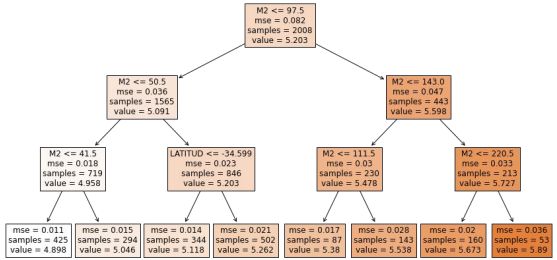
Aprendizaje no supervisado
Reducción de la dimensionalidad y PCA

Laboratorio de Datos - 1°C 2021

¿Qué veníamos haciendo hasta ahora?

Buscábamos asociar datos representados en el espacio features con una variable que tomábamos como target mediante un modelo.



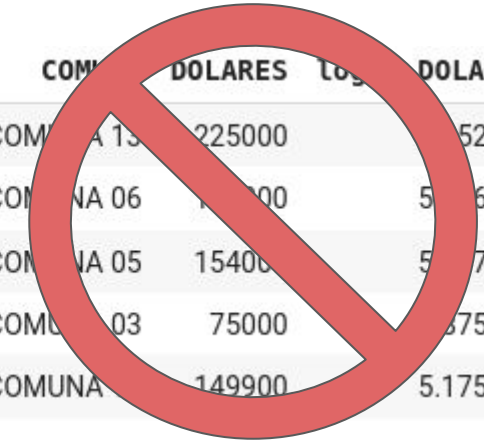
	M2	AMBIENTES	ANTIGUEDAD	BAÑOS	LATITUD	LONGITUD	COMUNA	DOLARES	log10_DOLARES
0	81	3	4	1	-34.581078	-58.449433	COMUNA 13	225000	5.352183
1	69	3	20	1	-34.623129	-58.439338	COMUNA 06	140000	5.146128
2	75	3	20	1	-34.604972	-58.421278	COMUNA 05	154000	5.187521
3	42	2	40	1	-34.604725	-58.399524	COMUNA 03	75000	4.875061
4	90	3	1	1	-34.623390	-58.504401	COMUNA 10	149900	5.175802

Los parámetros del modelo se aprendían buscando mejorar la predicción sobre la variable target: **aprendizaje supervisado**.

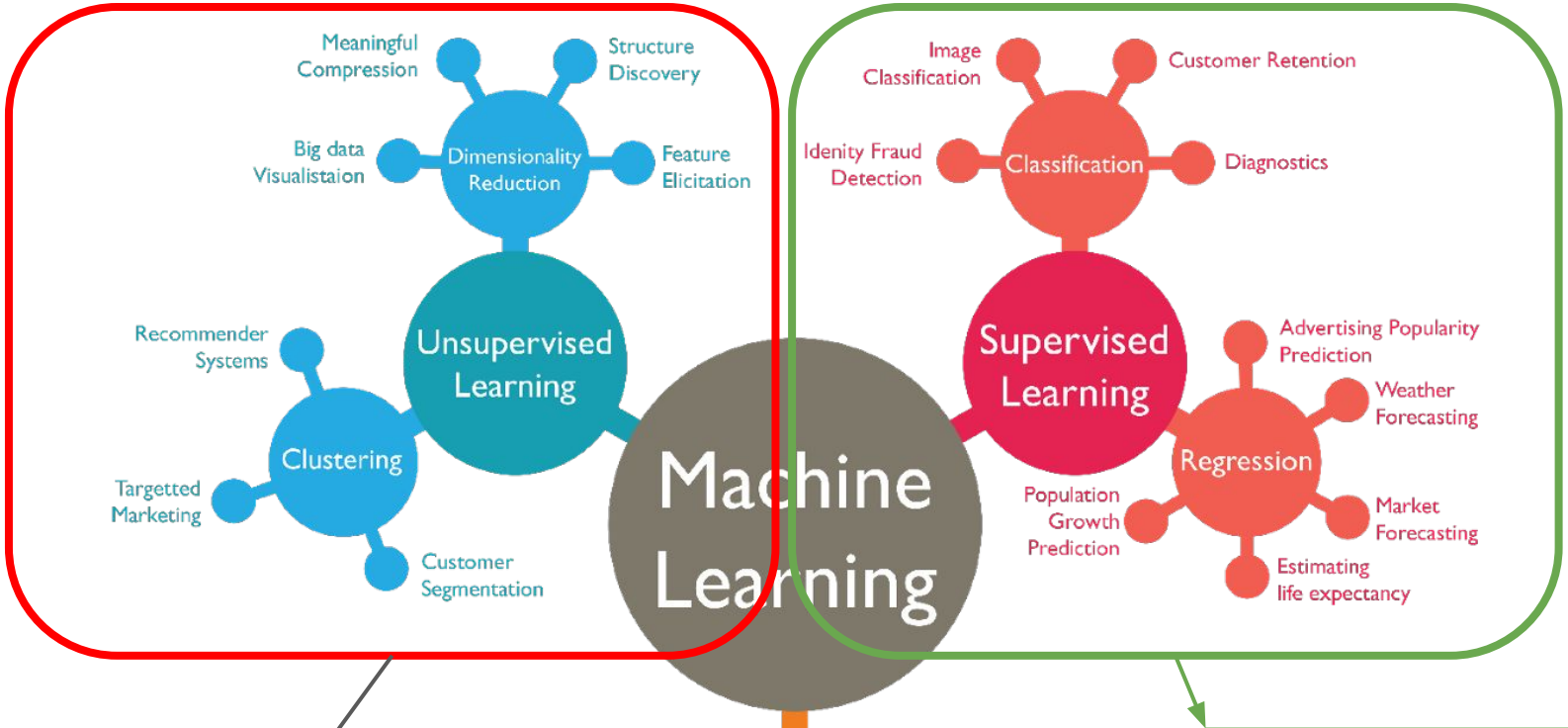
¿Qué otra cosa vamos a hacer a partir de hoy?

Vamos a olvidarnos por un momento de querer predecir una variable y enfocarnos en **cómo están representados los datos en el espacio de features**.

	M2	AMBIENTES	ANTIGUEDAD	BAÑOS	LATITUD	LONGITUD	COM	DOLARES	log	DOLARES
0	81	3	4	1	-34.581078	-58.449433	COM	A 15	225000	52183
1	69	3	20	1	-34.623129	-58.439338	CO	NA 06	00	5 6128
2	75	3	20	1	-34.604972	-58.421278	CO	A 05	15400	5 7521
3	42	2	40	1	-34.604725	-58.399524	COMU	03	75000	375061
4	90	3	1	1	-34.623390	-58.504401	COMUNA		149900	5.175802



Como no va a haber una variable target que nos diga si estamos aprendiendo bien o mal este tipo de técnicas se llaman de **aprendizaje no supervisado**.



Hoy y la próxima clase

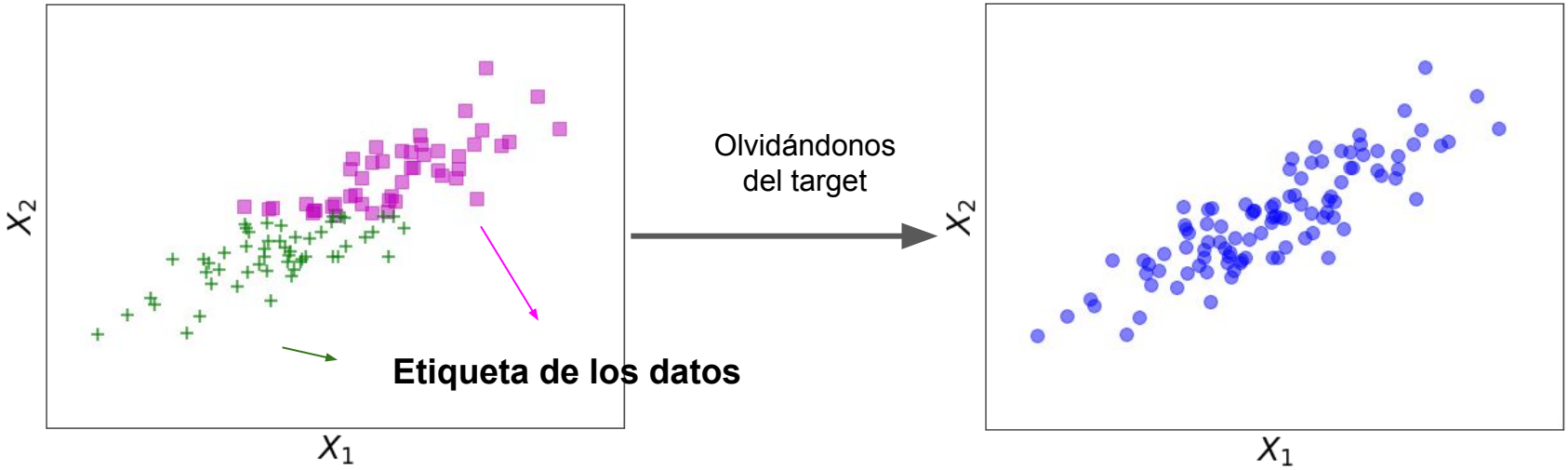
Ya visto en la materia

¿Quieren hablar del tenebroso contenido aquí abajo, o prefieren que siga con la clase?!!! :) :) :)

https://www.7wdata.be/visualization/types-of-machine-learning-algorithms-2/

¿Qué es el aprendizaje no supervisado?

El estudio o la exploración de cómo están representados datos y qué conclusiones podemos sacar de dicha representación.



A**prendizaje no supervisado** es todo lo que podemos hacer con solo la representación de los datos en el espacio de features.

¿Qué podemos hacer con solo el conocimiento de la representación de los datos en el espacio de features?

Exploración de los datos y extracción de conocimiento! Por ejemplo, aplicando los siguientes conceptos:

- **Clustering (próxima clase)**: agrupar instancias que tengan una descripción similar en el espacio de features (próxima clase). Básicamente respondemos si hay subconjuntos de datos muy parecidos entre si.
- **Reducción de la dimensionalidad (clase de hoy)**: encontrar combinaciones de features que reemplacen a los originales para reducir la dimensión del problema.

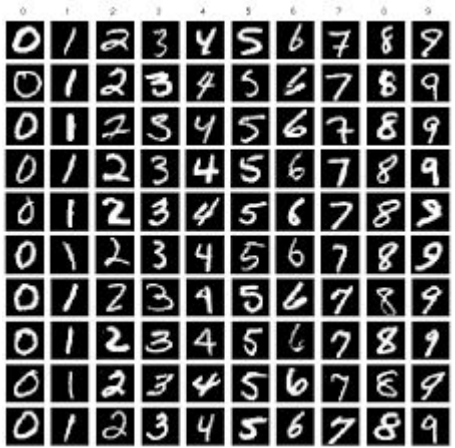
¿Por qué reducir la dimensionalidad del problema?

Pensar en datos descritos en un espacio de muchas dimensiones (muchos features): ¿todos los features aportan información relevante? ¿Hay redundancia en la información que aportan? ¿Es necesario trabajar con todos?

Reduciendo la dimensión podemos:

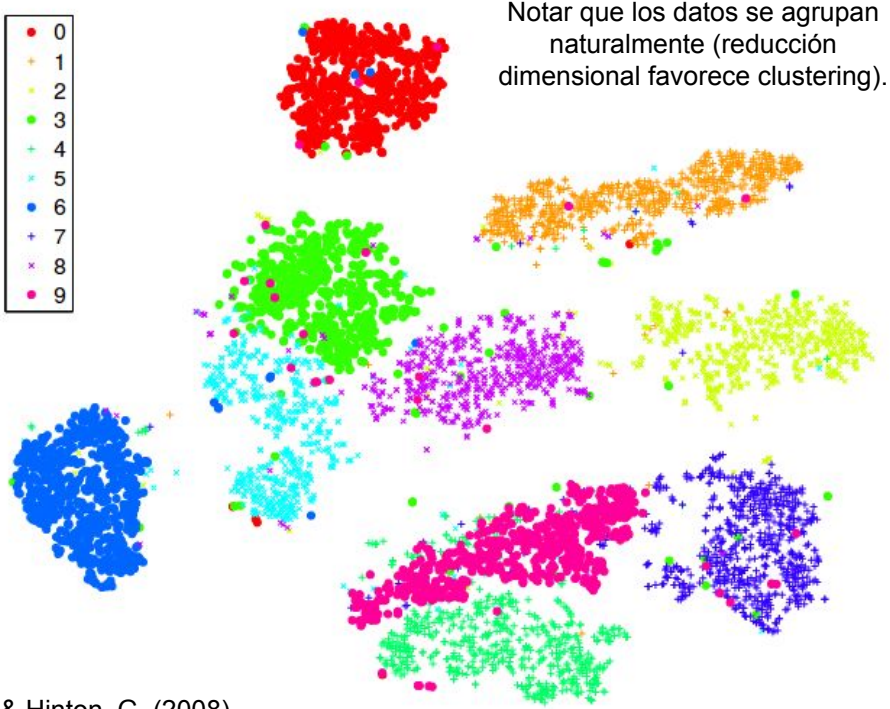
- **Visualizar** los datos en el espacio de dimensión reducida, más fácil de interpretar (ojalá siempre pudiéramos llevar todo a 2D).
- **Comprimir** la información: nos permite separar la señal del ruido (pérdida de información = abstracción).
- Tener un **punto de partida para clustering**: instancias parecidas en un espacio multidimensional son más parecidas en un espacio reducido (emergencia de estructuras).

¿Por qué reducir la dimensionalidad del problema?



MNIST dataset
Datos representados en un espacio de **748 píxeles**.

Visualización de datos multidimensionales en 2D



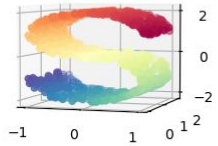
Notar que los datos se agrupan naturalmente (reducción dimensional favorece clustering).

(a) Visualization by t-SNE.

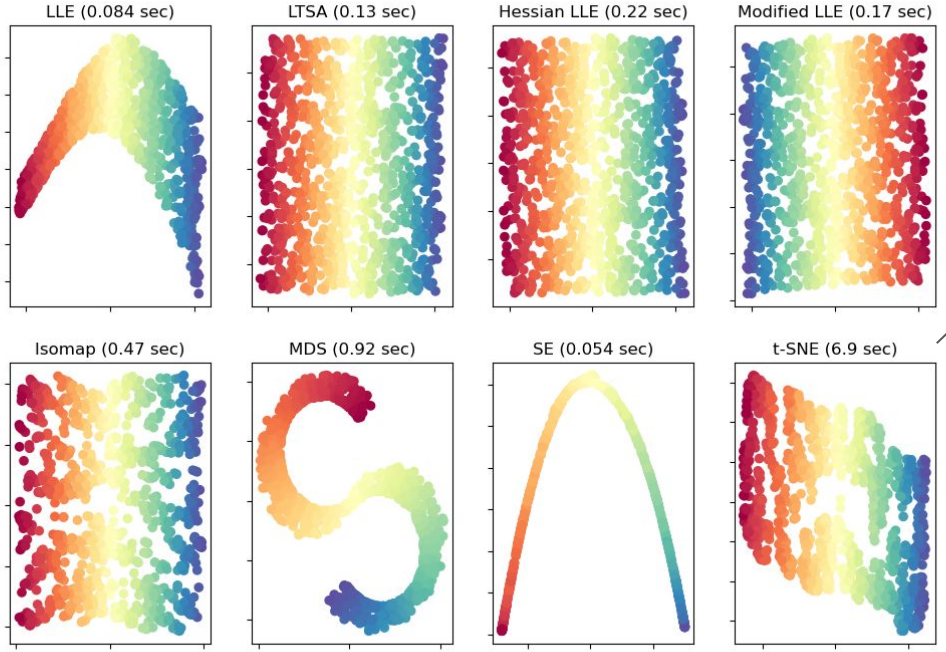
Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, *9*(11).

¿Por qué reducir la dimensionalidad del problema?

Figura original en 3D



Manifold Learning with 1000 points, 10 neighbors



Distintos algoritmos

Visualización de datos multidimensionales en 2D

https://scikit-learn.org/stable/auto_examples/manifold/plot_compare_methods.html#sphx-glr-auto-examples-manifold-plot-compare-methods-py

¿Por qué reducir la dimensionalidad del problema?

Compresión (con pérdida) de la información: separación de la señal del ruido.

Con 100 dimensiones (<< 4096) ya logramos una reproducción fiel de la imagen original.

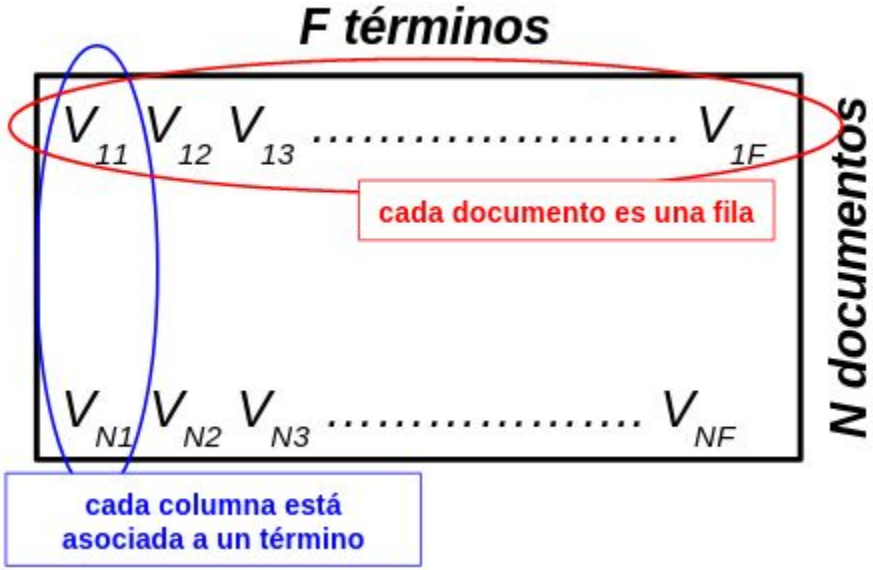
Datos representados en un espacio de **4096 píxeles**.

\+ dimensiones del espacio reducido

Imagen reconstruida de un espacio de dimensión reducida

Problema desarrollado en el colab...

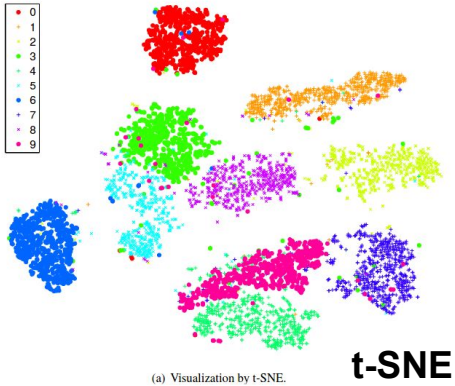
¿Por qué reducir la dimensionalidad del problema?



La reducción dimensional de una **matriz de documentos en el espacio de términos** lleva a agrupar **documentos en tópicos** y la **dimensiones reducidas son efectivamente los tópicos** (reducción dimensional favorece el clustering).

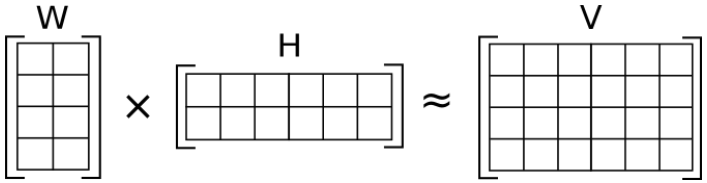
Más sobre detección de tópicos en futuras clases.

Algoritmos de reducción dimensional (solo para mencionar algunos…)

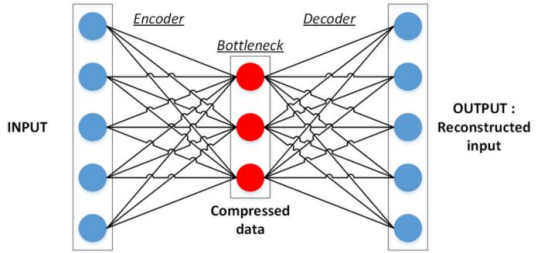


(a) Visualization by t-SNE.

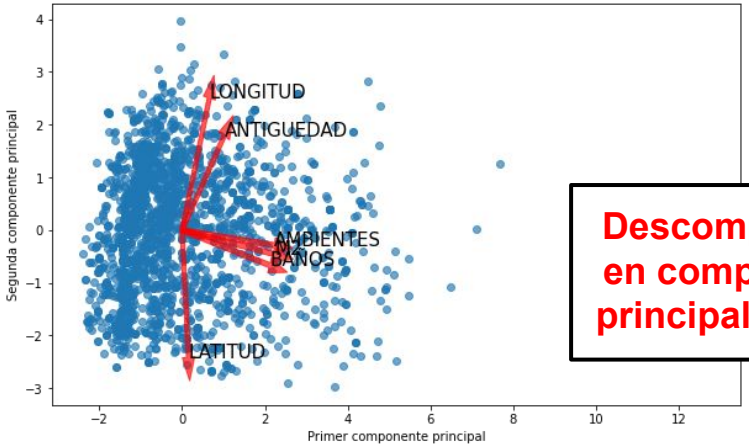
t-SNE



Non-negative matrix factorization (NMF)



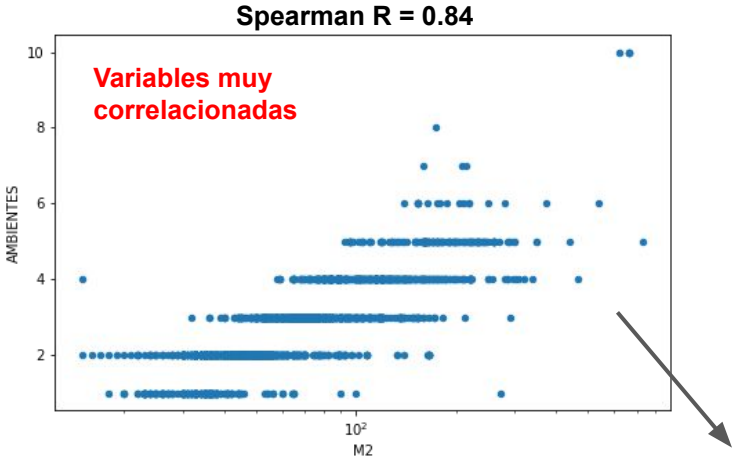
Autoencoders

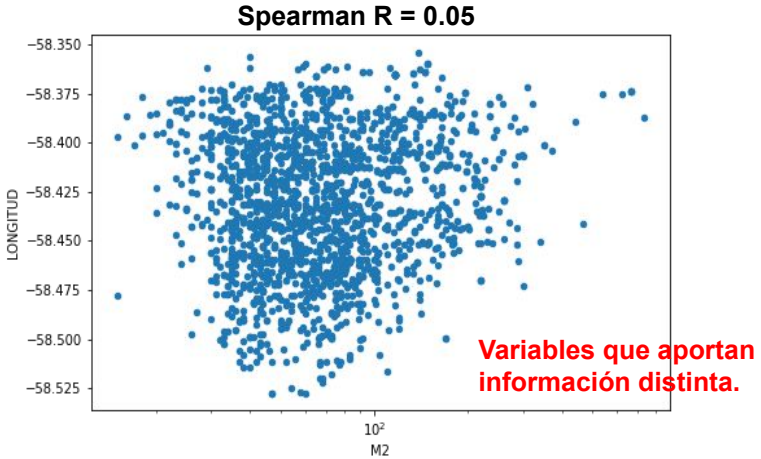


Descomposición en componentes principales (PCA)

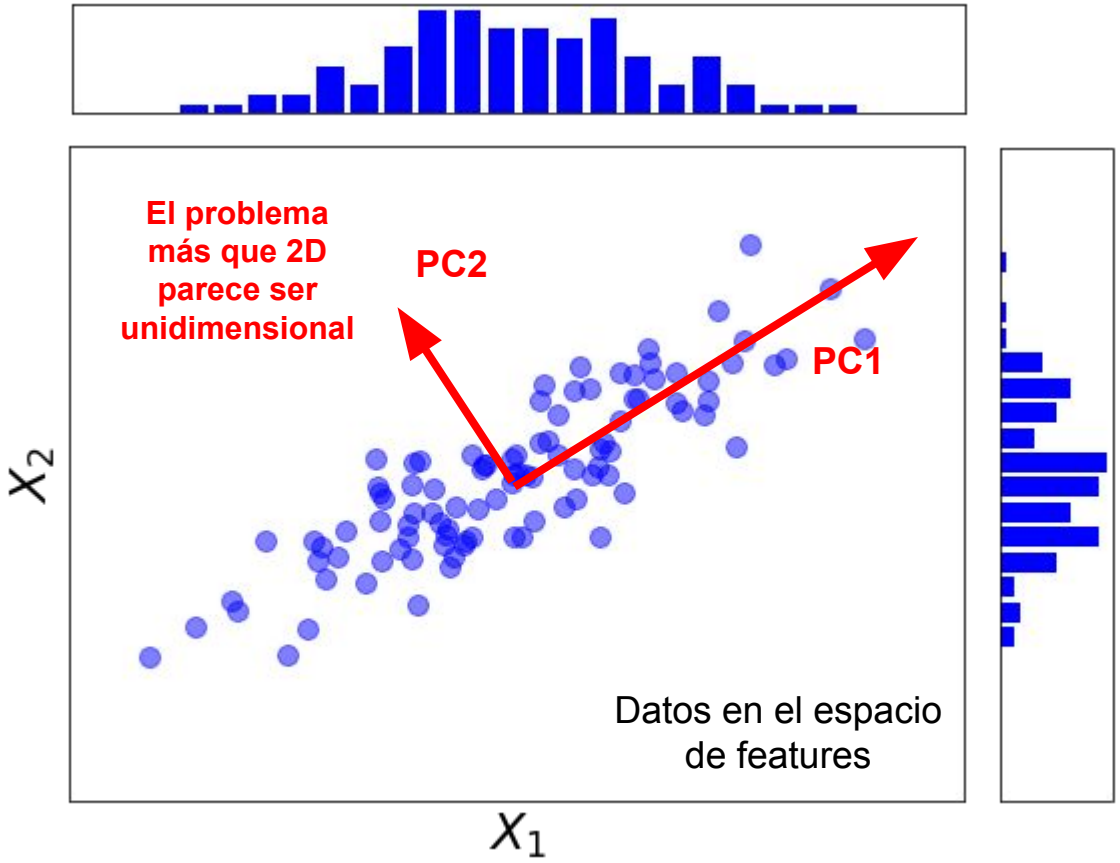
Descomposición en componentes principales (PCA)

Problema: muchas veces hay variables que contienen prácticamente la misma información que otras (están muy correlacionadas entre sí), **por lo que agregan una dimensión más al problema sin aportar muchas más información.**





Una opción para solucionar el problema sería tirar una de las dimensiones por redundante, sin embargo ¿estamos seguros que toda la información que tiramos no es necesaria? ¿Hay alguna otra forma más inteligente de tirar dimensiones?



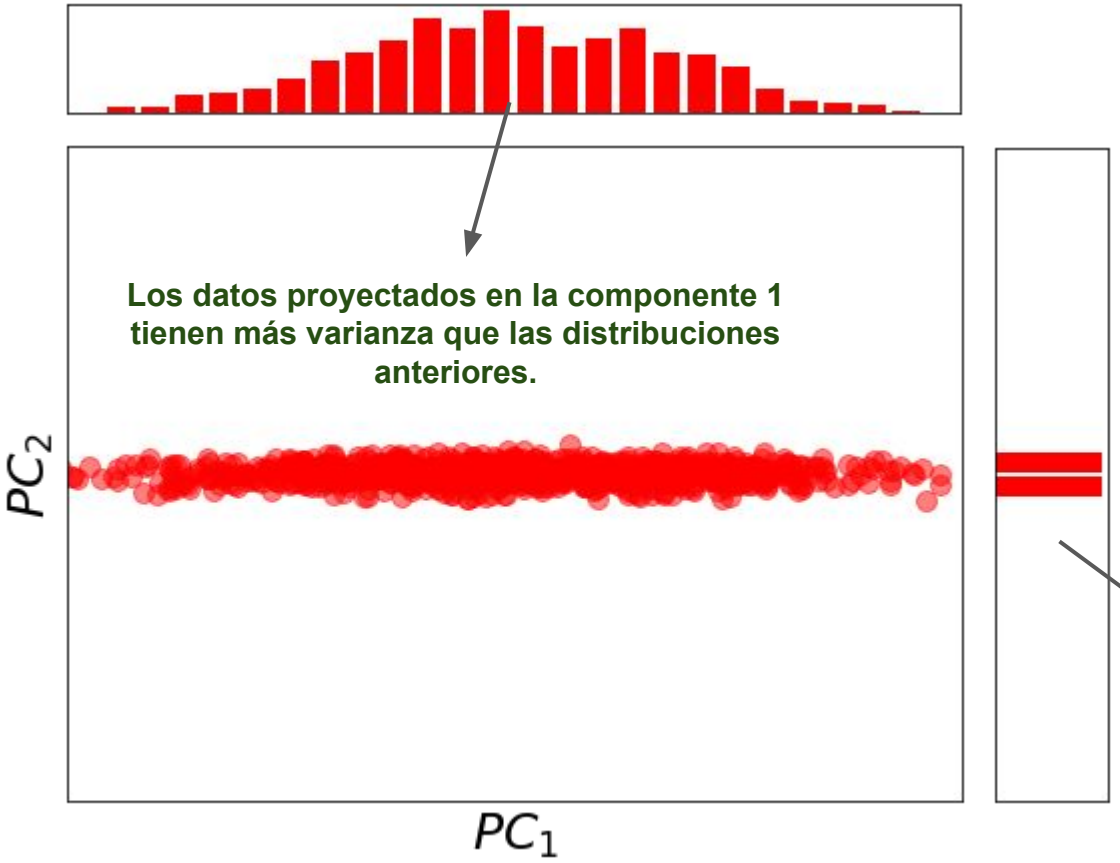
Esquema general

Podemos ver cómo se distribuyen los datos proyectando sobre cada uno de los features.

De las distribuciones podemos calcular su varianza, como una medida de qué tan dispersos están los datos en esa dirección.

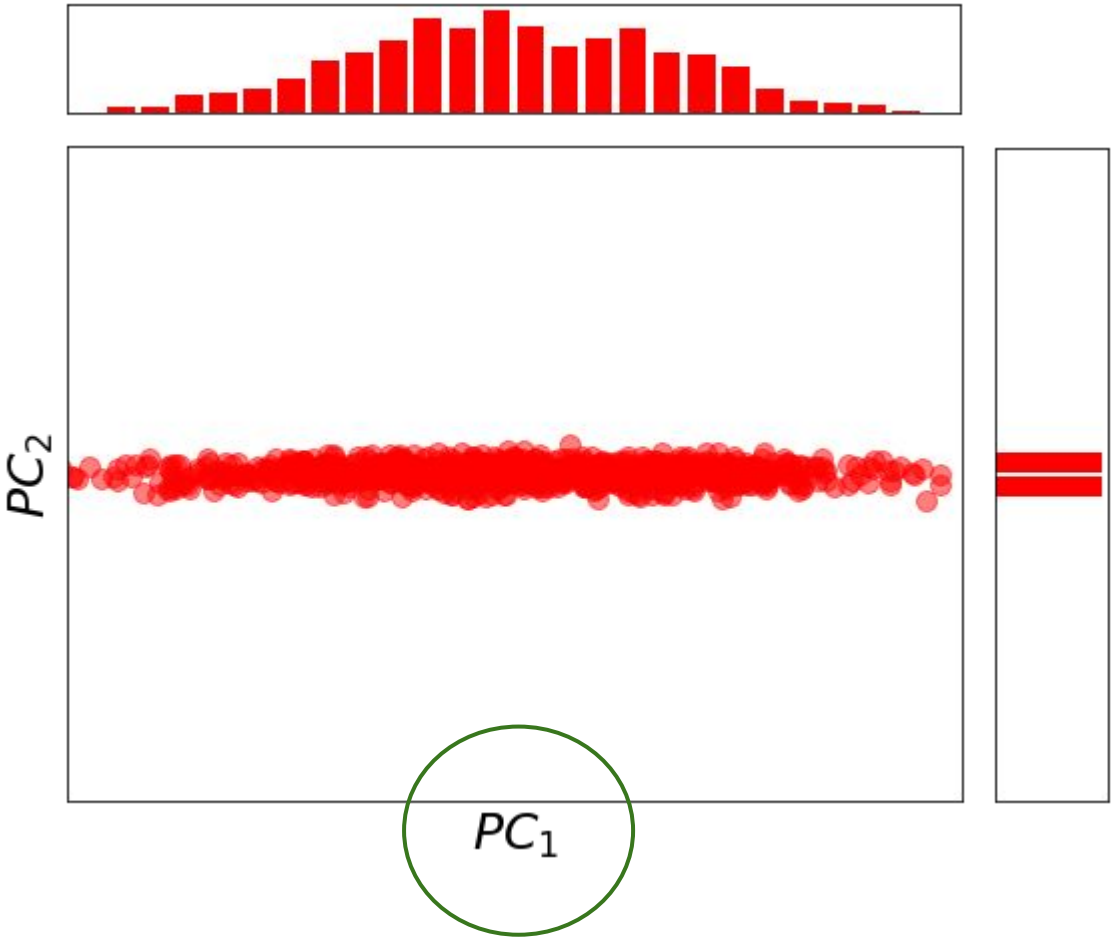
Sin embargo, notamos direcciones (combinación lineal de features) donde los datos parecen variar más.

Esquema general



Proyectando en esas nuevas direcciones (ortogonales) volvemos a ver la distribución de los datos proyectadas en las mismas.

La componente 2 tiene mucha menos variación (parece ser más ruido que información importante)



Esquema general

Notar que si nos olvidamos de la componente 2, no perdemos tanta información como si hubiésemos tirado alguno de los features originales.

Podemos reducir nuestro problema de 2 dimensiones a una sola.

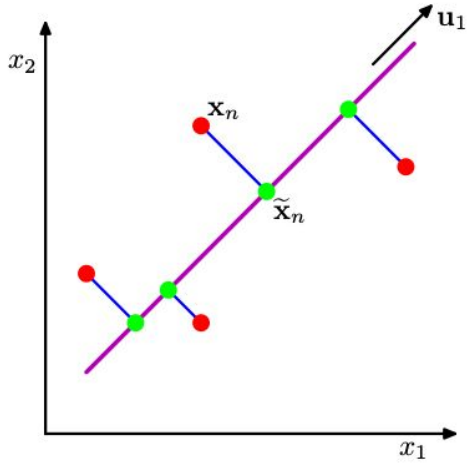
Las direcciones que se llevan las mayor cantidad de varianza de los datos se llaman **componentes principales**

PCA: descripción matemática (Bishop, capítulo 12)

Dado un conjunto $\{\mathbf{x}_n\}$ de N datos en un espacio de dimensión D (cada $\mathbf{x}_n$ es un vector en el espacio de D dimensiones).

Las M < D componentes principales son:

- las M direcciones que **maximizan la varianza** de las proyecciones en ese subespacio,
- o, equivalentemente, las M direcciones que **minimizan el error** en la proyección (figura).



PCA: descripción matemática (Bishop, capítulo 12)

Ambas ideas llevan a que las componentes principales son los autovectores de la matriz de covarianza **S**:

$$\bar{\bar{S}}\bar{u} = \lambda\bar{u}$$

Componentes principales

Autovalores: varianza en la dirección de la componente correspondiente.

Donde: $$S_{ij} = \frac{1}{N}\sum_{n}^{N}(x_{ni} - \bar{x_i})(x_{nj} - \bar{x_j})$$

Valor medio del feature j

Valor del feature j la instancia n

Pidiendo además: $$\sum_i u_i^2 = 1$$

PCA: descripción matemática (Bishop, capítulo 12)

Ambas ideas llevan a que las componentes principales son los autovectores de la matriz de covarianza **S**:

$$\bar{\bar{S}}\bar{u} = \lambda\bar{u}$$

Componentes principales

Autovalores: varianza en la dirección de la componente correspondiente.

- Las componentes principales son ortogonales.
- Ordenando las componentes desde el autovalor más grande al más chico, las primeras componentes llevan la mayor varianza de los datos (es lo que efectivamente hace PCA).

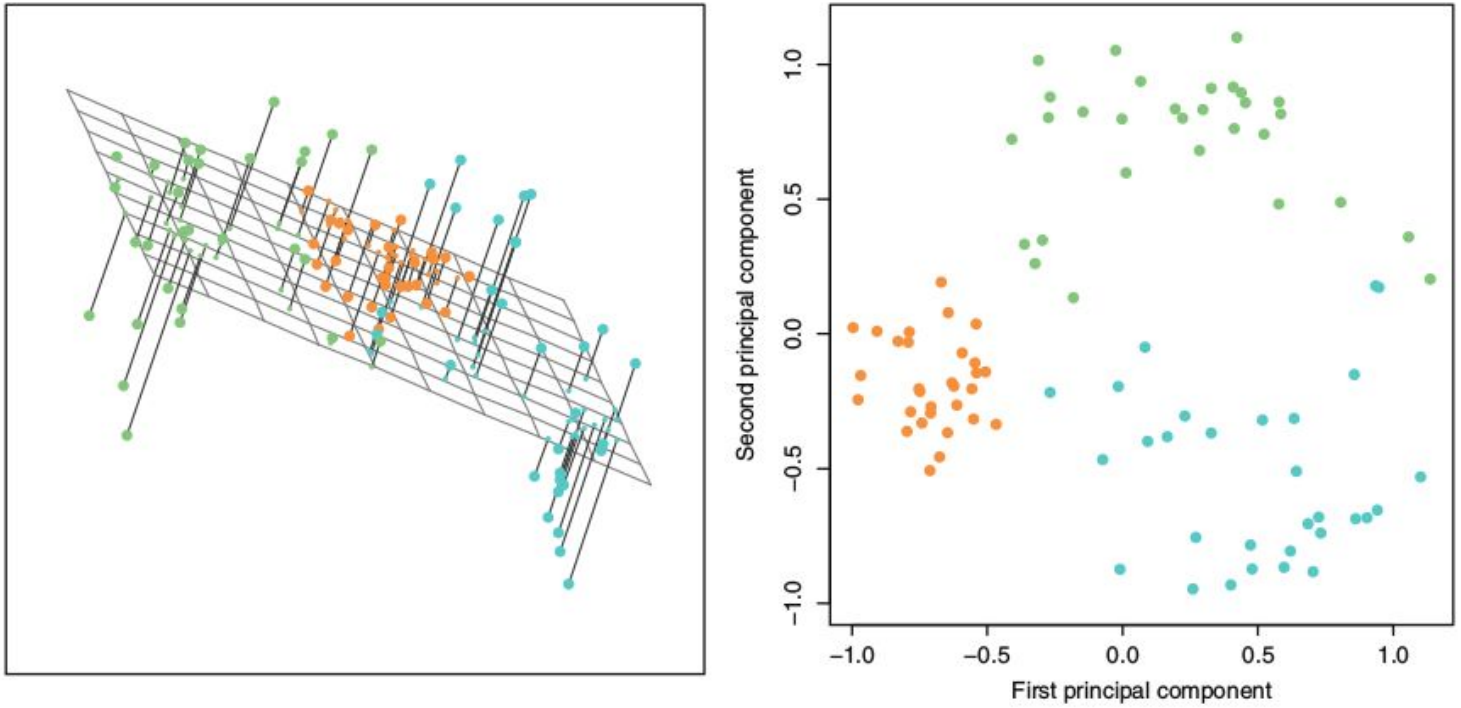
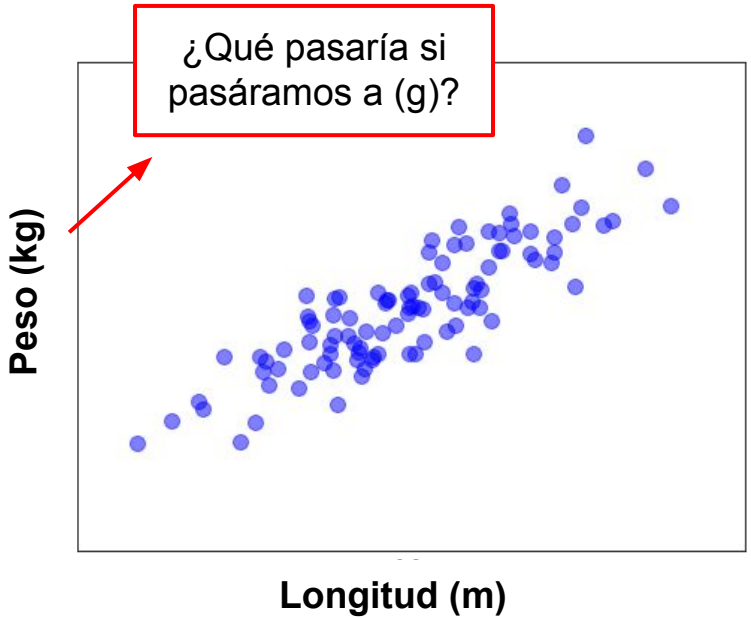


FIGURE 10.2. *Ninety observations simulated in three dimensions.* Left: *the first two principal component directions span the plane that best fits the data. It minimizes the sum of squared distances from each point to the plane.* Right: *the first two principal component score vectors give the coordinates of the projection of the 90 observations onto the plane. The variance in the plane is maximized.*

Dependencia de las unidades y escaleo

¿Qué pasa con la variabilidad si cambiamos las unidades de alguna de las variables?

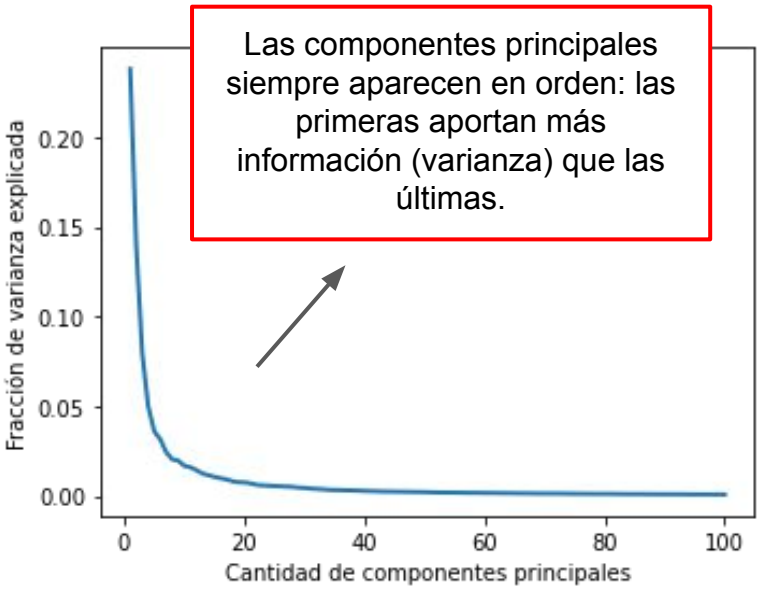


Para evitar introducir o sacar variabilidad por cambiar las unidades de alguna de las variables, una práctica habitual antes de hacer PCA es estandarizar las variables:

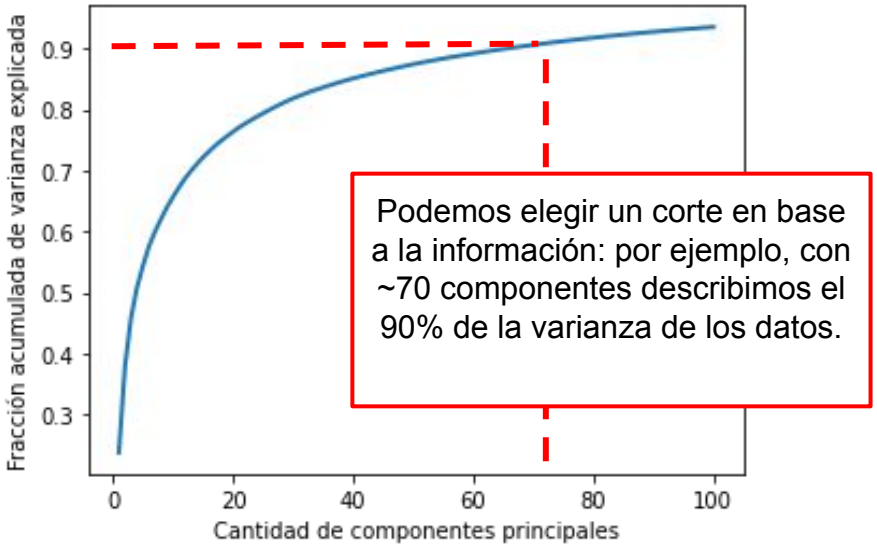
$$Z = \frac{X - \mu}{\sigma}$$

¿Cuántas componentes?

Al ser un problema de aprendizaje no-supervisado, no tenemos un conjunto de test para validar el número de componentes que elijamos. ¿Con cuántas nos quedamos?



Fracción de varianza que aporta cada componente



Fracción de varianza acumulada

Reducción dimensional como input de otros modelos

Podemos usar las primeras componentes principales (Z_m) como variables independientes en modelos de regresión o clasificación.

$$y \sim \beta_0 + \beta_1 Z_1 + \ldots + \beta_M Z_M$$

Ventajas:

- En regresión lineal se pide que las variables independientes sean no colineales. En PCA las variables son por defecto ortogonales y por lo tanto buenas candidatas.
- Al ser estos modelos de aprendizaje supervisado podemos validar nuestro modelo y por lo tanto tomar la cantidad de componentes principales como un hiperparámetro.

Principal Components Regression. Sección 6.3.1 del libro Introduction to Statistical Learning.

Resumen de PCA

- Las componentes principales son una combinación lineal de los features originales y se corresponden con los autovectores de la matriz de covarianza de los datos.
- Las componentes están ordenadas de mayor a menor, en el sentido de la información (varianza) que se llevan. Si tiramos las últimas componentes, estamos reduciendo la dimensión de nuestro problema.
- Una práctica usual es estandarizar las variables antes de aplicar PCA (fundamental si los features corresponden a distintas magnitudes).
- Podemos usar las componentes principales para: comprimir la información, visualizar datos multidimensionales en bajas dimensiones, y como input para modelos de aprendizaje supervisado.

Bibliografía

- Sección 6.3 y Capítulo 10. Introduction to Statistical Learning. Hastie, et.al.

- Capítulo 12. Pattern Recognition and Machine Learning. Bishop.

PCA en scikit-learn

sklearn.decomposition.PCA

class `sklearn.decomposition.` **PCA**(*n_components=None, *, copy=True, whiten=False, svd_solver='auto', tol=0.0, iterated_power='auto', random_state=None*) [source]

Parámetro más importante: número de componentes.

https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html?highlight=pca#sklearn.decomposition.PCA