

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación



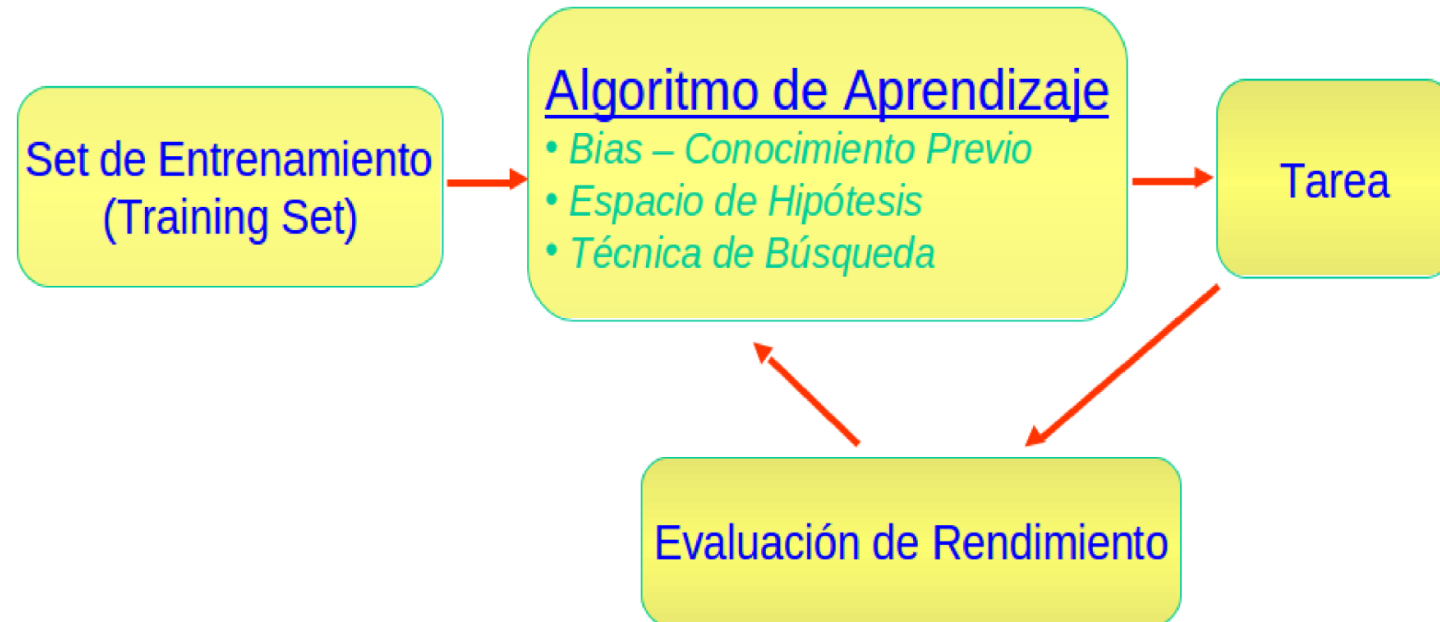
Sistemas Urbanos Inteligentes

Fundamentos de Machine Learning Parte 2

Hans Löbel

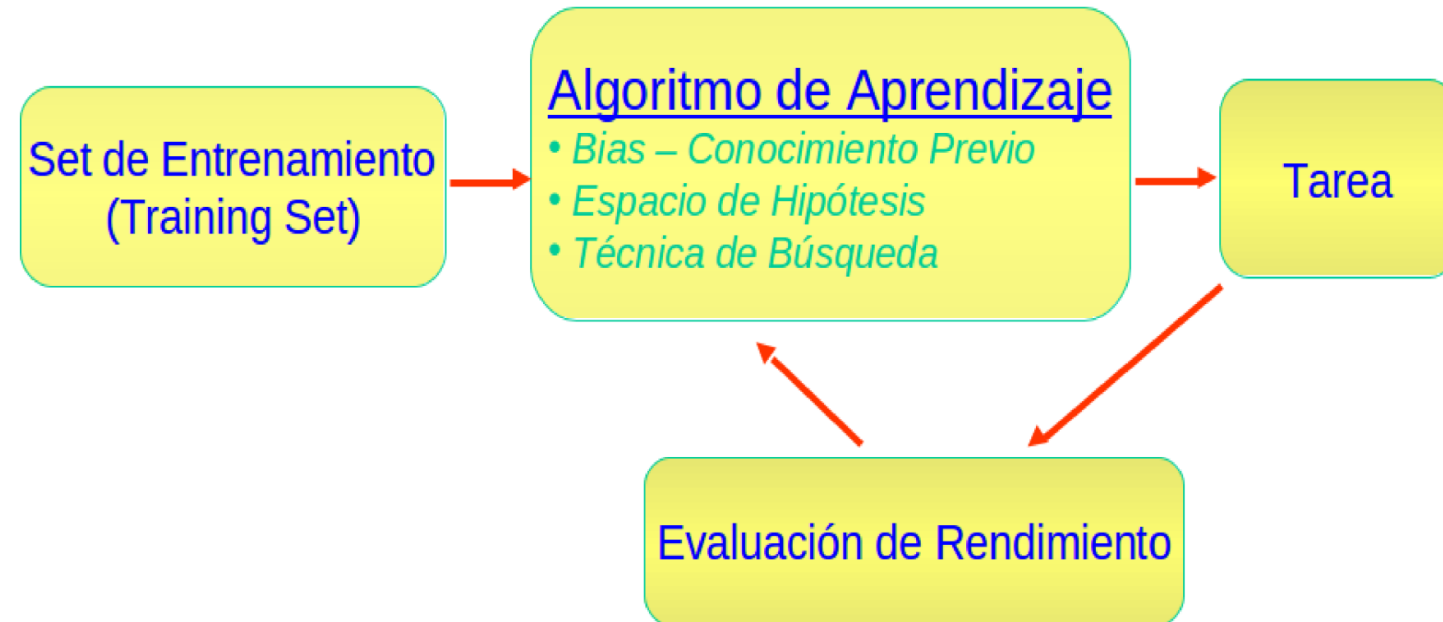
Dpto. Ingeniería de Transporte y Logística
Dpto. Ciencia de la Computación

Recordemos que Machine Learning se centra en **algoritmos** que **mejoran** su rendimiento en una tarea, a través de la **experiencia**



Buscamos la solución más adecuada en el **espacio de hipótesis**, usando **conocimiento previo** (sesgo inductivo) y datos de entrenamiento para guiar la **búsqueda**.

Recordemos que Machine Learning se centra en **algoritmos** que **mejoran** su rendimiento en una tarea, a través de la **experiencia**

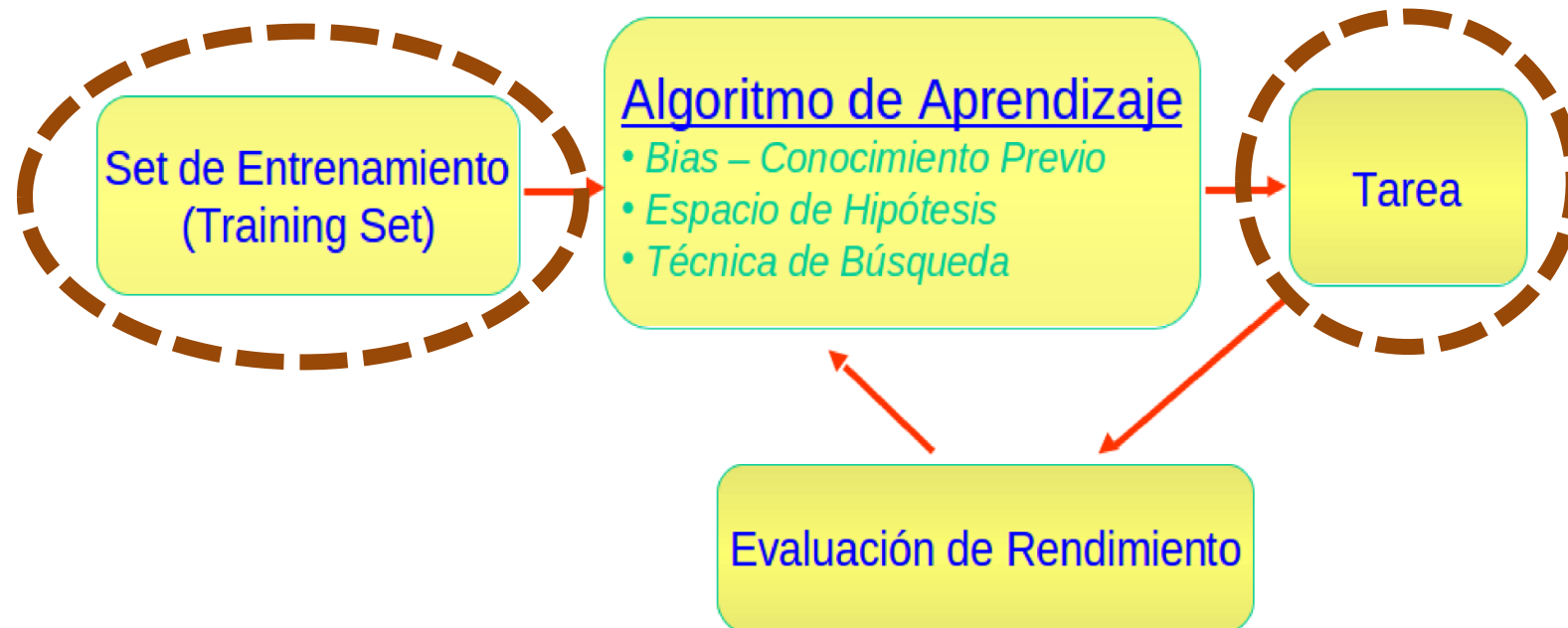


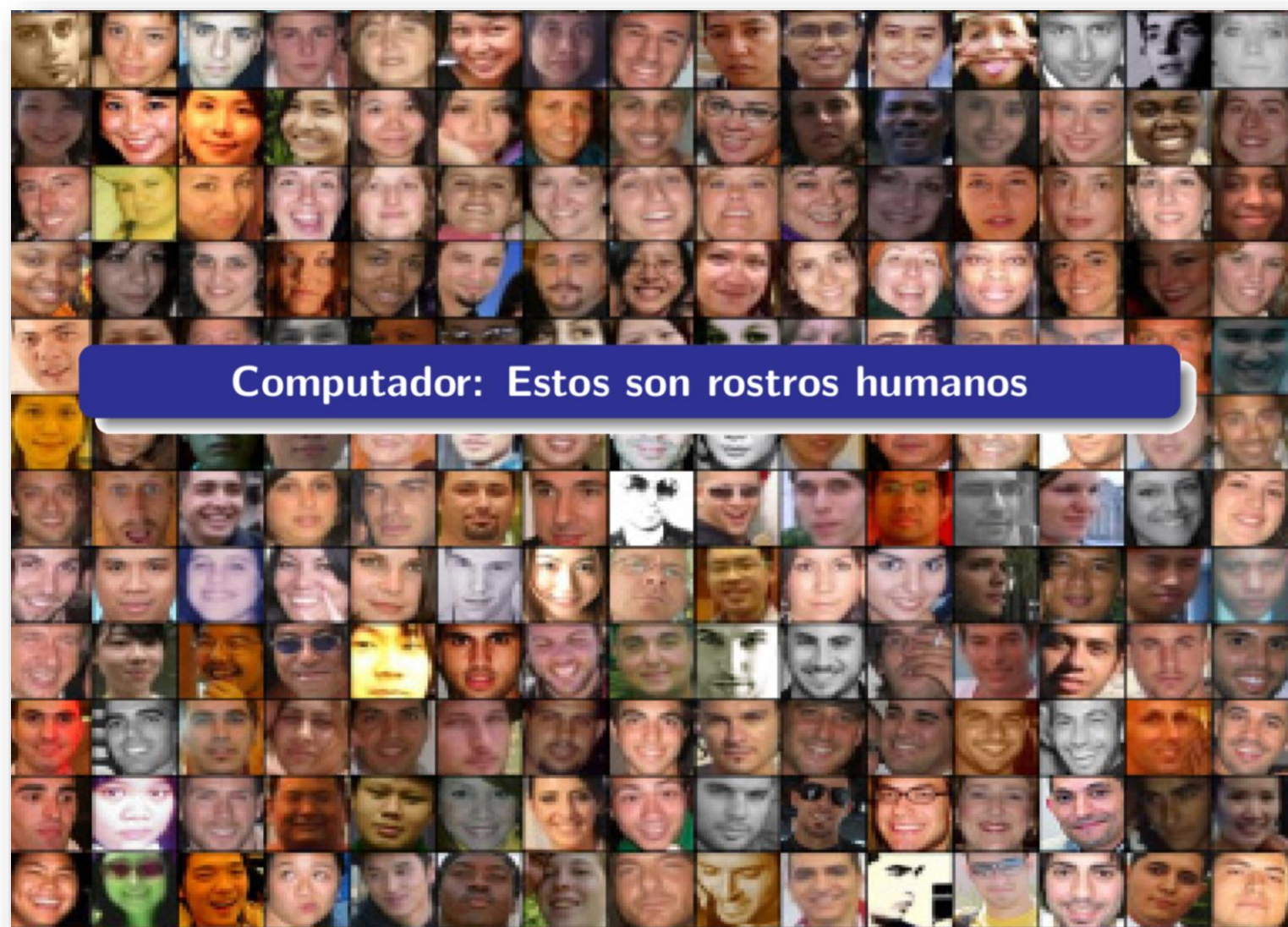
O visto de otra manera, debemos considerar 3 elementos centrales:
Representación (espacio hipótesis/sesgo inductivo), **Rendimiento** y **Optimización** (búsqueda)

Ok, súper lindo, pero, ¿cómo funciona?

¿Cómo podemos construir un detector de rostros?

¿Cómo podemos construir un detector de rostros?

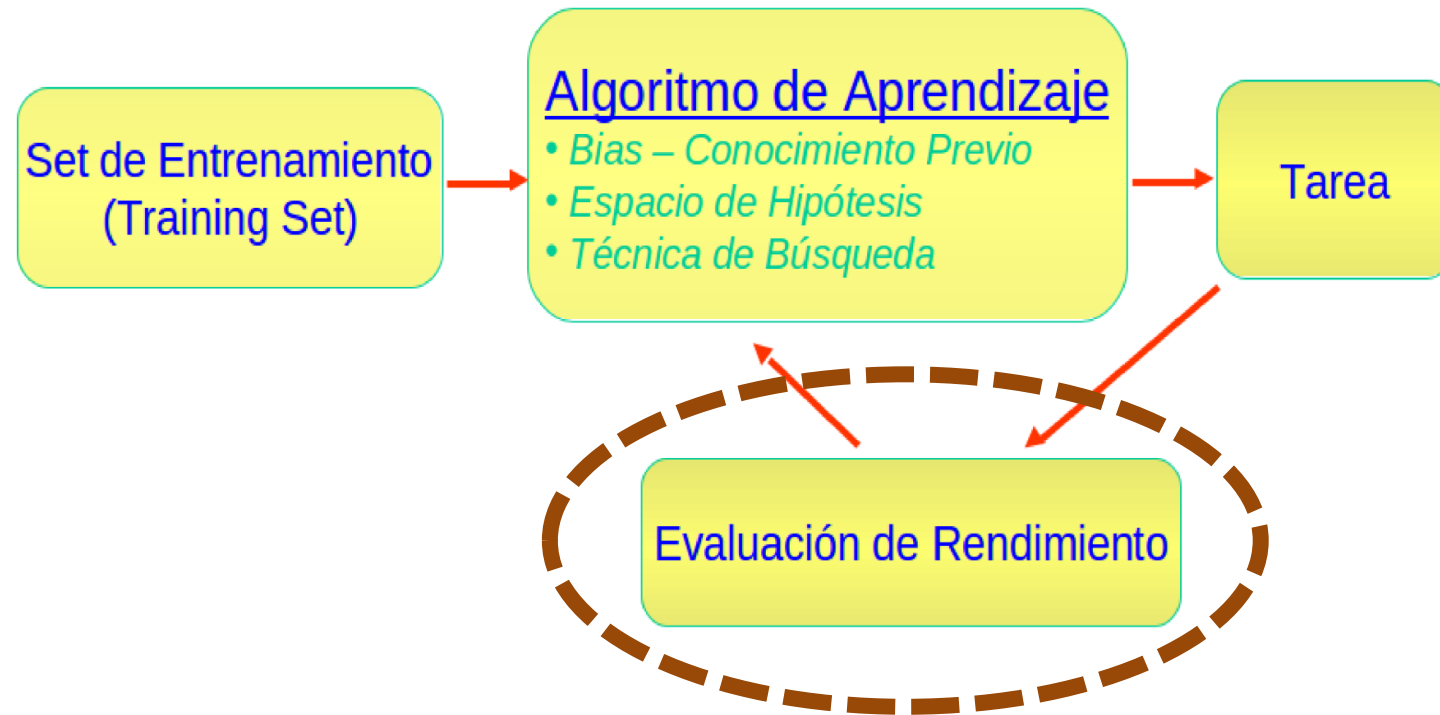






Computador: Estos **NO** son rostros

¿Cómo podemos construir un detector de rostros?



Computador: Algún rostro humano?



Veamos ahora un caso de estudio más avanzado e interesante

Image Inpainting for Irregular Holes Using Partial Convolutions



Veamos ahora un caso de estudio más avanzado e interesante

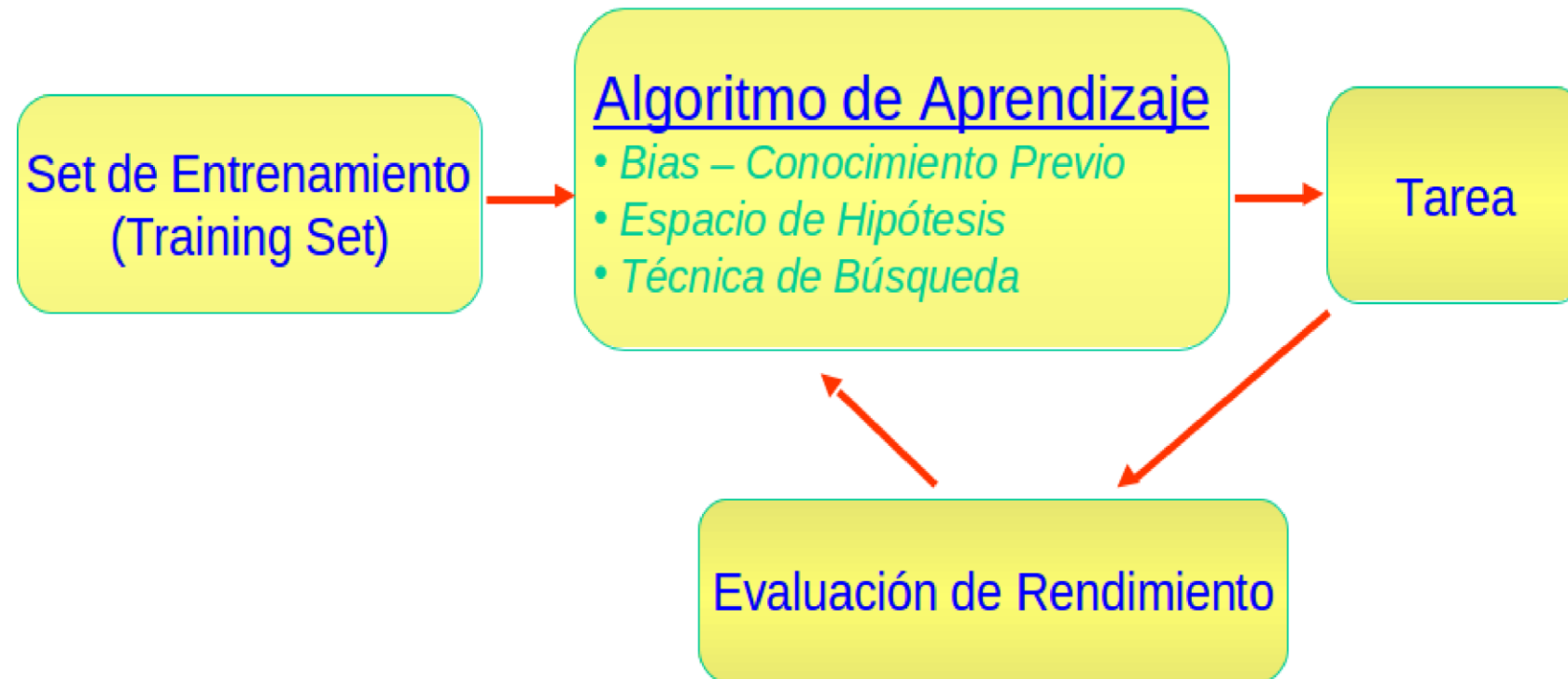
Image Inpainting for Irregular Holes Using Partial Convolutions

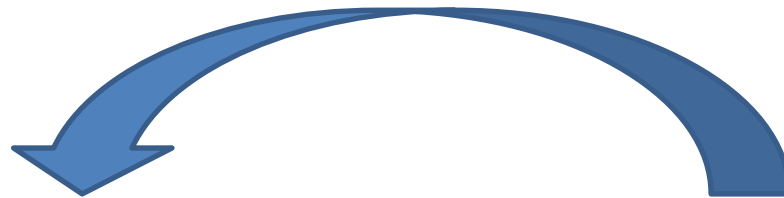
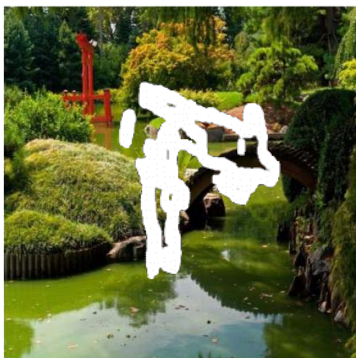


¿Qué es lo primero que necesitamos?



¿Cómo podemos estructurar el sistema de aprendizaje?





Algoritmo de aprendizaje supervisado



¿

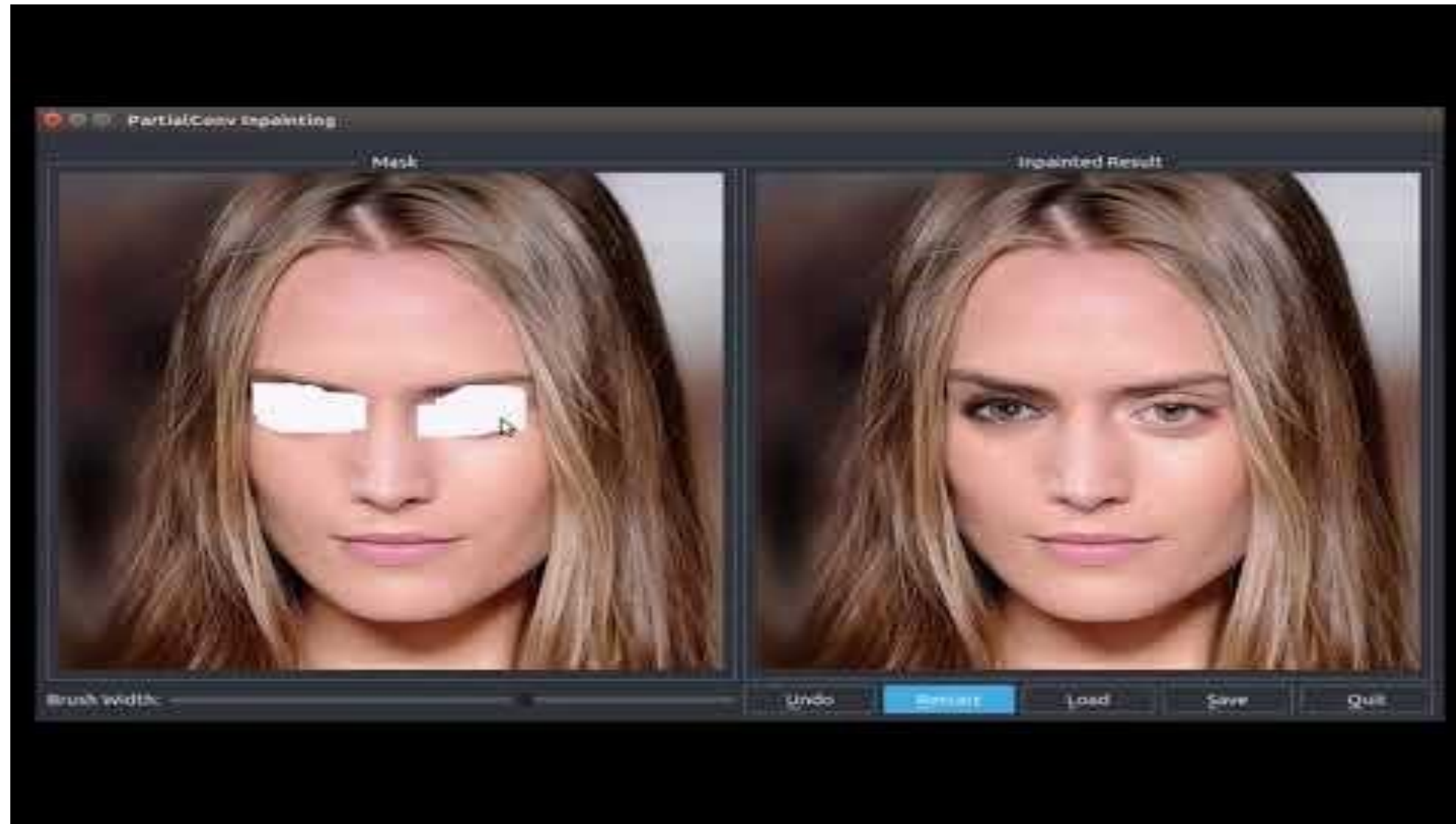


=



?

Veamos como funciona el sistema en la práctica
(<https://youtu.be/gg0F5JjKmhA>)



Algoritmos de ML trabajan sobre datos **multidimensionales**

- Cada dato esta caracterizado por una serie de **características = features = mediciones = atributos = variables**.
- La cantidad de **features** define la **dimensionalidad** del dato (vector).
- El espacio donde viven los datos (features) se conoce como **espacio de características** (*feature space*).

Wine Data Set

Download: [Data Folder](#), [Data Set Description](#)

Abstract: Using chemical analysis determine the origin of wines

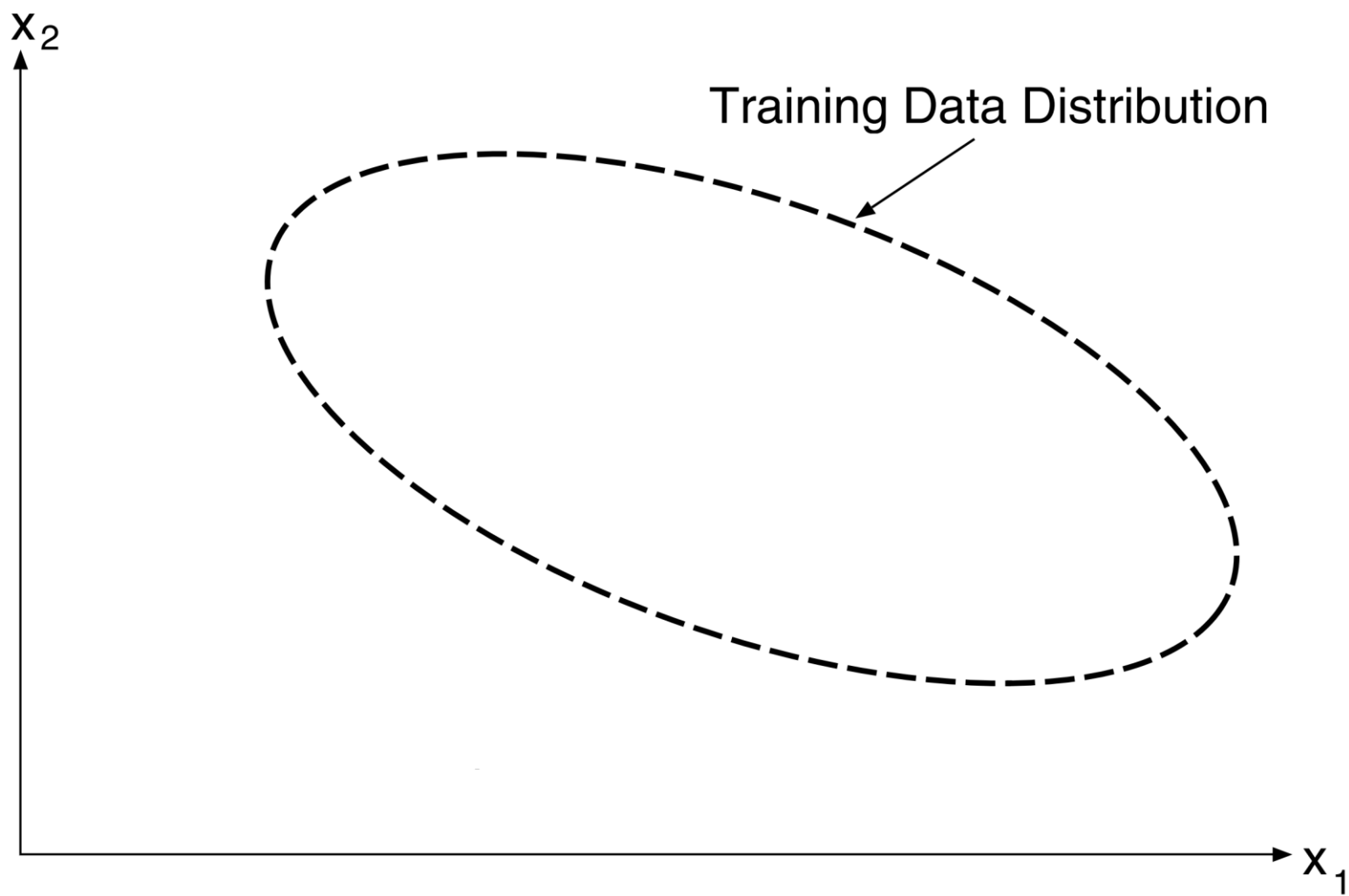


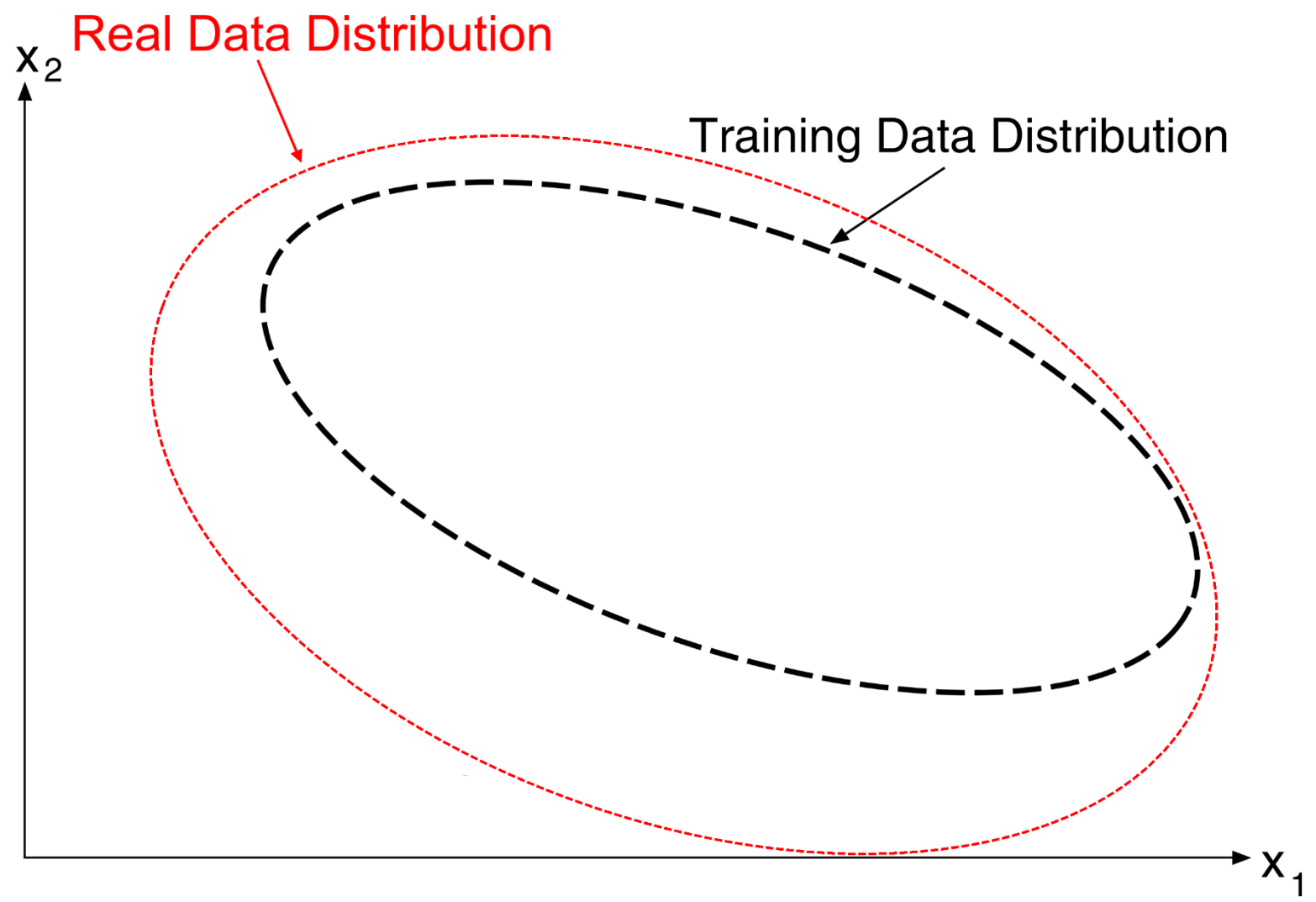
Data Set Characteristics:	Multivariate	Number of Instances:	178	Area:	Physical
Attribute Characteristics:	Integer, Real	Number of Attributes:	13	Date Donated	1991-07-01
Associated Tasks:	Classification	Missing Values?	No	Number of Web Hits:	573523

Para **entrenar** = ajustar = calibrar un modelo,
se utiliza un **set de entrenamiento**

Typhoon number	Input vectors				Response vector
	Distance from the eye of the storm (km)	Wind speed at site (m/s)	Pressure deficit at site (hPa)	Forward speed of the eye of the storm (km/h)	Storm surge (cm)
5111	96.0	20.7	20.6	27.6	47.4
5114	108.5	15.4	11.0	58.9	24.5
5201	181.2	8.1	1.7	40.1	7.9
5204	245.3	5.7	6.4	29.6	5.5
5209	117.5	23.3	22.0	46.6	61.7
5211	231.4	13.3	11.5	38.1	20.8
5309	293.6	4.0	7.2	35.4	5.6
5508	0.6	8.5	7.0	32.2	8.7
5512	227.6	10.0	10.4	19.3	16.0
5609	257.3	11.5	15.0	44.1	10.8

Cada dato (fila) del set de entrenamiento, puede considerarse
como un **vector** en el espacio de características.

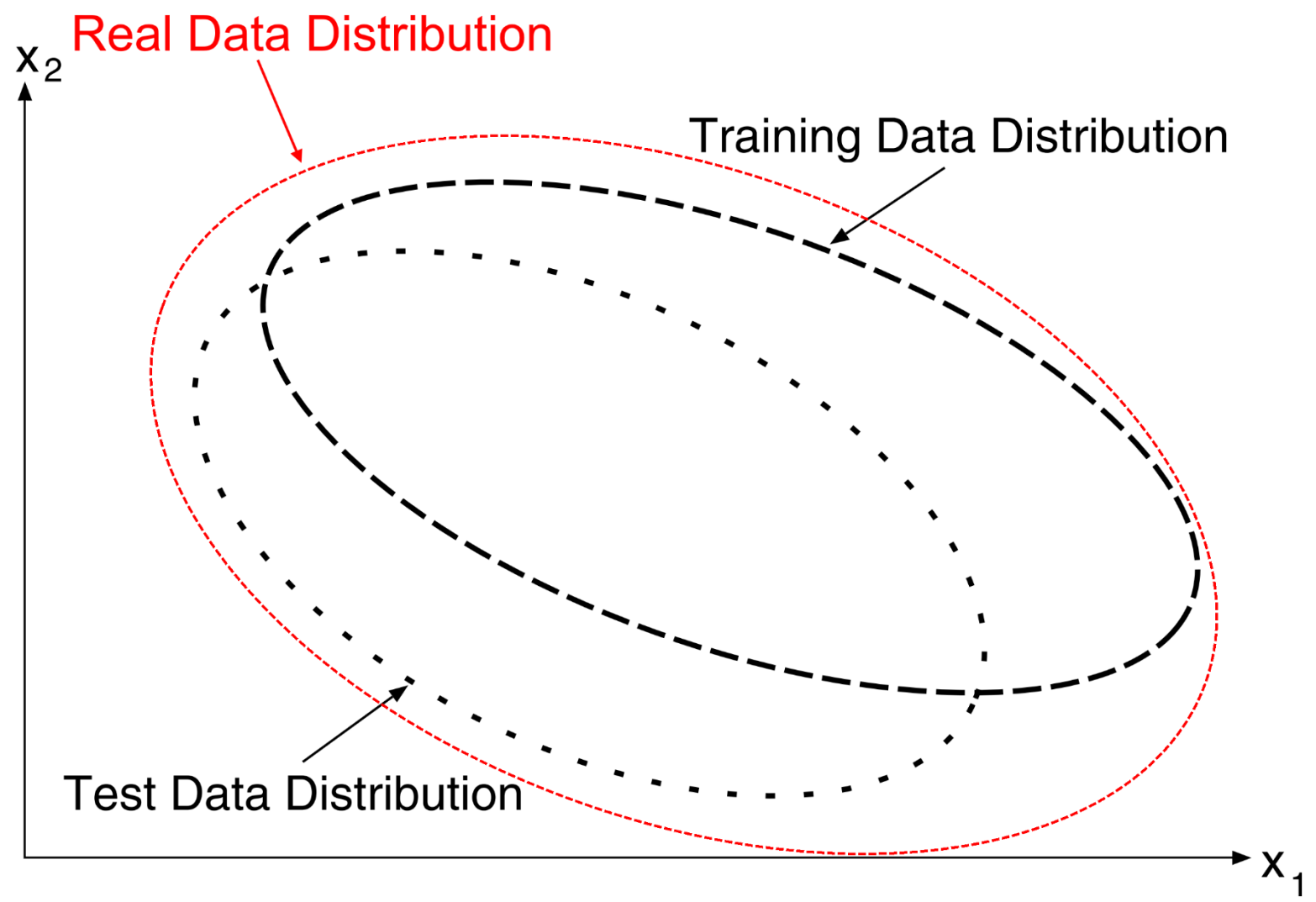


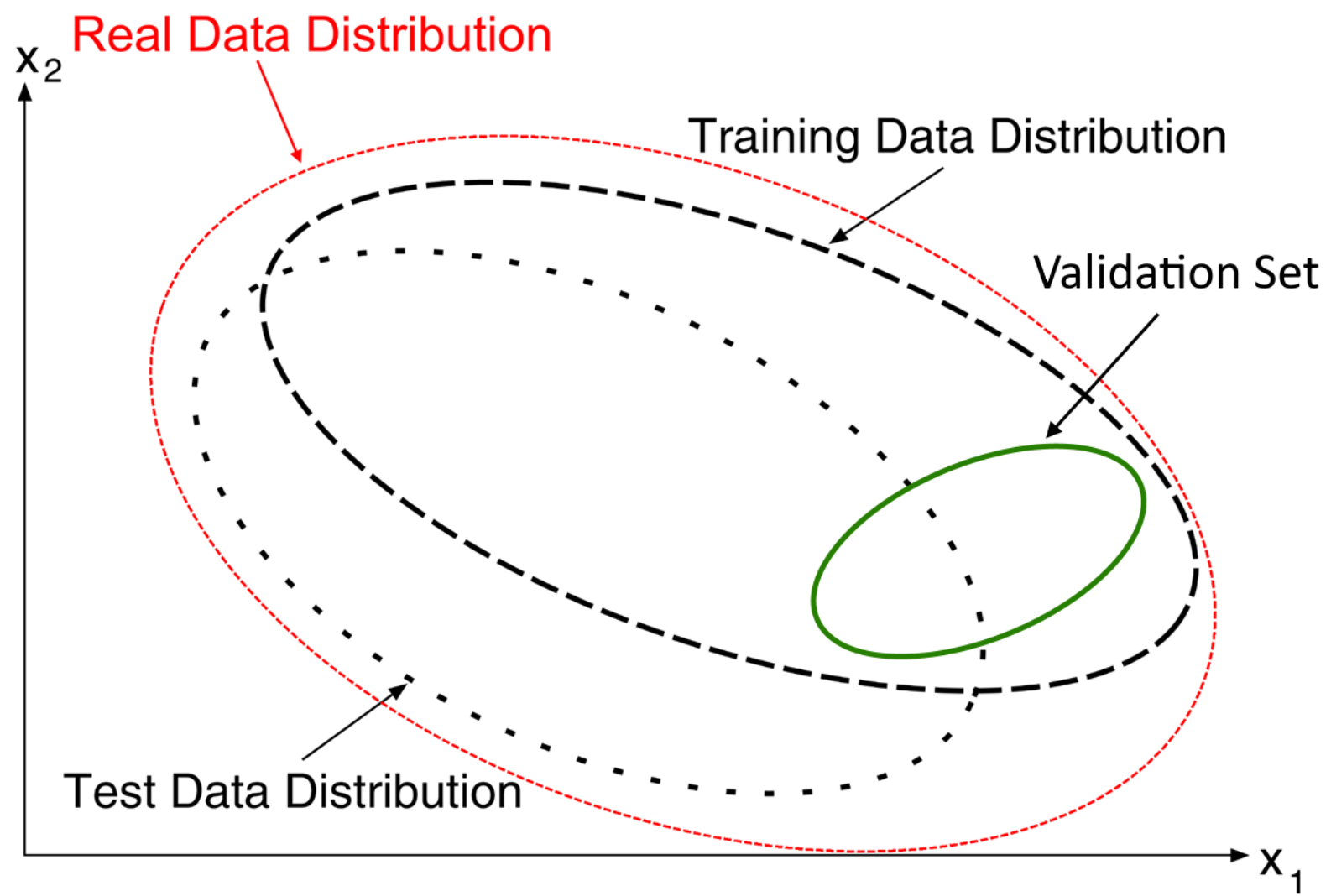


Objetivo último es la **generalización** (aprendizaje inductivo)

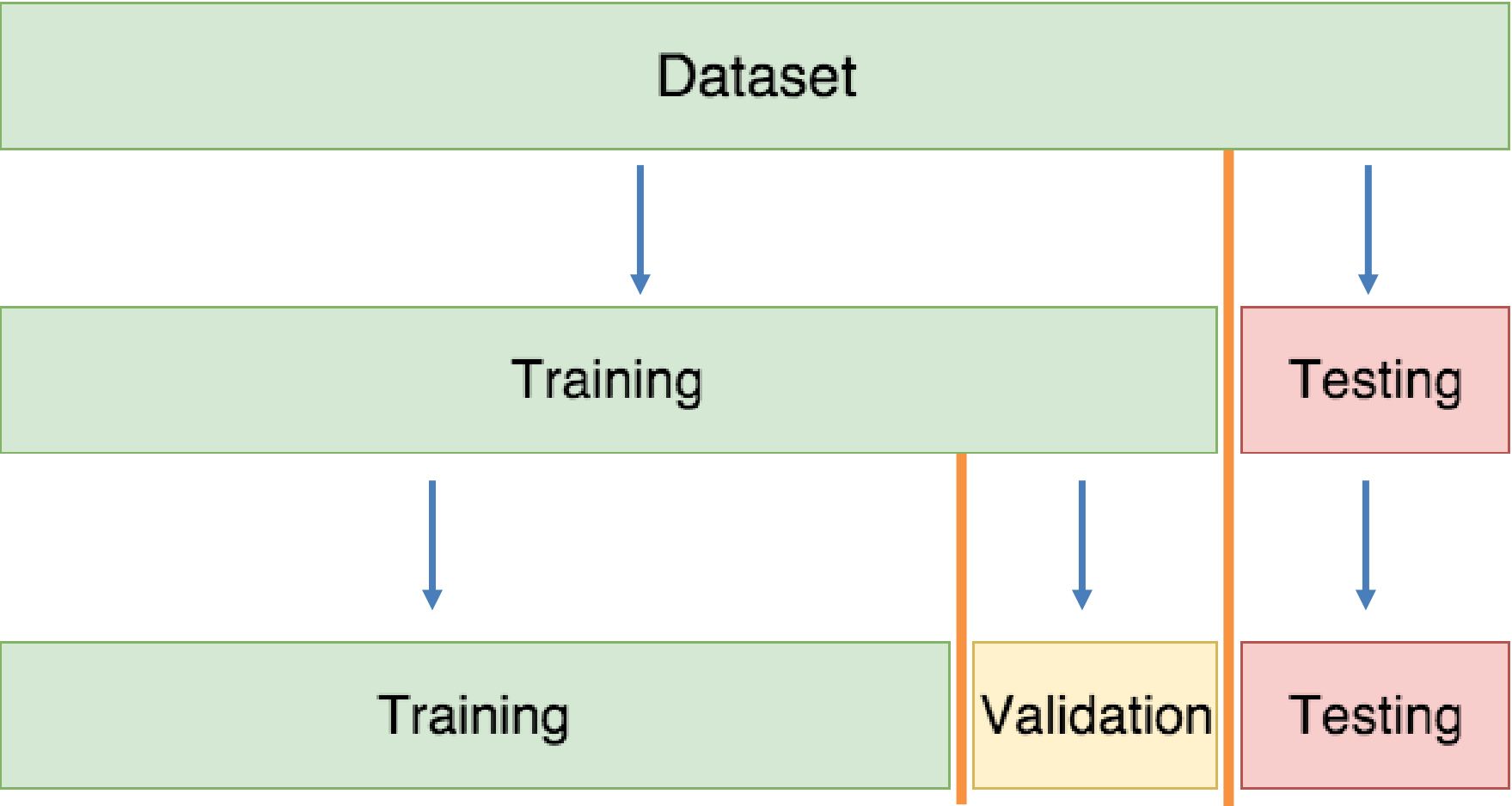
	Typhoon number	Input vectors				Response vector
		Distance from the eye of the storm (km)	Wind speed at site (m/s)	Pressure deficit at site (hPa)	Forward speed of the eye of the storm (km/h)	Storm surge (cm)
Entrenamiento	5111	96.0	20.7	20.6	27.6	47.4
	5114	108.5	15.4	11.0	58.9	24.5
	5201	181.2	8.1	1.7	40.1	7.9
	5204	245.3	5.7	6.4	29.6	5.5
	5209	117.5	23.3	22.0	46.6	61.7
	5211	231.4	13.3	11.5	38.1	20.8
	5309	293.6	4.0	7.2	35.4	5.6
	5508	0.6	8.5	7.0	32.2	8.7
	5512	227.6	10.0	10.4	19.3	16.0
	5609	257.3	11.5	15.0	44.1	10.8
Test	0209	290.6	9.5	13.6	46.9	?
	0215	245.3	10.6	14.2	77.6	
	0306	227.0	4.4	7.9	20.8	
	0314	279.1	4.4	7.8	29.5	
	0415	266.3	8.7	8.8	32.9	
	0515	165.6	19.2	16.4	45.6	
	0601	136.5	10.7	12.2	4.6	
	0603	207.9	4.4	8.0	14.1	

Set de test es útil para evaluar la capacidad de generalización del modelo





Otra forma de verlo, con conjuntos disjuntos (no distribuciones de probabilidad)



El set de datos **MNIST** permite construir **clasificadores de dígitos** a partir de imágenes (OCR)



- MNIST es un set de datos compuesto de imágenes de dígitos escritos a mano.
- Cada imagen muestra un sólo dígito entre 0 y 9. Las imágenes son binarias con una resolución de $28 \times 28 = 784$ píxeles.
- El dataset consta de 60.000 ejemplos de entrenamiento y 10.000 de test.

El set de datos **MNIST** permite construir **clasificadores de dígitos** a partir de imágenes (OCR)



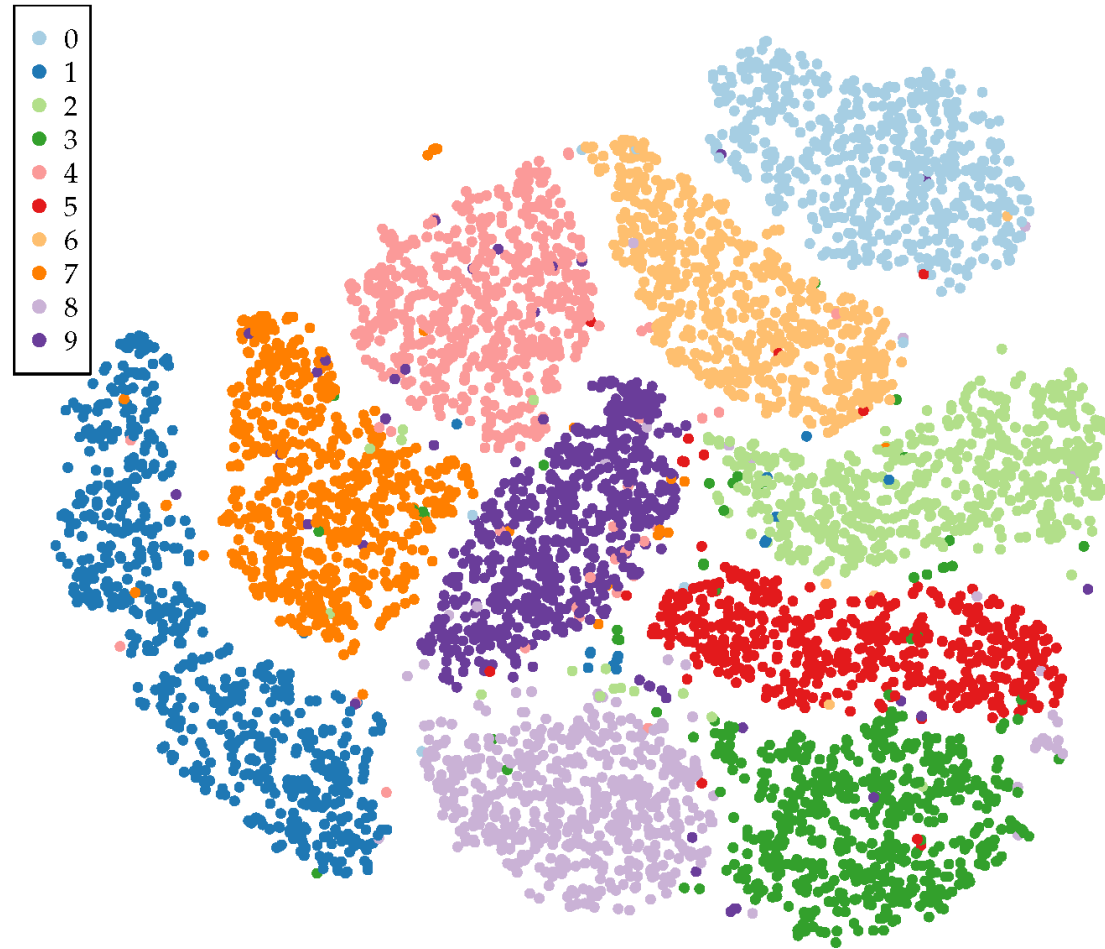
- ¿Cómo podríamos resolver este problema (clasificación de dígitos)?
- Quizá, visualizar el espacio de características nos da una pista.

¿Es posible **visualizar** directamente el espacio de características de MNIST?

No

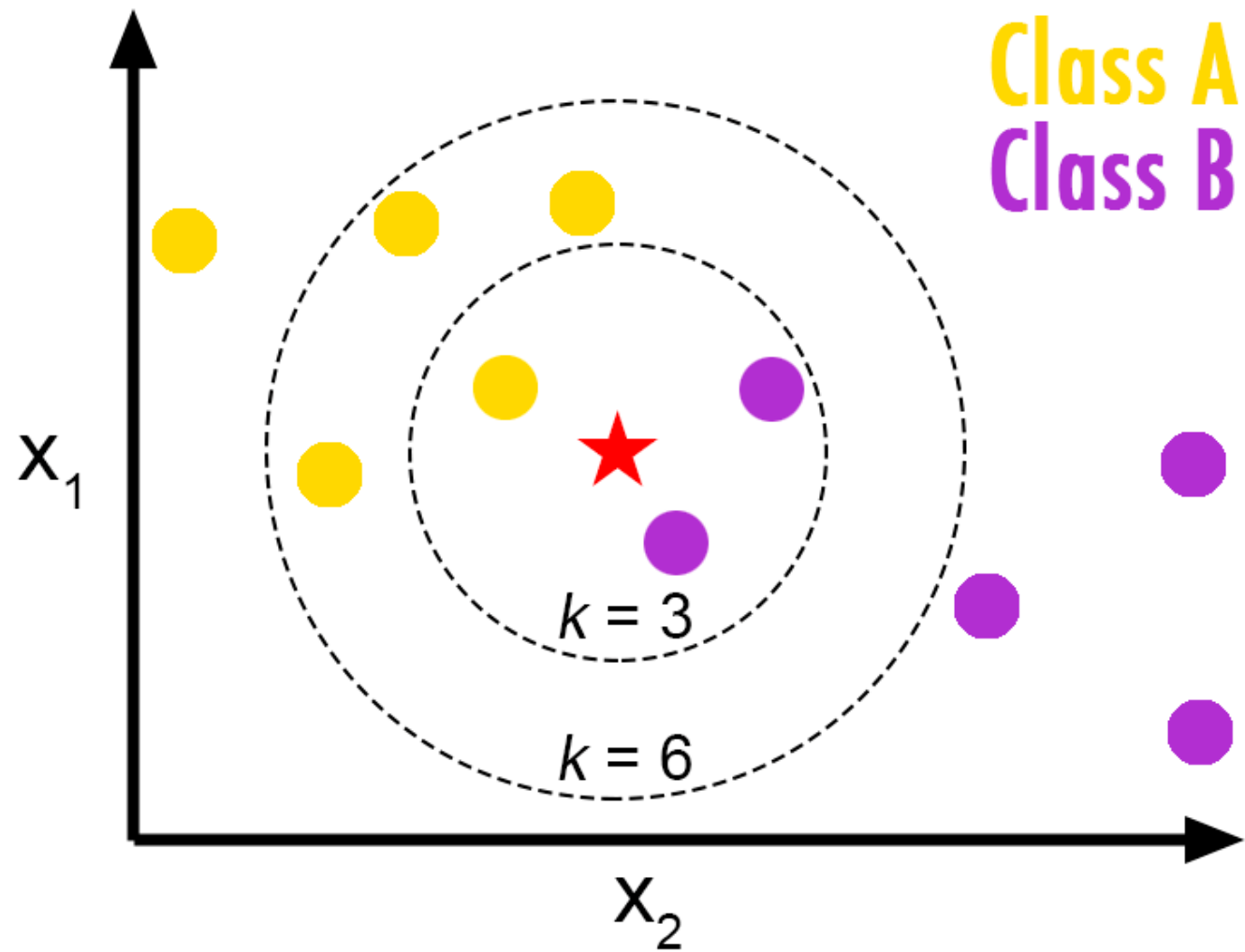
¿Por qué?

Usando técnicas **no supervisadas** de reducción de dimensionalidad (tSNE*), es posible **transformar** el espacio de características



¿Cómo podríamos resolver este problema?
(aka cuál es el algoritmo más simple que podríamos usar)

Clasificador de **k-vecinos cercanos** permite realizar la tarea de manera intuitiva



Clasificador de **k-vecinos cercanos** permite realizar la tarea de manera intuitiva

- Dado un dato sin clasificar, su clase se define como el resultado de la votación de los k-vecinos más cercanos.
- Con k=1, se obtienen los siguientes resultados:

Real

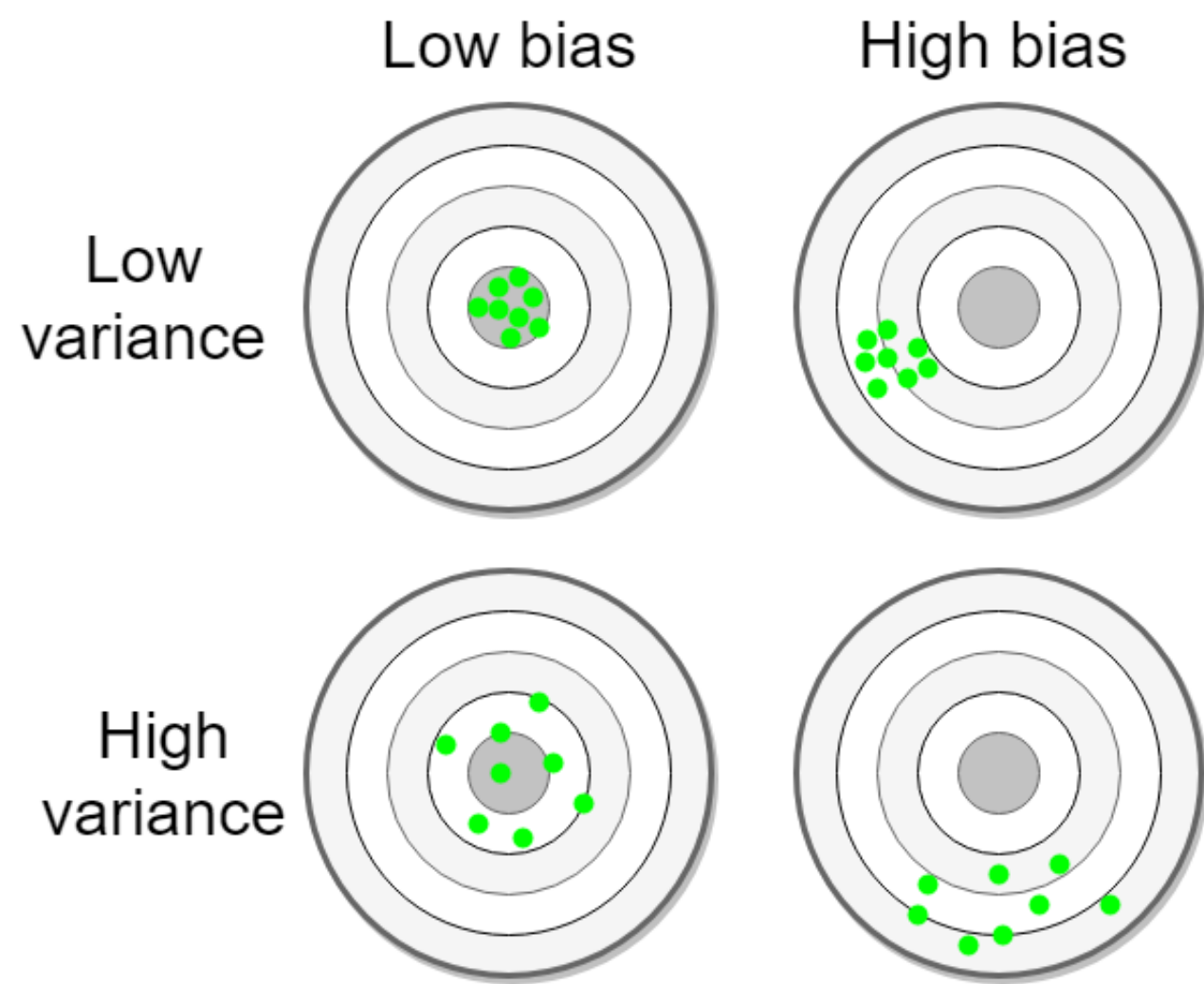
	Predicción									
	0	1	2	3	4	5	6	7	8	9
0	972	1	1	0	0	1	3	1	0	0
1	0	1129	3	0	1	1	1	0	0	0
2	7	6	992	5	1	0	2	16	3	0
3	0	1	2	970	1	19	0	7	7	3
4	0	7	0	0	944	0	3	5	1	22
5	1	1	0	12	2	860	5	1	6	4
6	4	2	0	0	3	5	944	0	0	0
7	0	14	6	2	4	0	0	992	0	10
8	6	1	3	14	5	13	3	4	920	5
9	2	5	1	6	10	5	1	11	1	967

Vamos a Colab...



A pesar de ser clave, el **set de entrenamiento** no lo es todo

- En general, los algoritmos de aprendizaje viven y mueren por el set de entrenamiento.
- Lamentablemente, tener un buen set de entrenamiento, **no asegura siempre tener buena generalización**.
- Poder de representación del algoritmo de aprendizaje pasa a ser también un tema central.
- El porqué de esto está dado por un problema llamado **Bias-Variance Tradeoff**



BV tradeoff se da de **forma natural** en ML

$$Y = f(x) + \varepsilon, \varepsilon \sim N(0, \sigma^2)$$

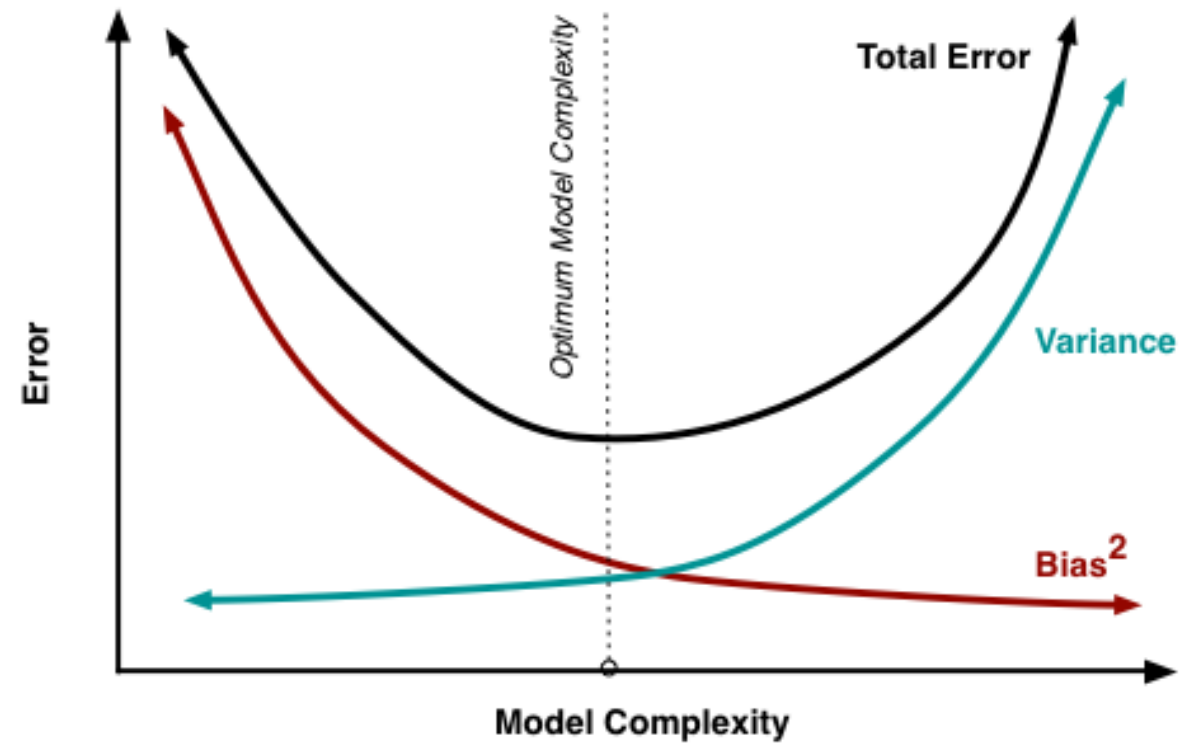
$$f(x) \approx \hat{f}(x)$$

$$Err(x) = E \left[\left(Y - \hat{f}(x) \right)^2 \right]$$

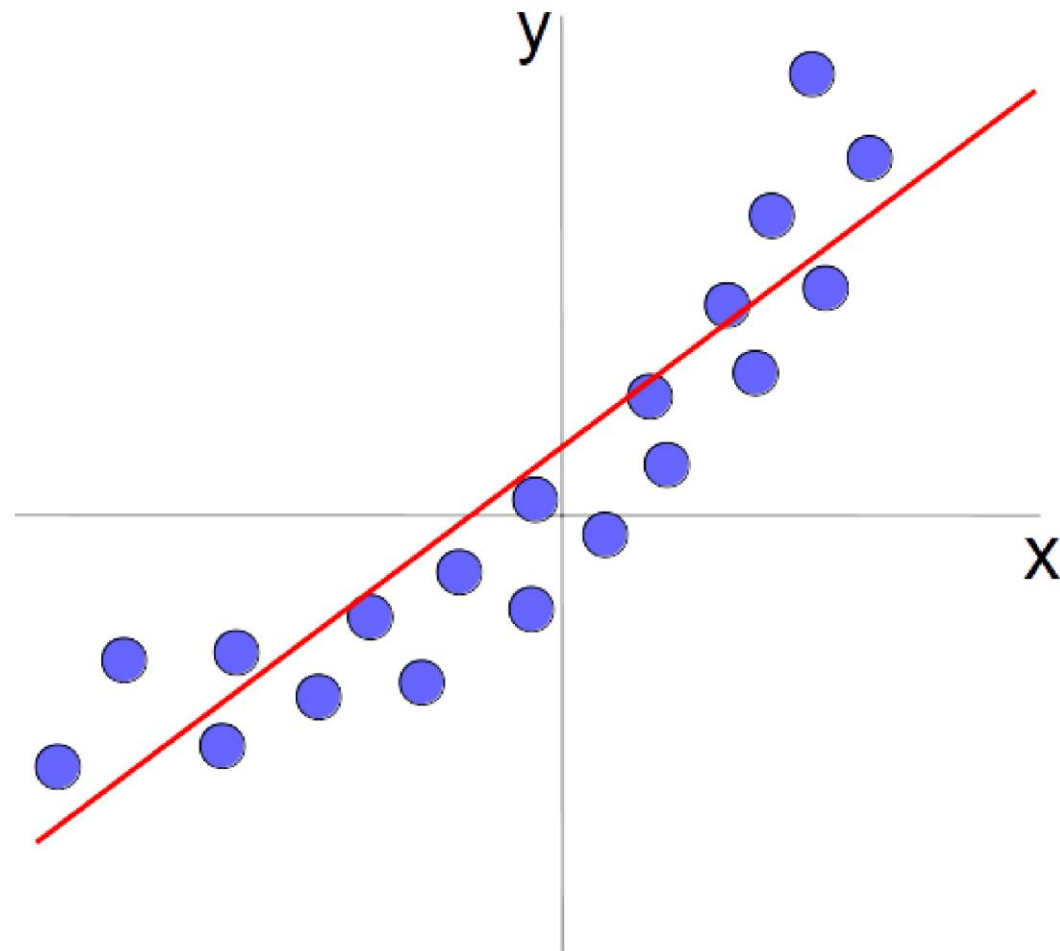
$$Err(x) = \left(E[\hat{f}(x)] - f(x) \right)^2 + E \left[\left(\hat{f}(x) - E[\hat{f}(x)] \right)^2 \right] + \sigma^2$$

$$Err(x) = Bias^2 + Varianza + Error\ irreducible$$

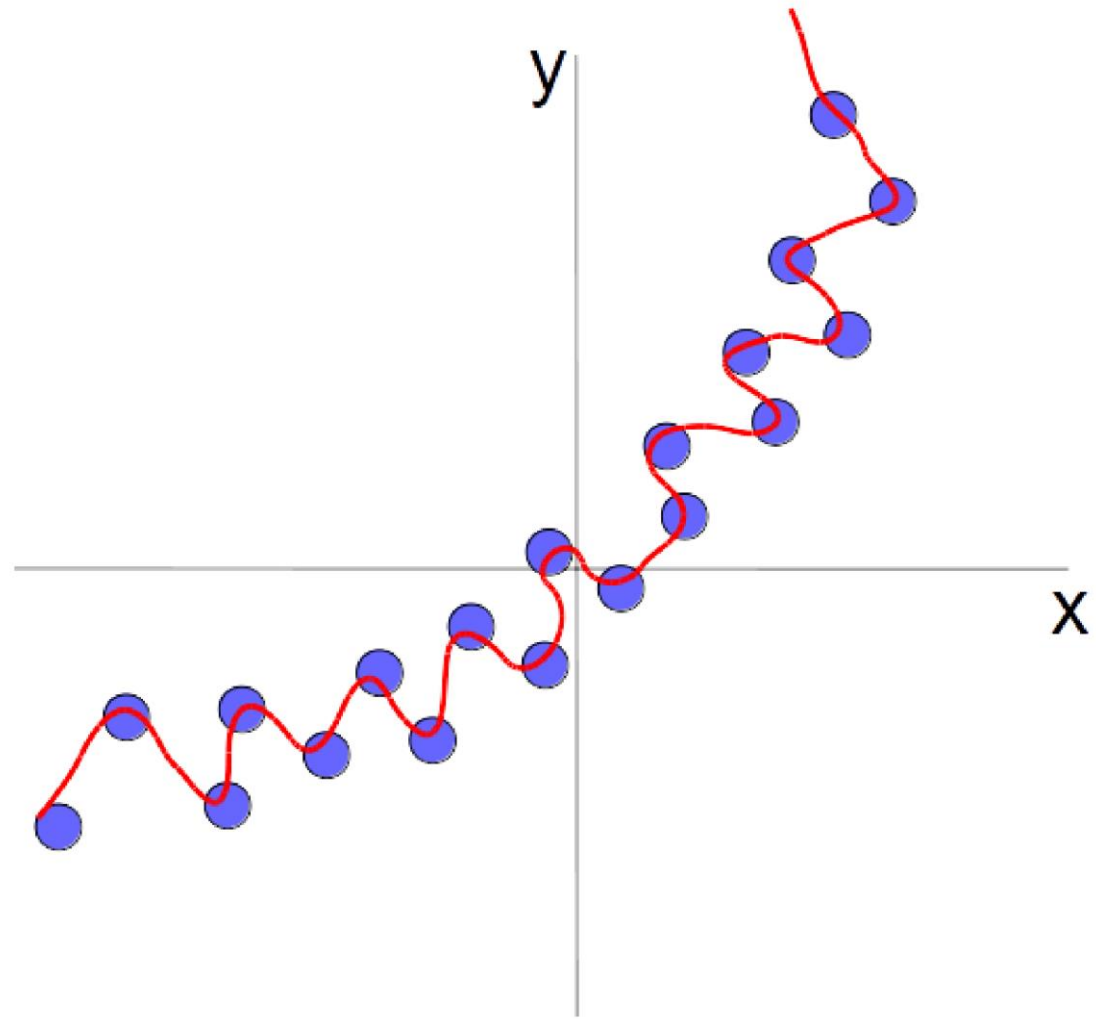
Complejidad del modelo es el parámetro que permite capturar el *BV tradeoff*



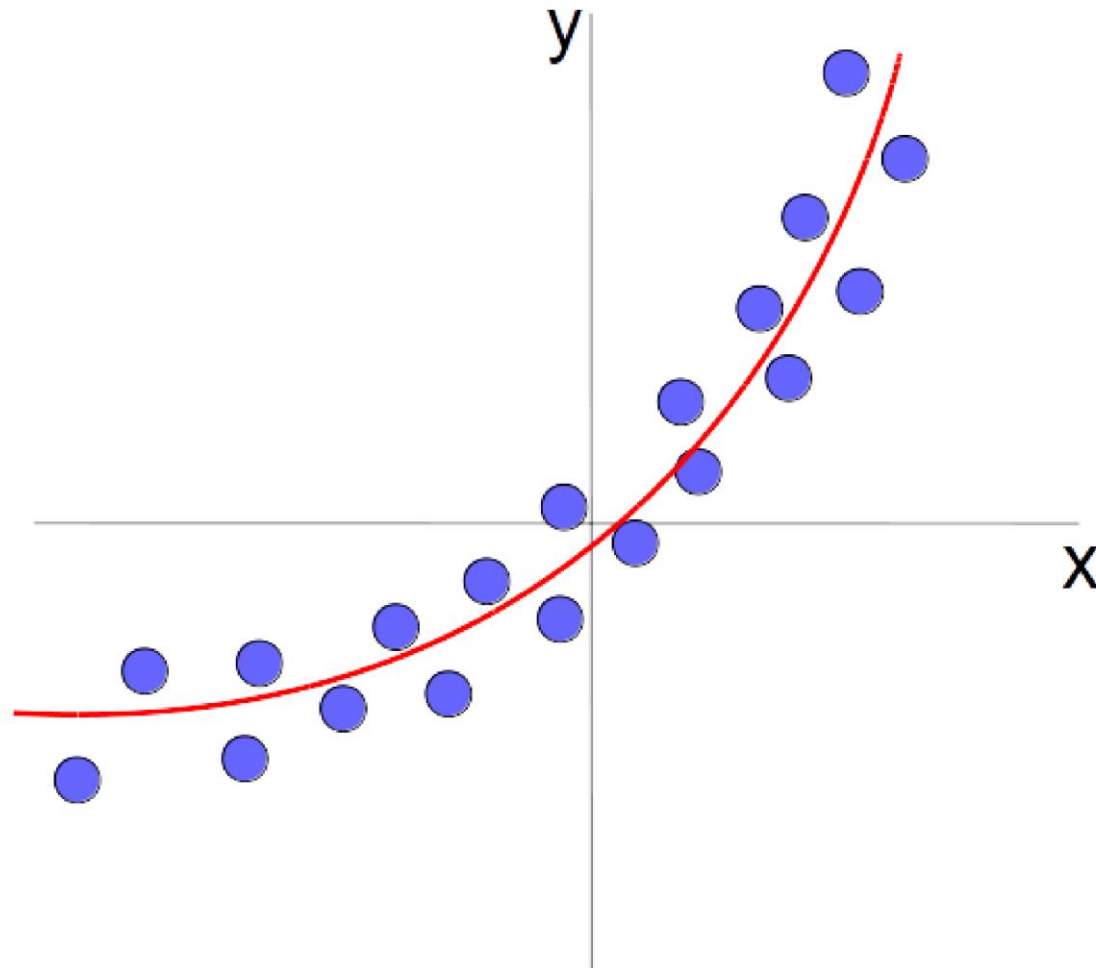
$$Err(x) = Bias^2 + Varianza + Error\ irreducible$$



Modelo es demasiado simple para capturar el comportamiento de los datos (*underfitting, alto sesgo*).



Modelo es muy complejo, y captura hasta el ruido presente en los ejemplos (*overfitting, alta varianza*).



Modelo tiene la complejidad necesaria para capturar los patrones relevantes, **controlando sesgo y varianza**.

Vamos a Colab...



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación



Sistemas Urbanos Inteligentes

Fundamentos de Machine Learning Parte 2

Hans Löbel

Dpto. Ingeniería de Transporte y Logística
Dpto. Ciencia de la Computación