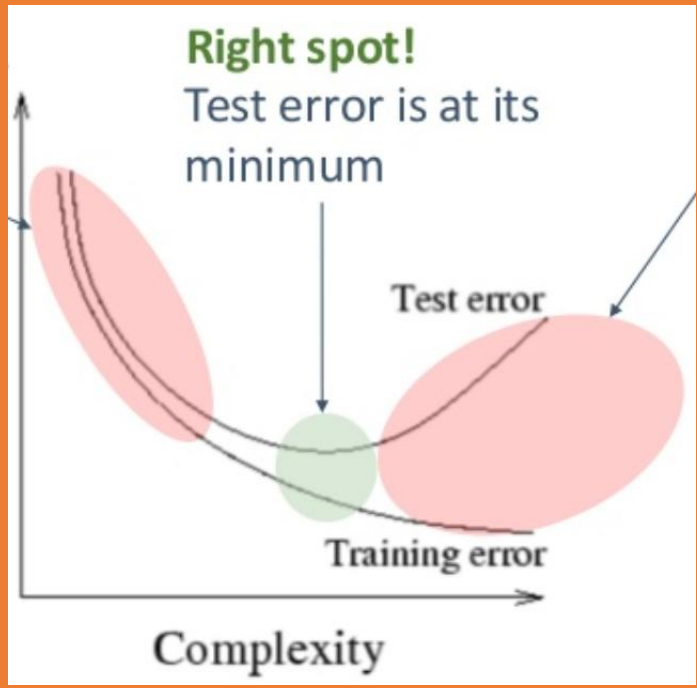
		 Overfitting	 Underfitting
Posibles Soluciones	Datos		
			ás características
	Algoritmos		itmos más complejos
		Regularización	
		Bagging	Boosting

Datos: Los datos son la clave. Reducen la varianza (**overfitting**) con más instancias, y reducen la bias (**underfitting**) con más características

¿Cómo se soluciona?

		 **Overfittin**	
Posibles Soluciones	**Datos**	Más datos de entre	
		Menos caracte	as
	Algoritmos	Algoritmos más simples	Algoritmos más complejos
		Regularización	
		Bagging	Boosting

Complejidad de los Algoritmos



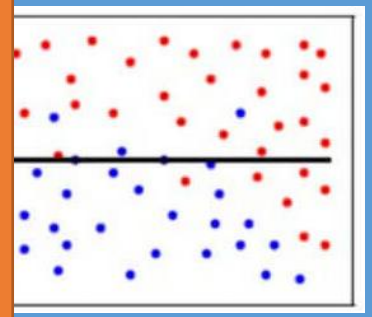
¿Cómo se soluciona?

Regularización: Reduce overfitting añadiendo un término de complejidad a la función de coste a minimizar (Navaja de Ockam).

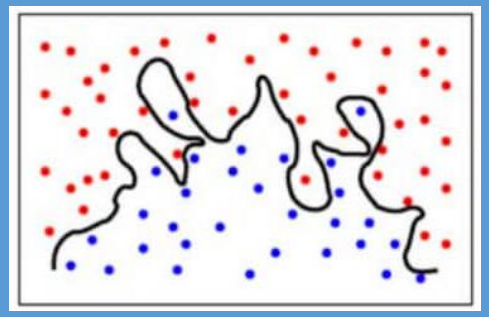
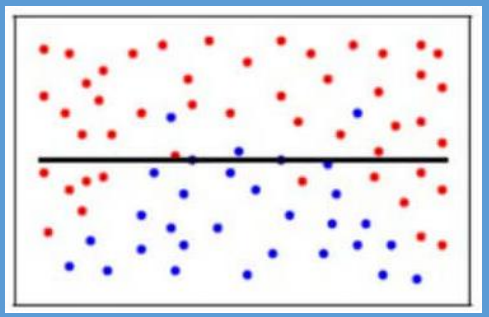
$$J(\beta) = \sum_{i=0}^{m} (Y_i - X_i\beta)^2 + \alpha \sum_{j=0}^{n} \beta_j^2$$

Regularization term

α = "Complexity" / "Penalty" / "Regularization" parameter

			
			nderfitting
Posibles Soluciones	**Datos**	Más	
		Menos sticas	Más características
	Algoritmos	Algoritmo as simples	Algoritmos más complejos
		Regularización	
		Bagging	Boosting

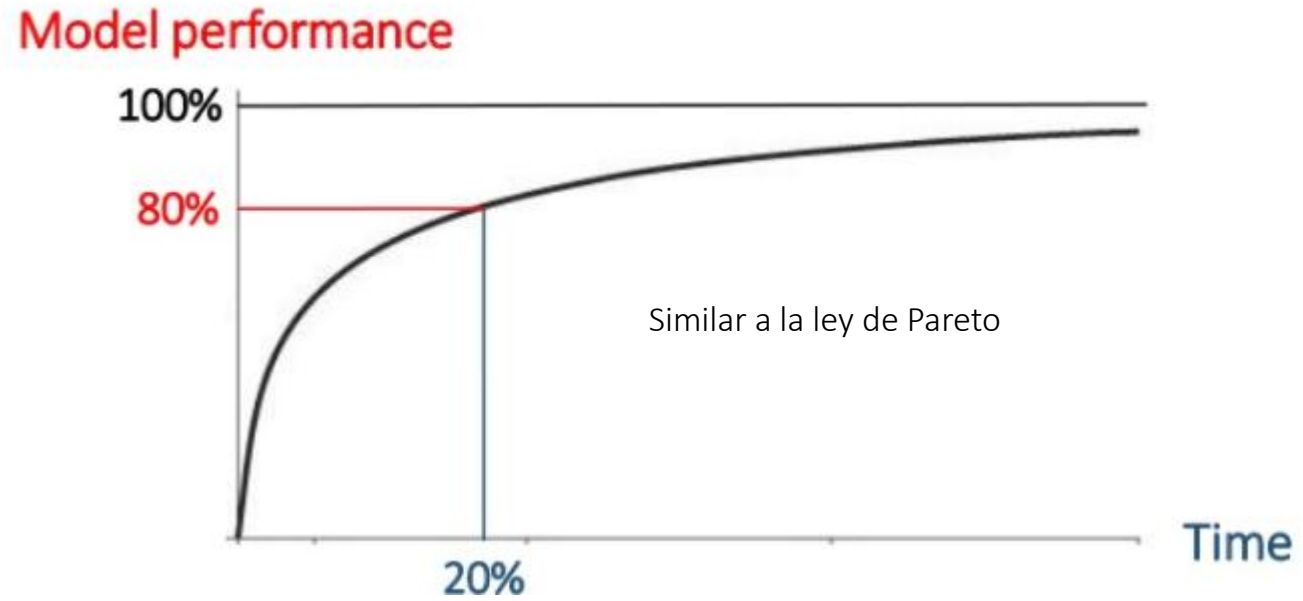
¿Cómo se soluciona?

		Overfitting	Underfitting
			
Posibles Soluciones	**Datos**	Más datos de entrenamiento	
		Menos características	Más características
	Algoritmos	Algoritmos más simples	Algoritmos más complejos
		R	
		Bagging	Boosting

Métodos Avanzados: Ensemble

Pero no olvides que…

Así es como suele funcionar el tiempo empleado en mejorar tu modelo:



Así que cuando no merezca la pena el esfuerzo, detente…

Recuerda la Metodología

Hay 5 pasos básicos para construir un modelo ML: