

# Patrones de Diseño en Machine Learning

Introducción y patrones de representación de datos

# ¿Qué es un patrón de diseño?



- ¿Qué entienden por *patrón de diseño*?
- ¿En qué disciplinas han escuchado/visto/usado esta idea?

# ¿Qué es un patrón de diseño?



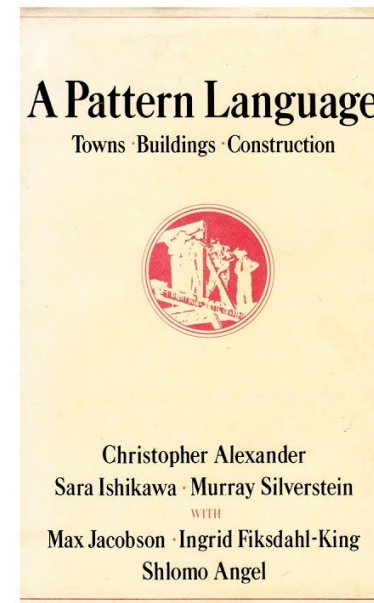
OOP Design Patterns			
Creational Patterns	Abstract Factory	Builder	Prototype
	Singleton	Factory Method	
Structural Patterns	Adapter	Bridge	Composite
	Decorator	Facade	Flyweight
	Proxy		
Behavioral Patterns	Chain of Responsibility	Command	Iterator
	Mediator	Memento	Observer
	State	Strategy	Visitor
	Interpreter	Template Method	

prasadct.com



# ¿Qué es un patrón de diseño?

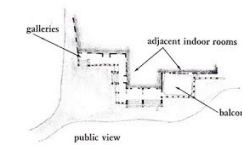
- La idea de patrón de diseño se introdujo por primera vez por Christopher Alexander et al. en el libro *A Pattern Language*.
- Un patrón **describe un problema recurrente para después describir la esencia de su solución** de tal manera que se pueda usar un millón de veces sin hacerlo de la misma forma más de una vez.



## 167 SIX-FOOT BALCONY\*\*

Therefore:

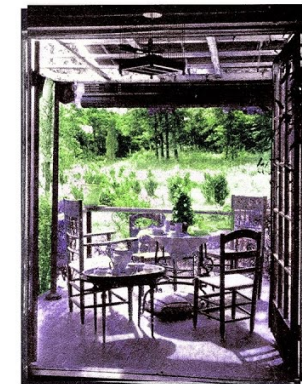
Whenever possible, and at every story, build porches, galleries, arcades, balconies, niches, outdoor seats, awnings, trellised rooms, and the like at the edges of buildings—especially where they open off public spaces and streets, and connect them by doors, directly to the rooms inside.



A warning: take care that such places are not stuck artificially onto the building. Keep them real; find the places along the building edge that offer a direct and useful connection with the life indoors—the space outside the stair landing, the space to one side of the bedrooms above, and so on.

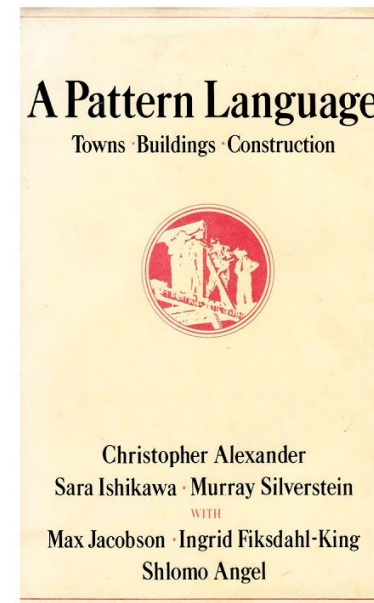
These places should be an integral part of the building territory, and contain seats, tables, furniture, places to stand and talk, places to work outside—all in the public view—PRIVATE TERRACE ON THE STREET (140), OUTDOOR ROOM (163); make the spaces deep enough to be really useful—SIX-FOOT BALCONY (167)—with columns heavy enough to provide at least partial enclosure—HALF-OPEN WALL (193), COLUMN PLACES (226). . . .

## 167 SIX-FOOT BALCONY\*\*



# ¿Qué es un patrón de diseño?

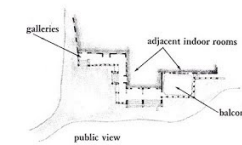
- Cada solución se plantea de tal forma que proporciona las relaciones del campo esenciales para resolver el problema, pero de manera general y abstracta.
- Esto con el fin de que nosotros resolvamos el problema que enfrentamos sujeto a nuestras preferencias y condiciones particulares del mismo.



## 167 SIX-FOOT BALCONY\*\*

Therefore:

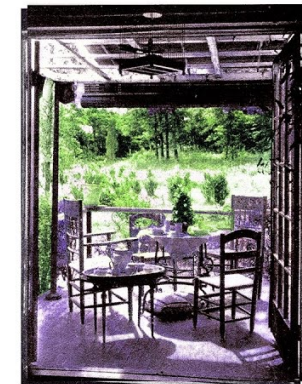
Whenever possible, and at every story, build porches, galleries, arcades, balconies, niches, outdoor seats, awnings, trellised rooms, and the like at the edges of buildings—especially where they open off public spaces and streets, and connect them by doors, directly to the rooms inside.



A warning: take care that such places are not stuck artificially onto the building. Keep them real; find the places along the building edge that offer a direct and useful connection with the life indoors—the space outside the stair landing, the space to one side of the bedrooms above, and so on.

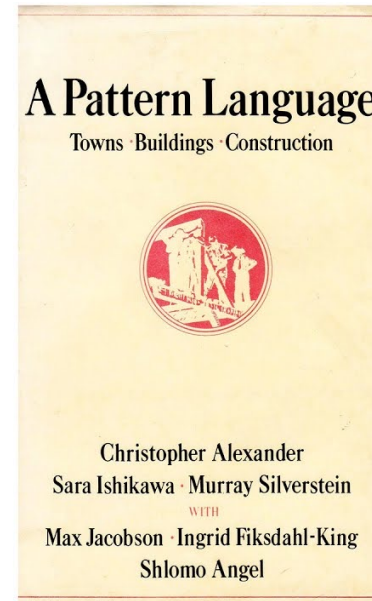
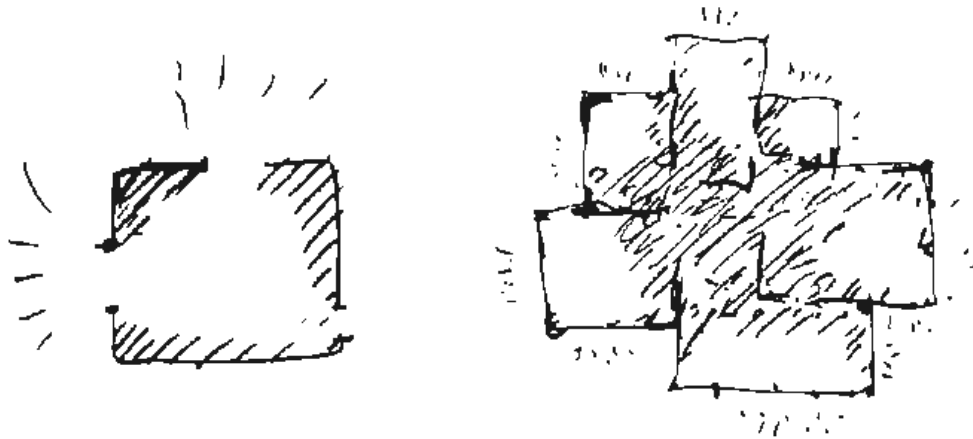
These places should be an integral part of the building territory, and contain seats, tables, furniture, places to stand and talk, places to work outside—all in the public view—PRIVATE TERRACE ON THE STREET (140), OUTDOOR ROOM (163)—make the spaces deep enough to be really useful—SIX-FOOT BALCONY (167)—with columns heavy enough to provide at least partial enclosure—HALF-OPEN WALL (193), COLUMN PLACES (226) . . .

## 167 SIX-FOOT BALCONY\*\*



# ¿Qué es un patrón de diseño?

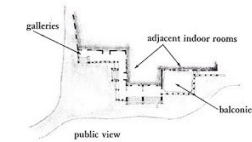
each room has light on two sides



167 SIX-FOOT BALCONY\*\*

Therefore:

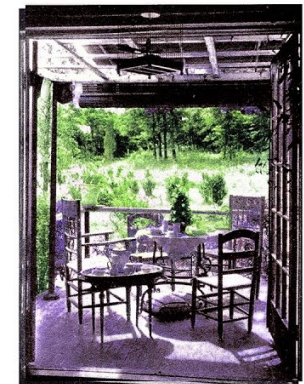
Whenever possible, and at every story, build porches, galleries, arcades, balconies, niches, outdoor seats, awnings, trellised rooms, and the like at the edges of buildings—especially where they open off public spaces and streets, and connect them by doors, directly to the rooms inside.



A warning: take care that such places are not stuck artificially onto the building. Keep them real; find the places along the building edge that offer a direct and useful connection with the life indoors—the space outside the stair landing, the space to one side of the bedrooms above, and so on.

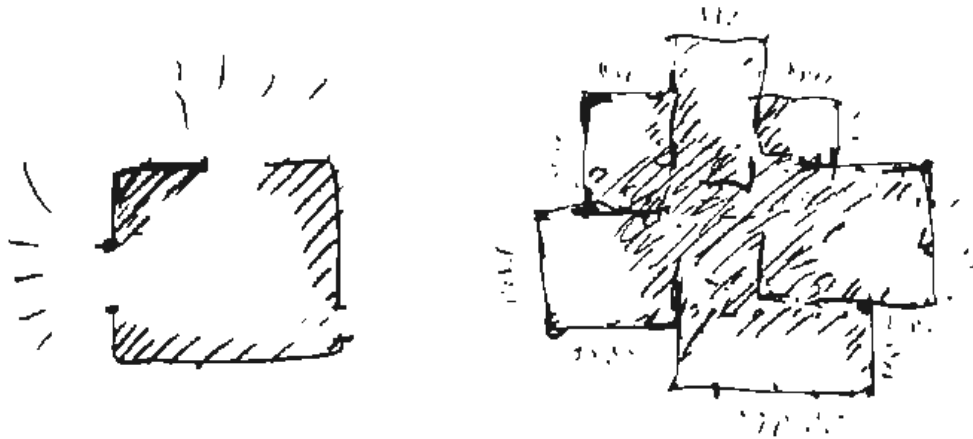
These places should be an integral part of the building territory, and contain seats, tables, furniture, places to stand and talk, places to work outside—all in the public view—PRIVATE TERRACE ON THE STREET (140), OUTDOOR ROOM (163); make the spaces deep enough to be really useful—SIX-FOOT BALCONY (167)—with columns heavy enough to provide at least partial enclosure—HALF-OPEN WALL (193), COLUMN PLACES (226) . . .

167 SIX-FOOT BALCONY\*\*



# ¿Qué es un patrón de diseño?

each room has light on two sides



- La idea de los patrones es crear plantillas o maquetas para que se usen, en lugar de redescubrir el conocimiento.
- Cómo lo usamos es el toque que cada persona le da a la solución del problema.
- También se pueden ver como *buenas prácticas*.

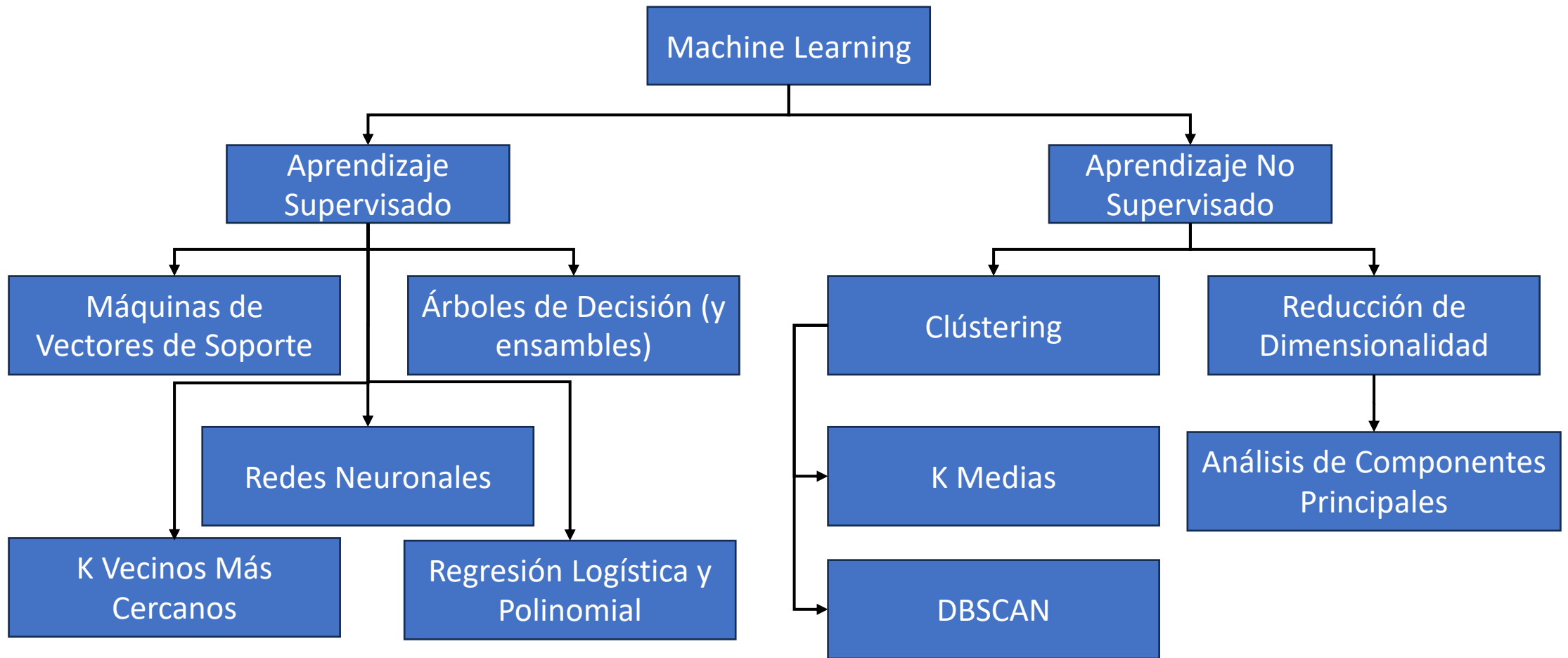
# ¿Qué es un patrón de diseño?

- ¿Se puede aplicar la idea de patrones de diseño en Machine Learning?
- ¿Qué ventajas nos ofrecen?
- ¿Cómo creen que se apliquen en este campo?



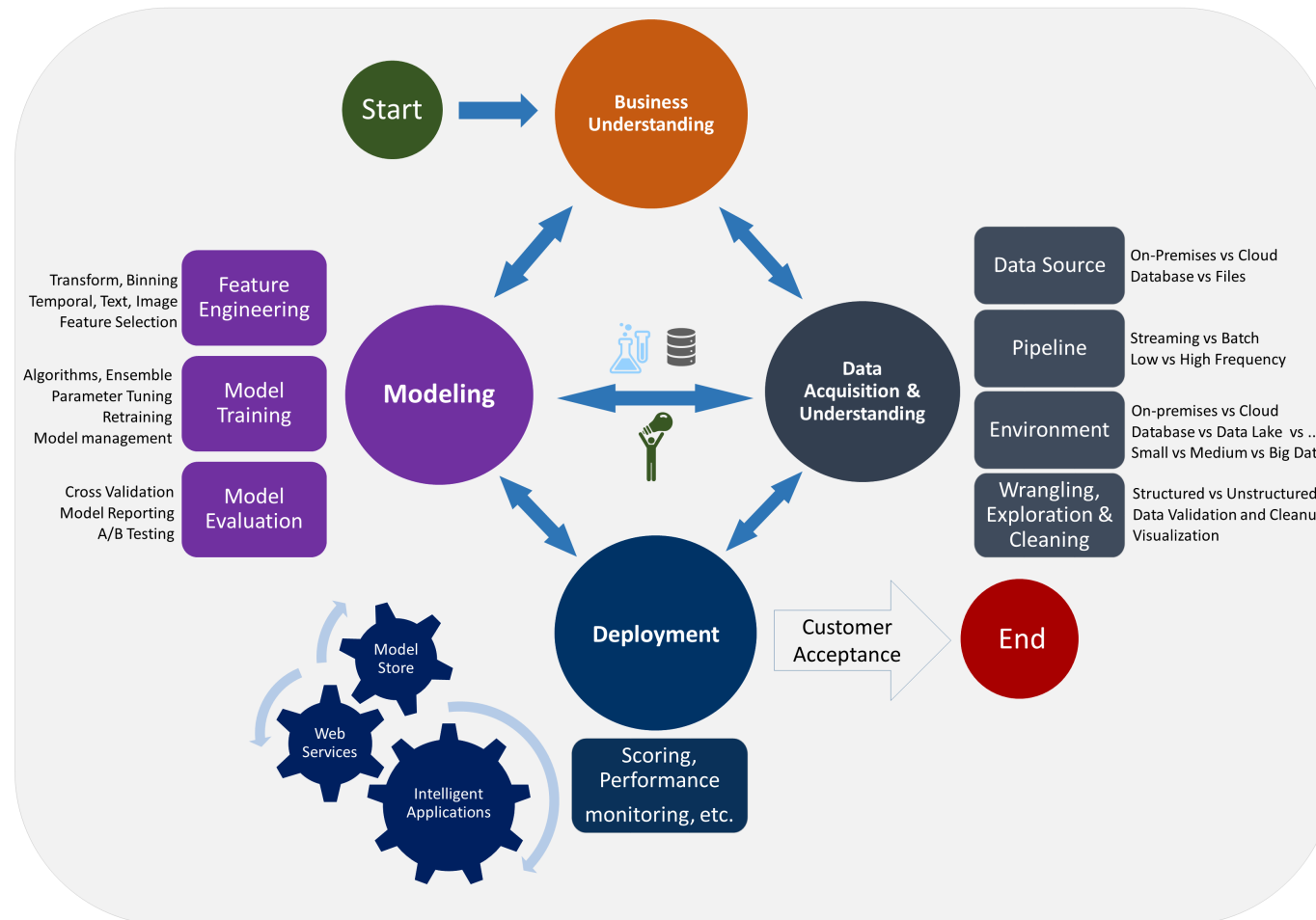
# Patrones de diseño y Machine Learning

# Patrones de diseño y Machine Learning

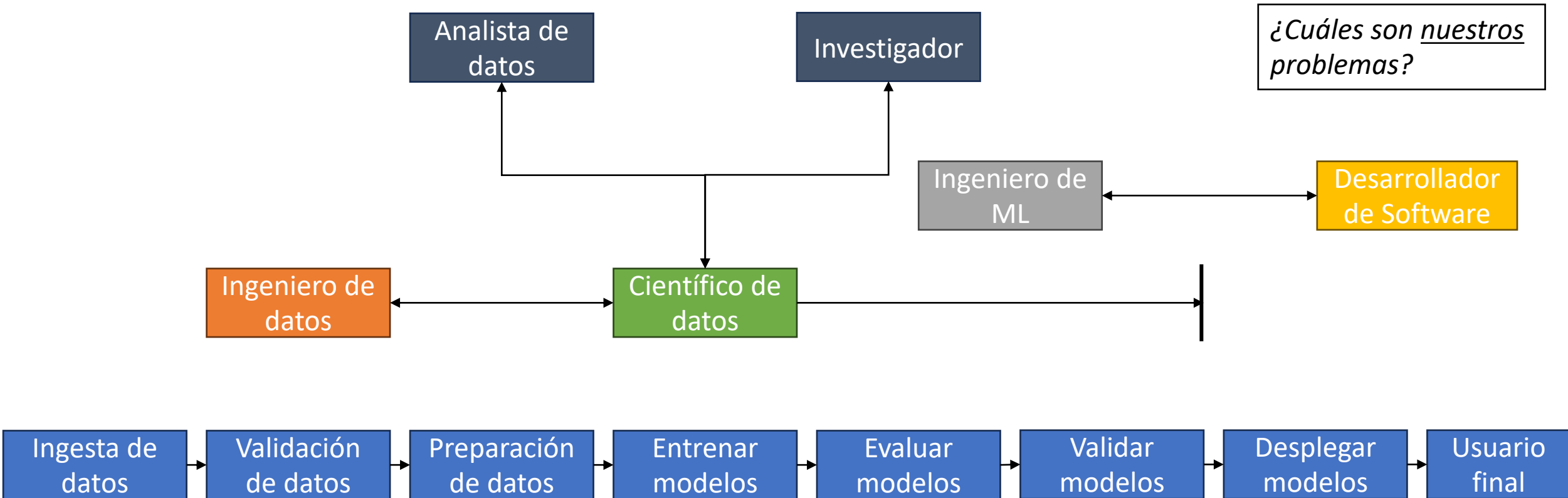


# Patrones de diseño y Machine Learning

## Data Science Lifecycle



# Patrones de diseño y Machine Learning



**Figura:** Roles de trabajo relacionados con datos y Machine Learning. Inspirado en [1].

# ¿Qué problemas existen en ML?

## Representación de datos

- Tipos de datos
- Escalamiento de datos
- Datos categóricos y numéricos
- Datos no estructurados

## Representación del problema

- ¿Regresión o clasificación?
- ¿Otras ideas?
- Elección de modelos de aprendizaje

## Entrenamiento de modelos

- Reproducibilidad
- Ajuste de parámetros e hiperparámetros
- Bias vs variance
- Tamaño de entrenamiento y conjuntos desequilibrados

# Patrones de representación de datos

# Primeras representaciones de datos

## Tipos de variables

### Variables categóricas

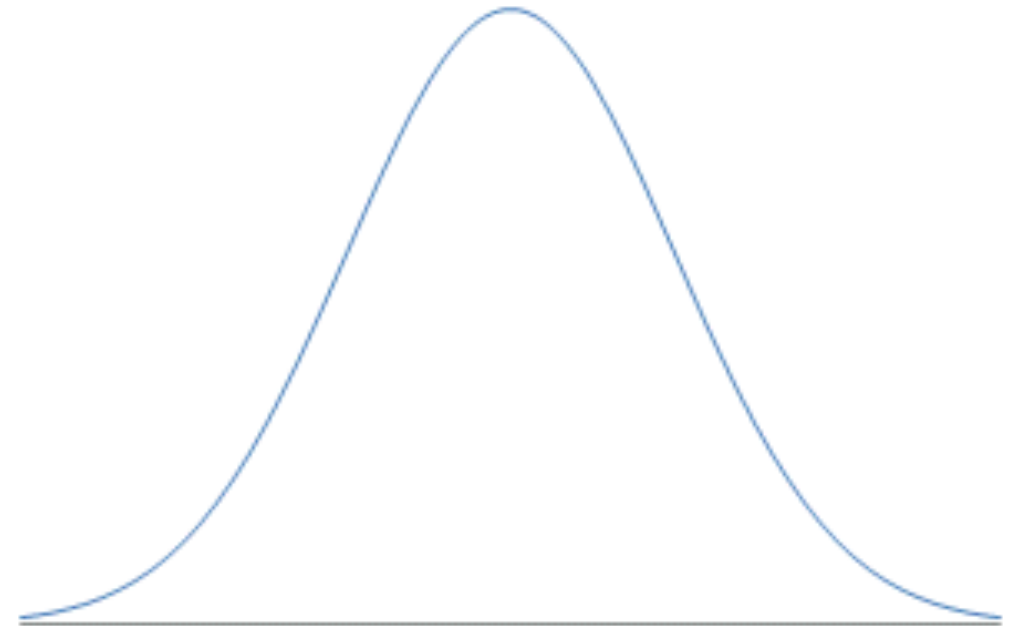
- Género de una persona
- Idioma
- Estado civil

### Variables numéricas

- Estatura (cm)
- Likes de una publicación
- Precio de una acción

# Primeras representaciones de datos: Variables Numéricas

- Si recuerdan, muchos modelos de aprendizaje que estudiamos **requieren que los datos numéricos sufran ciertas transformaciones** para poder funcionar adecuadamente.
- Algunos otros les es indiferente y pueden trabajar directamente.
- Requieren escalar los datos. ¿Recuerdan cuáles usamos?

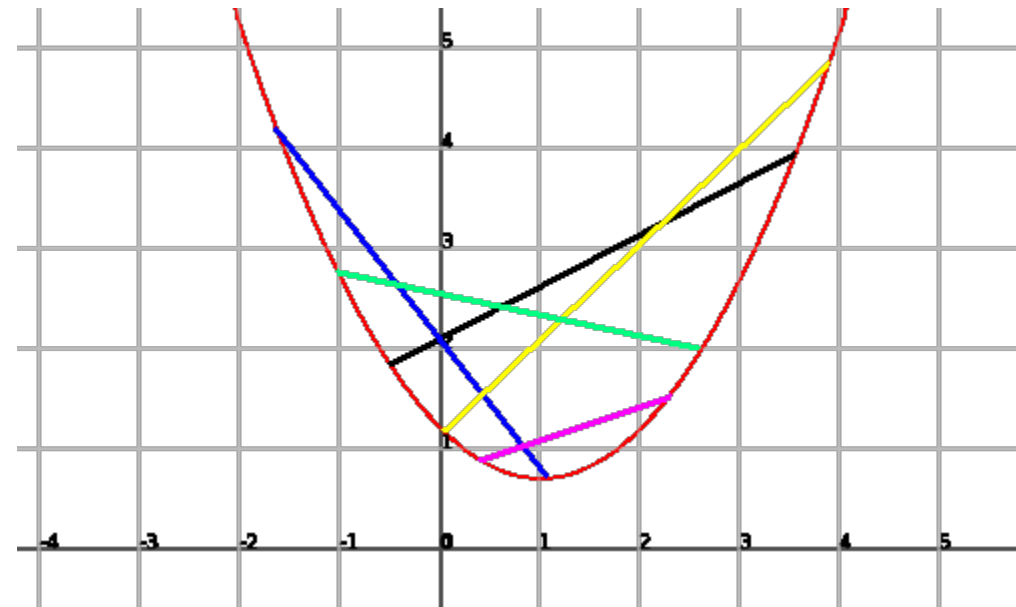


[Esta foto](#) de Autor desconocido está bajo licencia [CC BY-SA-NC](#)

# Primeras representaciones de datos: Variables Numéricas

Escalar datos es deseable porque:

- Optimizamos funciones (convexas).
  - Los algoritmos funcionan mejor con datos entre  $[-1,1]$  (o  $[0,1]$ ).
  - Convergen más rápido a una solución.
  - En este rango existe la mayor precisión de punto flotante.
  - Los problemas de optimización usan esos valores.

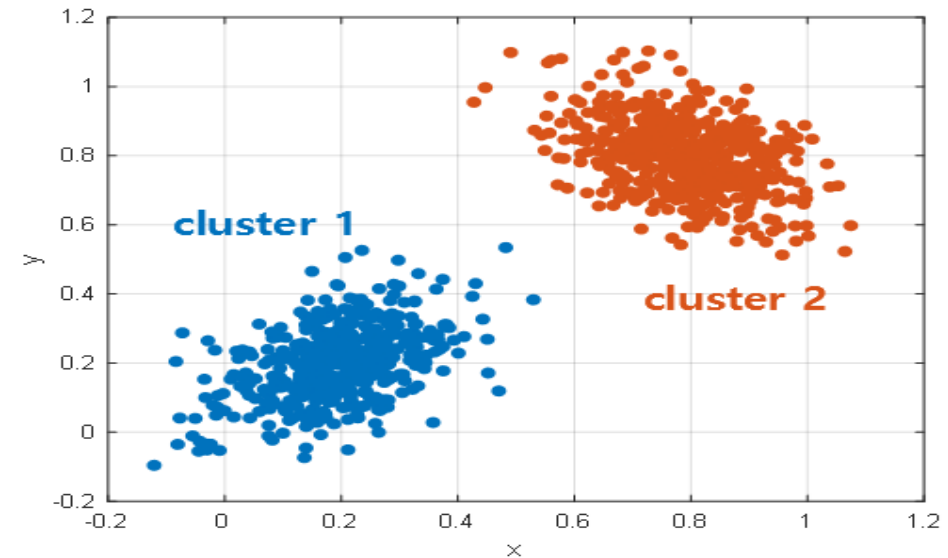


[Esta foto](#) de Autor desconocido está bajo licencia [CC BY-SA-NC](#)

# Primeras representaciones de datos: Variables Numéricas

Escalar datos es deseable porque:

- Algunos modelos son sensibles a las escalas (o magnitud) de los datos.
  - K-Medias con distancia euclidiana.
  - Regularizar con norma L1 o L2.



[Esta foto](#) de Autor desconocido está bajo licencia [CC BY-NC](#)

# Primeras representaciones de datos: Variables Numéricas

Algunas formas de escalas lineales:

## **Min-max scaling:**

El valor numérico  $x$  se escala linealmente de tal forma que el valor mínimo que puede tomar una variable es  $-1$  y el máximo  $1$ . Para obtener el nuevo valor  $x'$  escalado con valores arbitrarios en  $[a, b]$ , la fórmula es:

$$x' = a + \frac{(x - \min(X))(b - a)}{\max(X) - \min(X)}$$

Los valores mínimo y máximo se estiman del conjunto de entrenamiento  $X$ , y pueden ser outliers, lo que ocasiona que los datos reales se encuentren entre  $a$  y  $b$ .

# Primeras representaciones de datos: Variables Numéricas

Algunas formas de escalas lineales:

## **Clipping (con Min-max scaling):**

Evita el problema descrito anteriormente al seleccionar valores *razonables* en lugar del máximo y mínimo del conjunto de datos. Cualquier valor arriba o debajo de las cotas seleccionadas se vuelven -1 o 1, respectivamente

# Primeras representaciones de datos: Variables Numéricas

Algunas formas de escalas lineales:

## **Z-score normalization (standardization)**

Se encarga del problema de los outliers sin requerir conocimiento previo de los datos (e.g., saber cuál es el valor es máximo o mínimo) al escalar linealmente cada entrada usando la media y desviación estándar estimadas del conjunto de entrenamiento:

$$x' = \frac{x - \bar{x}}{\sigma}$$

# Primeras representaciones de datos: Variables Numéricas

Algunas formas de escalas lineales:

## Mean normalization

$$x' = \frac{x - \bar{x}}{\max(X) - \min(X)}$$

# Primeras representaciones de datos: Variables Numéricas

Algunas formas de escalas lineales:

## **Winsorizing**

Se usa la distribución empírica del conjunto de entrenamiento para cortar los datos según las cotas que se generan al considerar el 10° y 90° percentil de los valores de los datos.

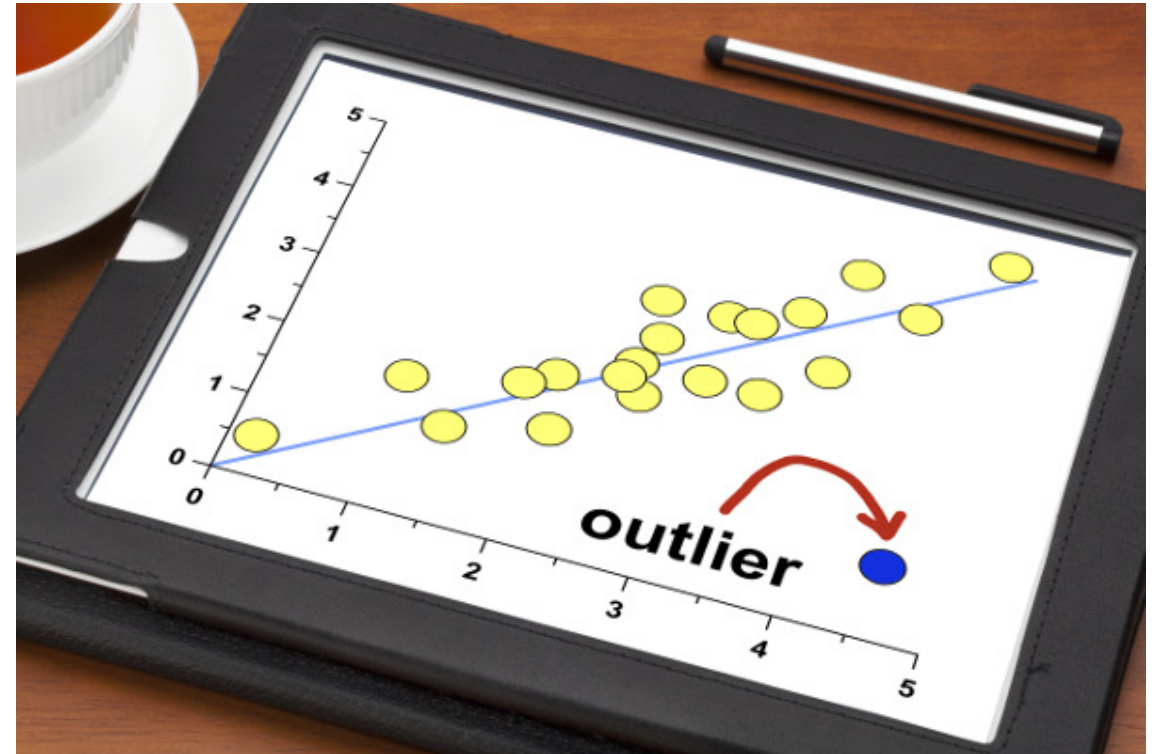
# Primeras representaciones de datos: Variables Numéricas

- Los métodos anteriores escalan los datos **linealmente**.
- Min-max scaling y clipping funcionan mejor con datos distribuidos de forma **uniforme**.
- Z-score normalization funciona mejor con datos con **distribución normal**.
- Aplicar estas transformaciones *no es buena práctica*, **es requerido**.
- Vamos a Google Colab para aplicar lo que hemos visto hasta el momento...

# Outliers



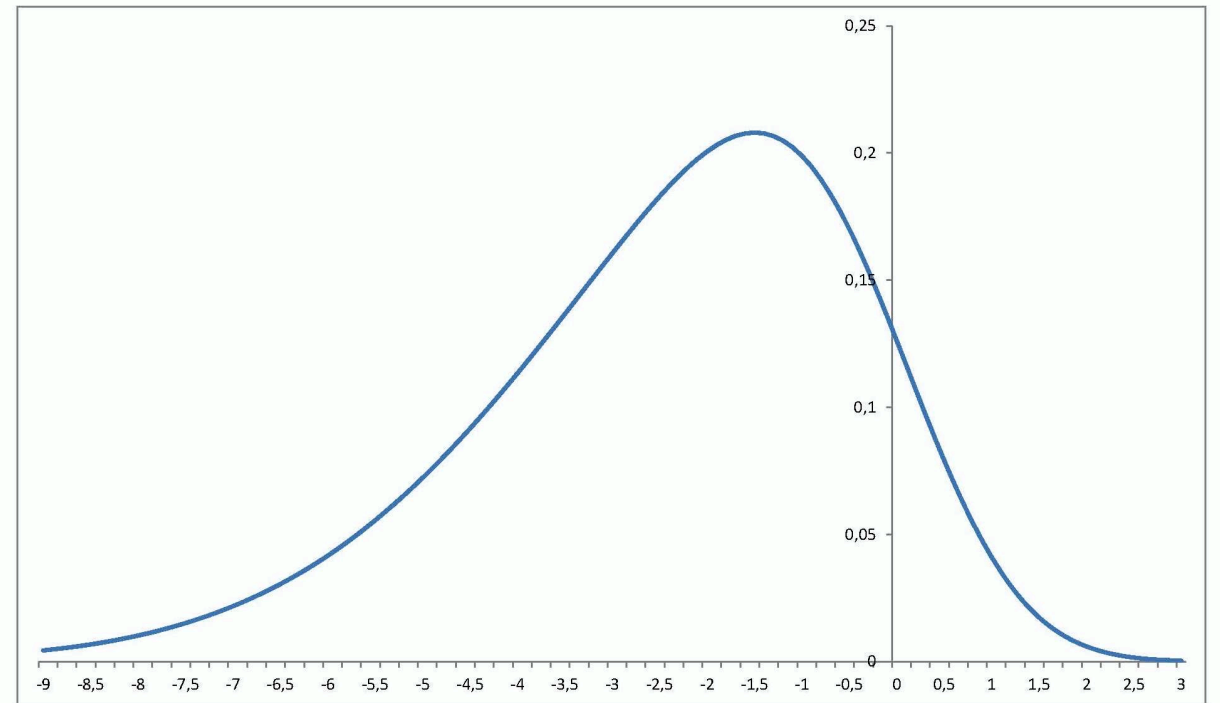
- Una postura común para tratar con estos datos es eliminarlos, ya que pueden afectar severamente el comportamiento de algunos modelos de aprendizaje.
- Sin embargo, deben tener cuidado con diferenciar entre eliminar datos inválidos y datos válidos.
  - E.g., eliminar un dato cuya madre tenga -1 año vs eliminar un dato cuya madre tenga 50 años.



[Esta foto](#) de Autor desconocido está bajo licencia [CC BY-NC](#)

# Transformaciones no lineales

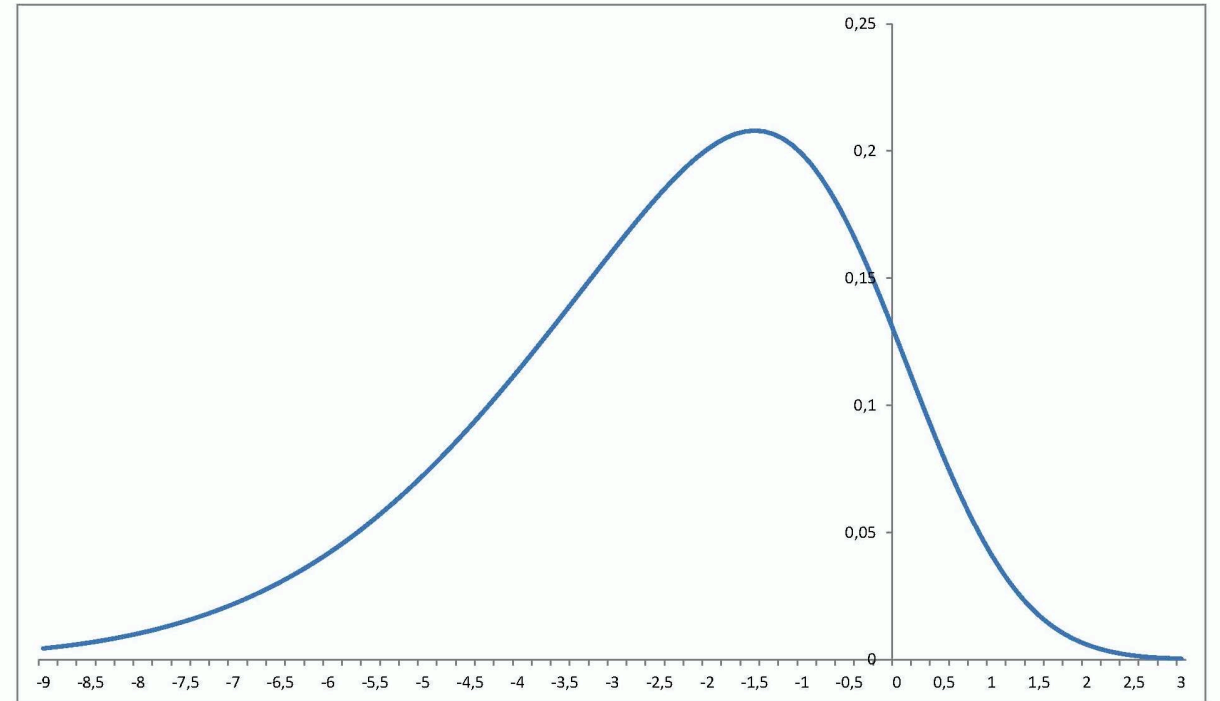
- ¿Qué pasa si los datos no siguen una distribución normal o uniforme?
- ¿Qué pasa si los datos se encuentran sesgados?



[Esta foto](#) de Autor desconocido está bajo licencia [CC BY-SA](#)

# Transformaciones no lineales

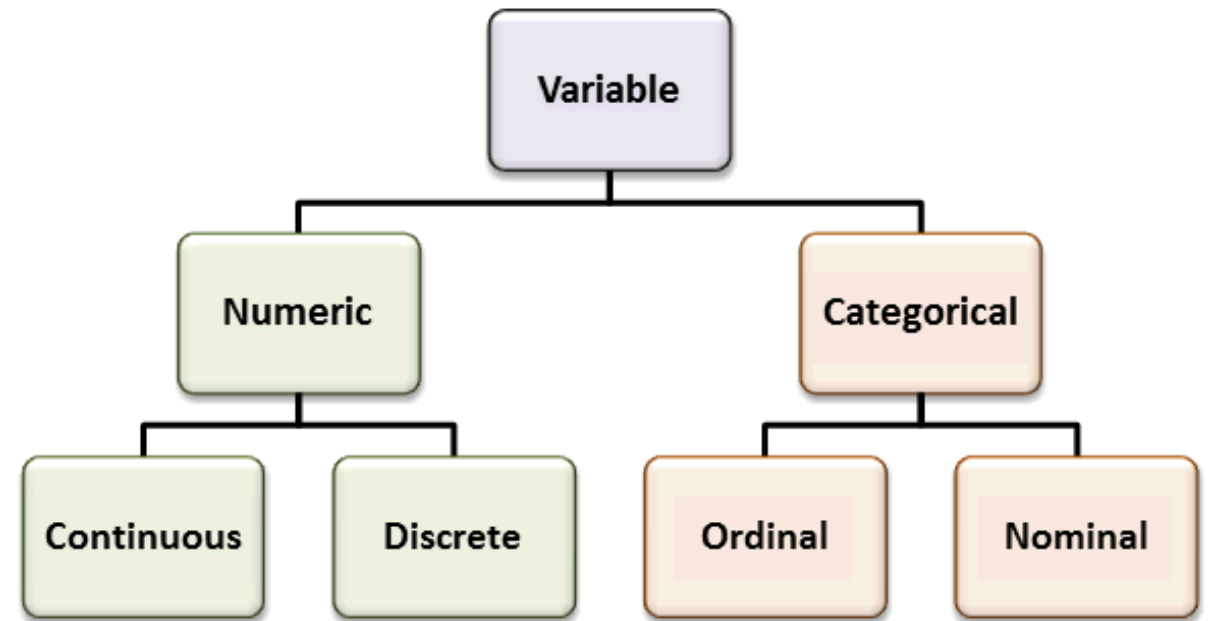
- Antes de aplicar transformaciones lineales para escalar los datos, puede ser útil aplicar transformaciones no lineales a los datos.
- E.g., el logaritmo, función sigmoide, expansiones polinomiales, transformada de Box-Cox.
- ¿Cuál elegir? Aquella que transforme los datos a una distribución normal o uniforme.



[Esta foto](#) de Autor desconocido está bajo licencia [CC BY-SA](#)

# Primeras representaciones de datos: Variables Categóricas

- Todos los modelos de ML funcionan con números.
- Debemos tratar las variables categóricas como números.
- ¿Cómo hacerlo?



[Esta foto](#) de Autor desconocido está bajo licencia [CC BY-SA](#)

# Primeras representaciones de datos: Variables Categóricas

## Primera idea:

Enumerar los posibles valores y mapearlos a una escala ordinal.

¿Bueno o malo?

Idioma	Valor numérico asignado
Inglés	1.0
Francés	2.0
Japonés	3.0

# Primeras representaciones de datos: Variables Categóricas

## Primera idea:

Enumerar los posibles valores y mapearlos a una escala ordinal.

**Malo:** El modelo intenta interpolar los valores asignados a Japonés e Inglés para determinar algo del Francés. No existe relación ordinal en este caso.

Idioma	Valor numérico asignado
Inglés	1.0
Francés	2.0
Japonés	3.0

# Primeras representaciones de datos: Variables Categóricas

## Segunda idea:

La forma más sencilla es utilizar **one-hot encoding**.

Ya lo vimos antes en Árboles de Decisión. ¿Cómo quedaría?

Idioma	Valor numérico asignado
Inglés	1.0
Francés	2.0
Japonés	3.0

# Primeras representaciones de datos: Variables Categóricas

## Segunda idea:

One-hot-encoding requiere **que sepamos el vocabulario o el total de posibles opciones** que puede adoptar una variable.

Idioma	Valor numérico asignado
Inglés	1.0
Francés	2.0
Japonés	3.0

# Primeras representaciones de datos: Variables Categóricas

A veces es útil tratar una variable numérica como categórica:

- Cuando la variable numérica es más bien, un índice.
  - E.g., cuando la variable representa un día de la semana en el problema de predecir tráfico vehicular.
  - Las entradas pueden ser 1,2,...,7; pero no es una variable continua, solo un índice, el cual puede ser arbitrario (¿empieza en lunes o domingo?)

Y también aplicar one-hot-encoding.



# Primeras representaciones de datos: Variables Categóricas

A veces es útil tratar una variable numérica como categórica:

- Cuando la relación entre los datos y la variable no es continua.
  - Los niveles de tráfico de un día (viernes) no influyen en otro días (domingo).



# Primeras representaciones de datos: Variables Categóricas

A veces es útil tratar una variable numérica como categórica:

- Cuando es buena idea agrupar (*bucket*) la variable numérica.
  - Puede resultar útil agrupar los días en dos conjuntos: entre semana y fines de semana. A esto se le conoce como *bucketing*, y la idea es agrupar de más a menos.



# Primeras representaciones de datos: Variables Categóricas

A veces es útil tratar una variable numérica como categórica:

- Cuando queremos tratar diferentes valores de la variable numérica como independientes del resultado de clasificación.
  - Si observamos que el tráfico cambia mucho según la zona, es buena idea mapear las zonas con una variable categórica. De esta forma, el modelo es obligado a ajustar sus parámetros para los distintos valores que puede adoptar la variable.



# Arreglos de Variables Categóricas

- Supongamos que ahora queremos incorporar a un modelo para determinar la salud de una madre el número de partos que ha tenido.
- Podemos agruparlos en un arreglo:  
[natural, cesárea, inducido]



# Arreglos de Variables Categóricas

- De tal manera que si una paciente tiene la siguiente información:

[natural, inducido]

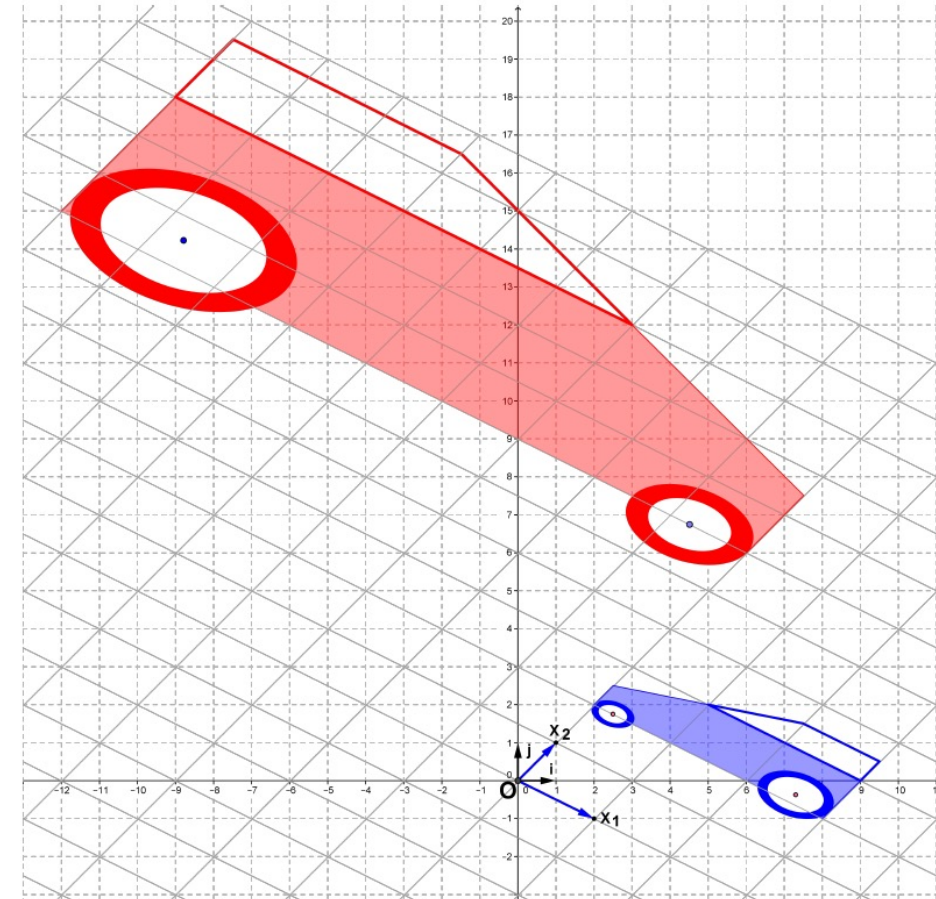
- La información se puede representar como:

[1,0,1]



# Patrón 0: Transformar las variables del modelo

- Lo ponemos como patrón 0 ya que no es una buena práctica, **es una necesidad** para que los modelos funcionen adecuadamente.
- **Transformar variables numéricas** con transformaciones lineales o no lineales.
- Determinar si una variable categórica o numérica es **bien representada**.
- Aplicar **one-hot encoding** para variables categóricas.



[Esta foto](#) de Autor desconocido está bajo licencia [CC BY-SA](#)

# Patrones de Representación de Datos

¿Qué problemas encuentran en el one-hot encoding?

Idioma	Valor numérico asignado
Inglés	1.0
Francés	2.0
Japonés	3.0
⋮	⋮

# Patrones de Representación de Datos

¿Qué problemas encuentran en el one-hot encoding?

- El **vocabulario** se encuentra incompleto.
  - Este se determina de nuestros datos de entrenamiento. ¿Qué pasa si no cuenta con todos los hospitales o médicos de la zona?

Idioma	Valor numérico asignado
Inglés	1.0
Francés	2.0
Japonés	3.0
⋮	⋮

# Patrones de Representación de Datos

¿Qué problemas encuentran en el one-hot encoding?

- Las variables categóricas tienen **cardinalidad alta**.
  - Tenemos siete días en una semana, un valor decente. ¿Qué sucede si una variable tiene miles o millones de categorías?

Idioma	Valor numérico asignado
Inglés	1.0
Francés	2.0
Japonés	3.0
⋮	⋮

# Patrones de Representación de Datos

¿Qué problemas encuentran en el one-hot encoding?

- ***Cold-start.***

- Entrenamos un modelo y se pone en producción. ¿Qué pasa si surgen nuevos hospitales o llegan nuevos médicos a la zona?

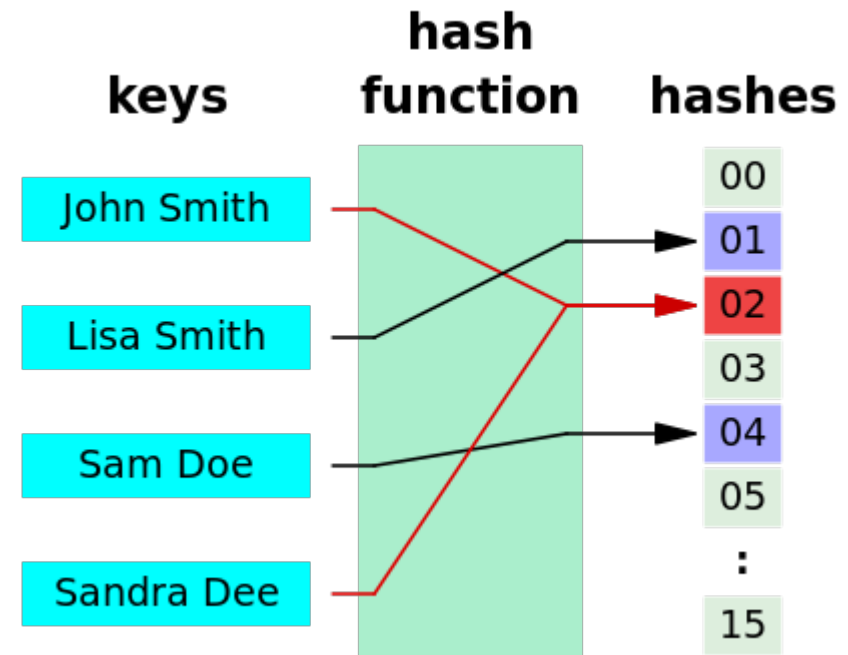
Idioma	Valor numérico asignado
Inglés	1.0
Francés	2.0
Japonés	3.0
⋮	⋮

# Patrón 1: Hashing Features

¿Qué es un hash/hashing?

## Solución: Hashing Features

1. Se convierte la variable categórica en turno en una cadena de texto única.
2. Se utiliza un algoritmo de *hashing* con dicha cadena de texto, este devuelve un número.
3. Se toma el resto del hash dividido por el número de grupos (*buckets*) deseados.

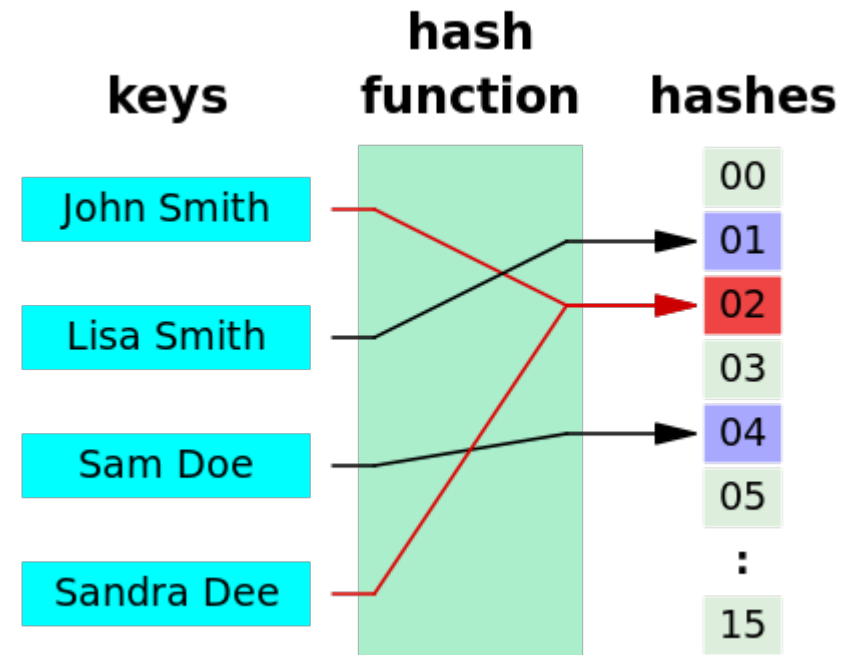


[Esta foto](#) de Autor desconocido está bajo licencia [CC BY-SA](#)

# Patrón 1: Hashing Features

¿Qué es un hash/hasing?

Hospital	Hospital (transf.)	Hash3	Hash10	Hash1000
Hospital 1	HZN	1	3	543
Hospital 2	ABA	2	9	699
Hospital 3	DEF	2	7	72
Hospital 4	HJM	0	6	25
Hospital 5	HAP	1	2	100



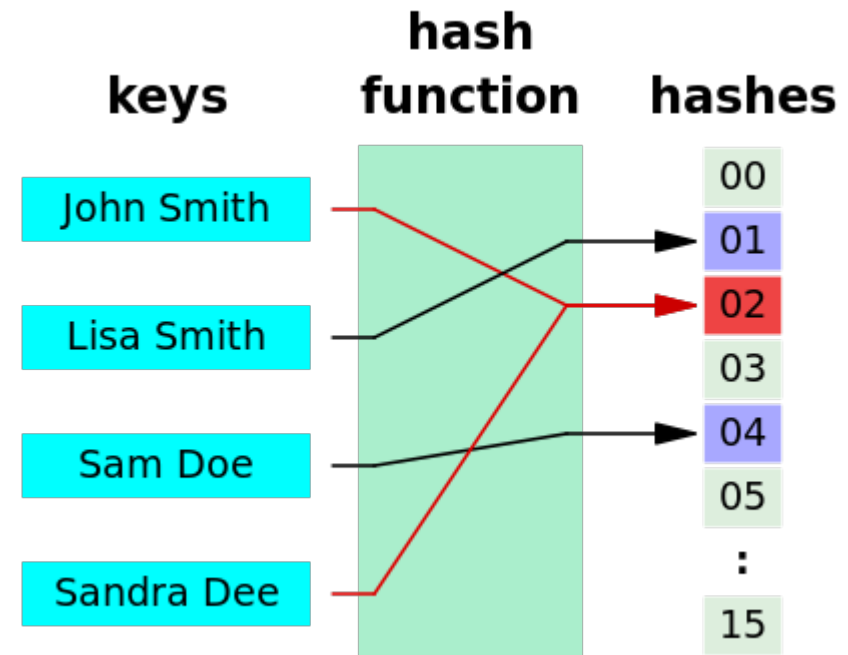
Esta foto de Autor desconocido está bajo licencia [CC BY-SA](#)

# Patrón 1: Hashing Features

¿Qué es un hash/hashting?

## Desventajas

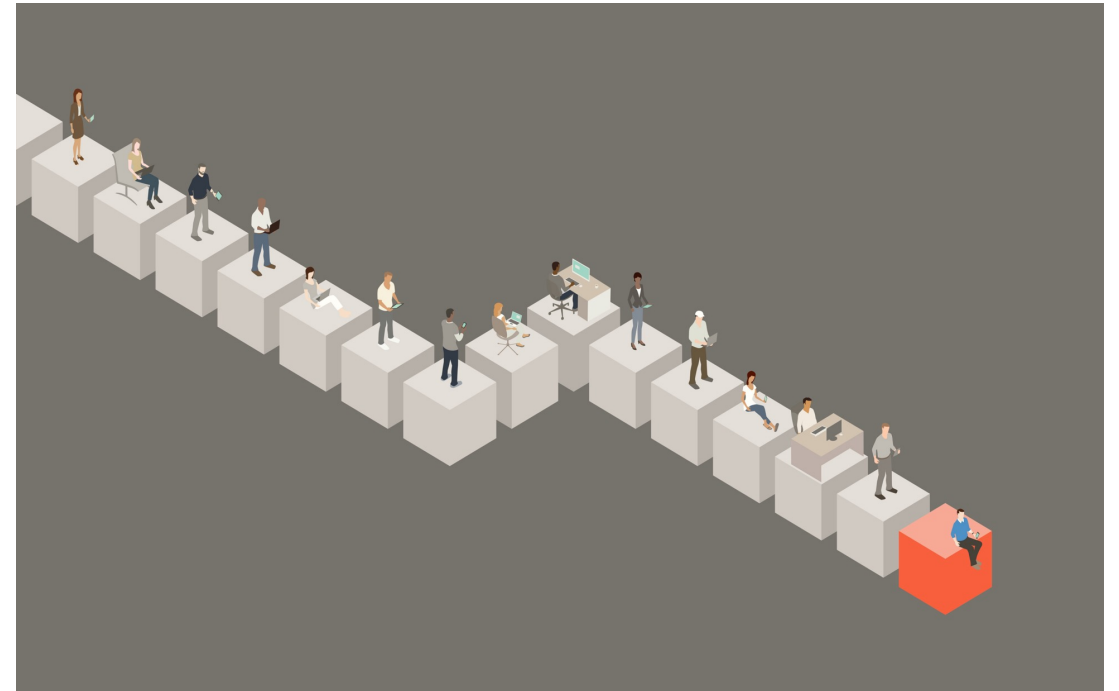
1. Colisión de cubetas.
2. Los datos se pueden distribuir con sesgo.
3. Elección del mejor número de cubetas (hiperparámetro).
4. ¿Qué algoritmo de hashing usar?
5. Posibilidad de cubetas vacías.



[Esta foto](#) de Autor desconocido está bajo licencia [CC BY-SA](#)

# Patrones de Representación de Datos

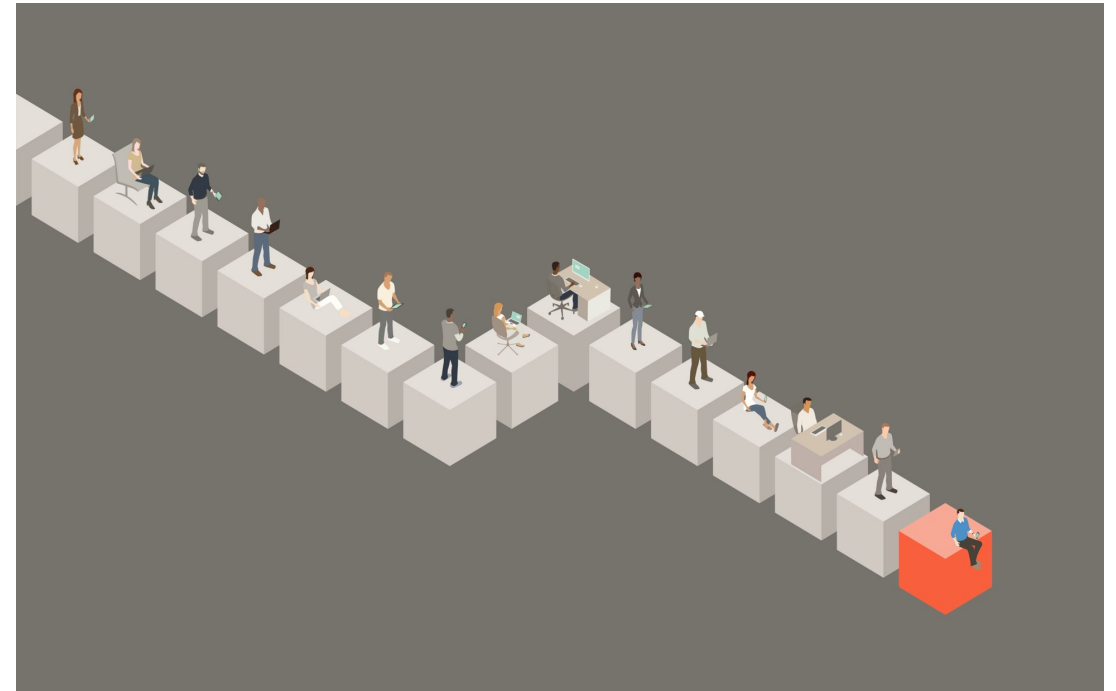
- Sabemos que los modelos de ML **buscan patrones en los datos** que capturan propiedades que se relacionan con las salidas.
- Por lo tanto, **cómo se representan las características** que modelan los datos **afecta la calidad final del modelo.**



# Patrones de Representación de Datos

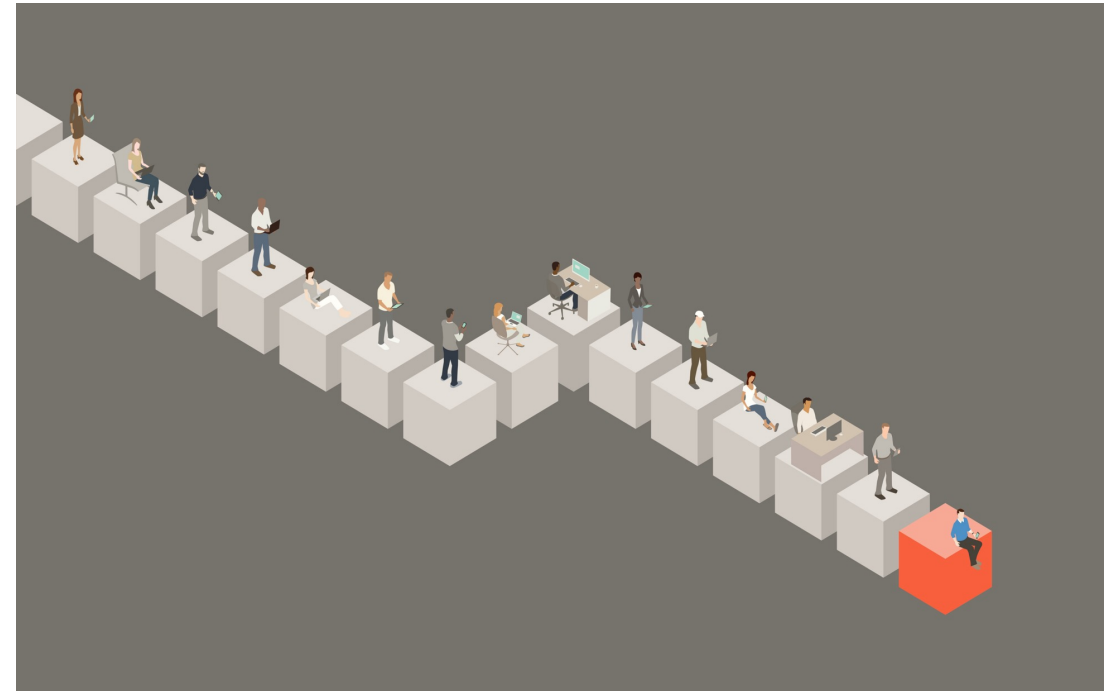
Existen dos tipos de datos que se manejan comúnmente en problemas de ML:

- **Datos estructurados:** números, o cualquier información que se pueda organizar como un Excel.
- **Datos no estructurados:** texto, imágenes, señales, videos, series.



# Patrones de Representación de Datos

- Maneja datos estructurados es muy directo, ya que mayormente son números y **no necesitan más representación.**
- Cuando manejamos, por ejemplo, texto, **¿cómo lo representamos (numéricamente) para que la computadora y el modelo de ML lo entienda?**

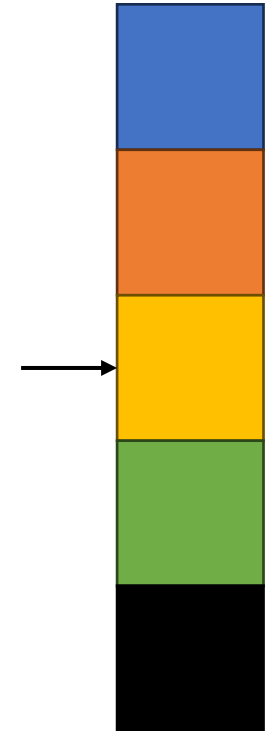


# Patrón 2: Embeddings

## Solución: Embeddings

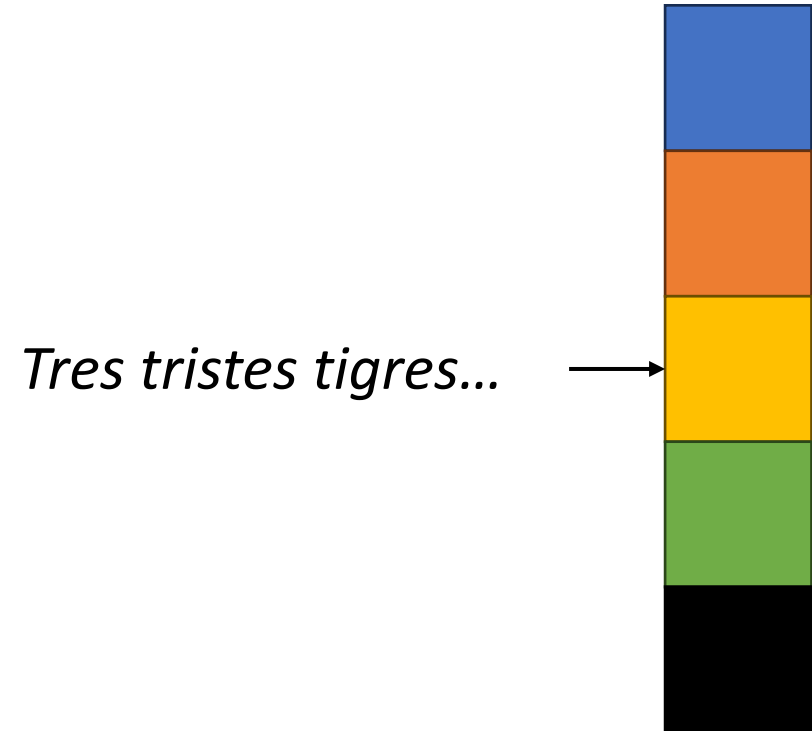
- Los embeddings son formas de representar información no estructurada numéricamente.
- Esto se hace mediante un mapeo de los datos hacia un vector, generalmente denso.
- Para variables categóricas, un embedding es el **one-hot encoding**.

*Tres tristes tigres...*

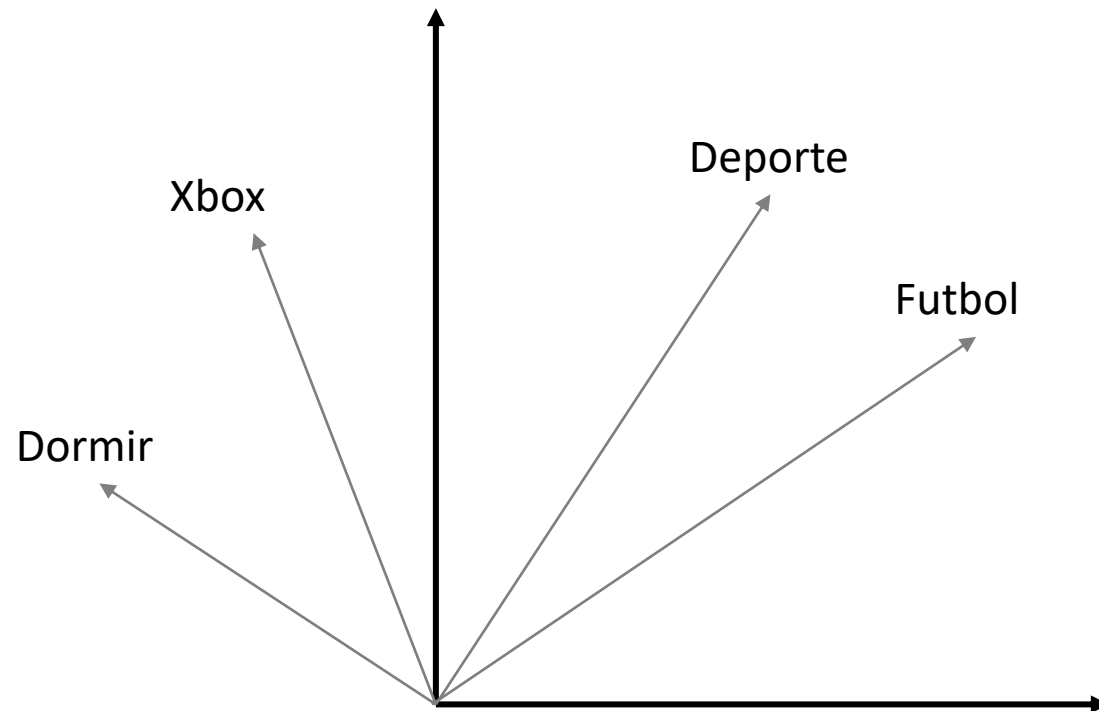


# Patrón 2: Embeddings

- La idea del embedding es que el mapeo captura las relaciones de cercanía de los datos en una representación de menor dimensión.
- Por ejemplo, esperamos que la palabra “*deporte*” y “*futbol*” sean cercanas.
  - ¿Cómo es que dos vectores se encuentran *cerca* entre ellos?

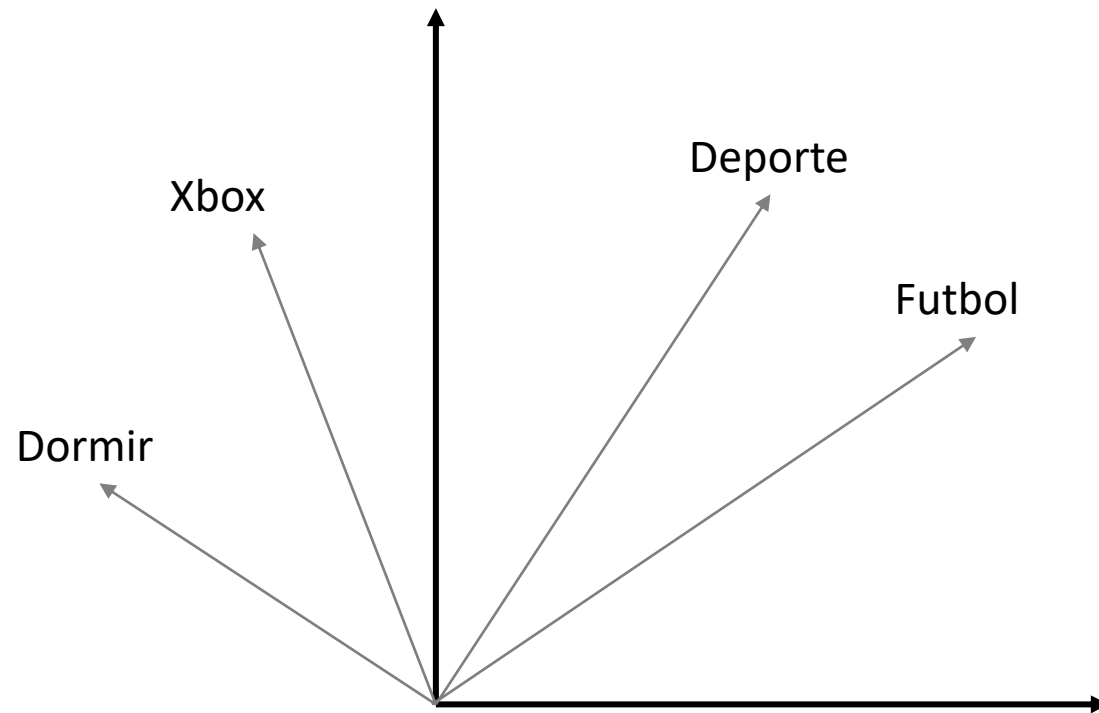


# Patrón 2: Embeddings



- Un **embedding se debe aprender**, i.e., no es una representación arbitraria.
- En la mayoría de los casos, **es una capa oculta** de una red neuronal.
  - Los pesos de esa capa se entrenan con la entrada deseada, y los pesos de esa capa se aprenden para generar la nueva representación.
- Al aprenderse, el embedding permite extraer **similitudes inherentes** entre categorías representadas con valores numéricos, los cuales nos permiten compararlos.

# Patrón 2: Embeddings

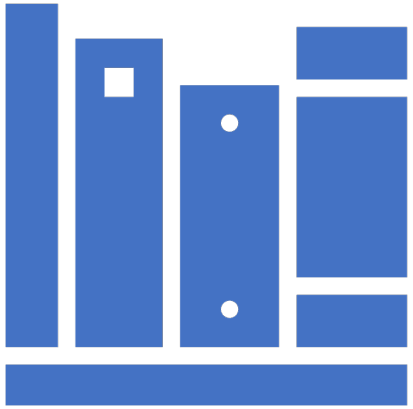


## Problemas:

- Dimensión del nuevo vector de representación.
- ¿Cómo hacerlo con los diferentes tipos de información no estructurada?
- Cantidad de datos para aprender “*buenos*” embeddings.
- Existen embeddings pre-hechos. ¿Vale la pena crear los nuestros?

# Patrón 2: Embeddings

Vamos a explorar algunas formas para representar texto...



# Fin de la presentación

¡Gracias por su atención!