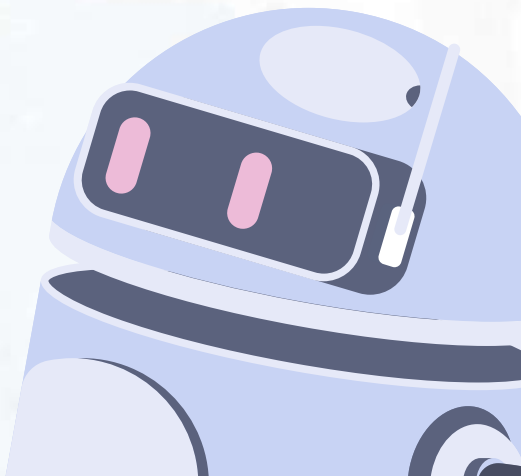


# Analítica de Datos (Aprendizaje de máquina)



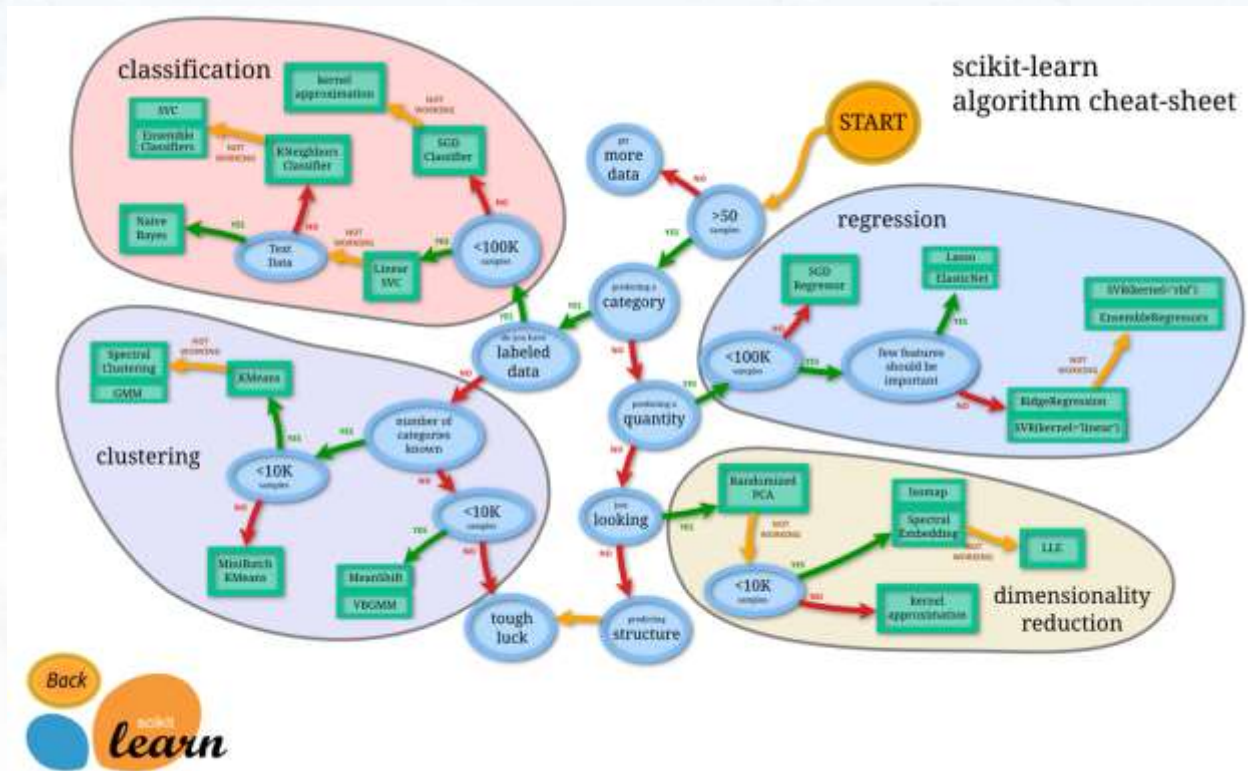
UNAL

Departamento de Eléctrica, Electrónica y Computación  
Facultad de Ingeniería y Arquitectura  
Sede Manizales



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

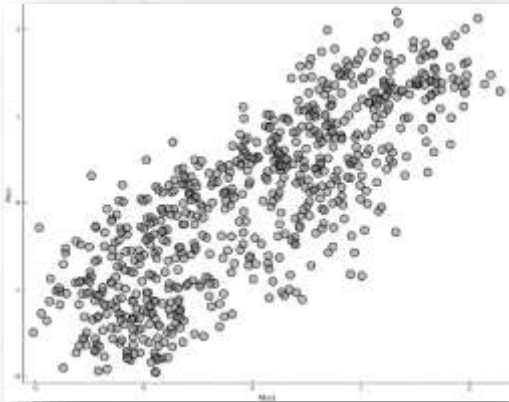
# Machine learning - Guideline



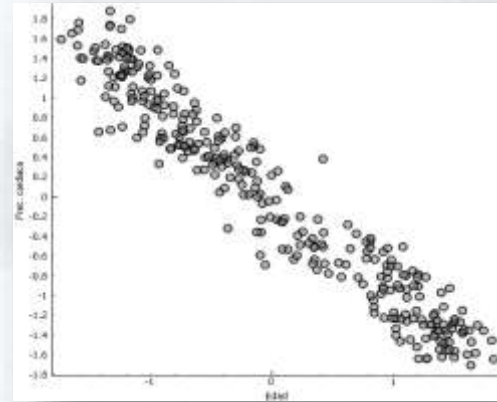
# Buscando relaciones

## Correlación

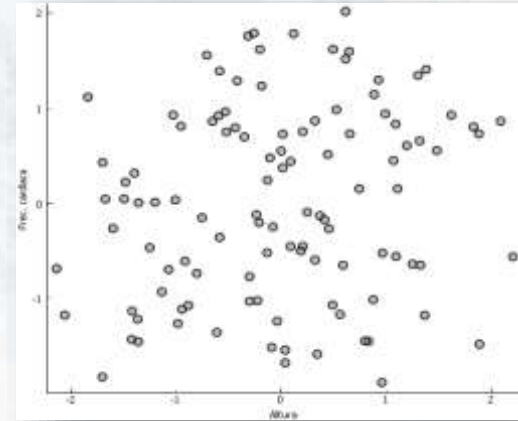
Cualquier **relación lineal** existente entre dos variables de un conjunto de datos se conoce como **correlación**.



*Correlación positiva*



*Correlación negativa*



*Sin Correlación*

# Buscando relaciones

## Correlación - Definición

La medida más familiar de dependencia entre dos cantidades es el **"Coeficiente de Correlación de Pearson"**.

Dependiendo del signo de nuestro coeficiente de correlación de Pearson, podemos terminar con una **correlación negativa** o **positiva** si existe algún tipo de relación entre las variables de nuestro conjunto de datos.

$$\rho_{X,Y} = \text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}, \quad \text{if } \sigma_X \sigma_Y > 0.$$

$$\rho_{X,Y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E(X)^2} \cdot \sqrt{E(Y^2) - E(Y)^2}}$$

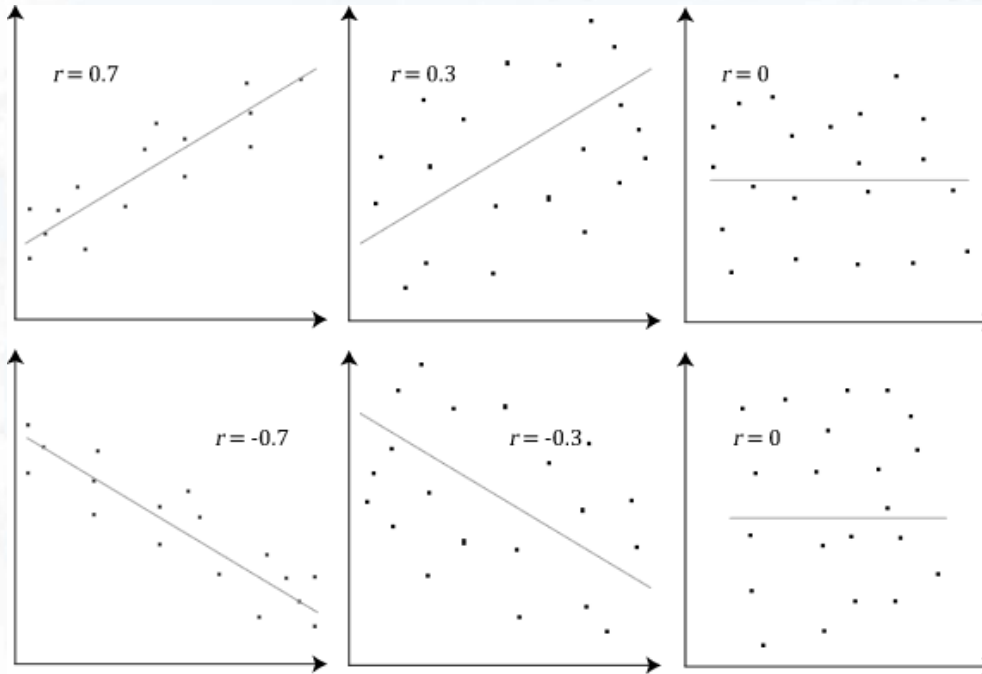
# Buscando relaciones

## Correlación y dependencia

- El **valor absoluto** del coeficiente de correlación de Pearson **no es mayor que 1**.
- El valor de un coeficiente de correlación **oscila entre  $-1$  y  $+1$**
- El coeficiente de correlación es  **$+1$**  en el caso de una relación lineal (correlación) directa (creciente) perfecta.
- El coeficiente de correlación es  **$-1$**  en el caso de una relación lineal (anti-correlación) inversa (decreciente) perfecta.
- Algún valor en el intervalo abierto  **$(-1,1)$**  indicando el **grado de dependencia lineal** entre las variables
- A medida que se acerca **a cero**, hay menos relación (más cercana a no correlacionada).

# Buscando relaciones

## Correlación y dependencia

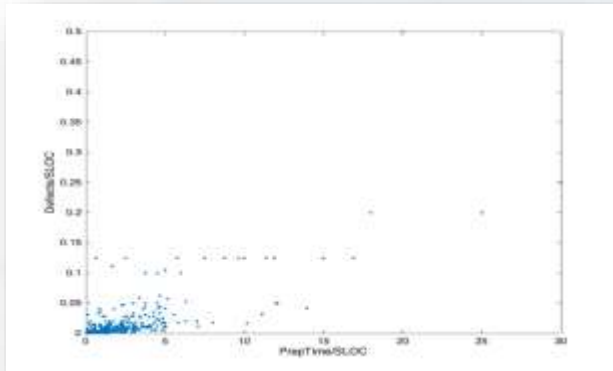


Diagramas de dispersión de ejemplo de varios conjuntos de datos con **varios coeficientes de correlación.**

# Buscando relaciones

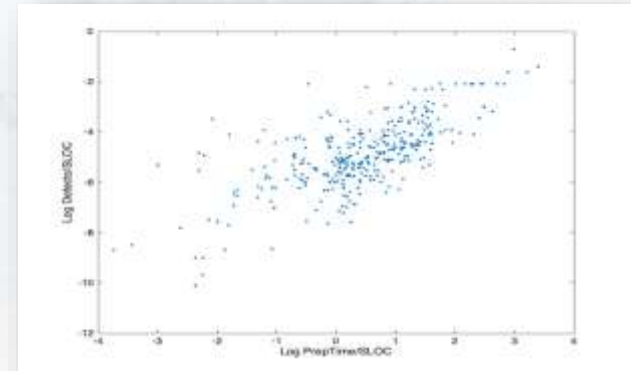
## Correlación y dependencia

- En muchas ocasiones la correlación **NO** es evidente. Es posible que tengamos que transformar los datos para encontrarla.



No hay correlación evidente

Calculando el logaritmo a cada una de las variables



Aparece una correlación positiva



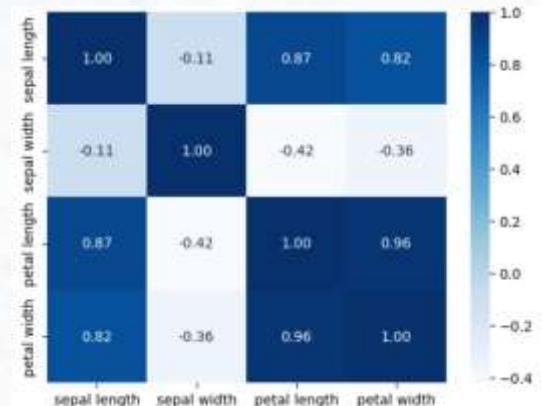
# Buscando relaciones

## Correlación - Mapa de correlación

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
longitude	1.000000	-0.924478	-0.105823	0.048909	0.076686	0.108071	0.063146	-0.019615	-0.047466
latitude	-0.924478	1.000000	0.005737	-0.039245	-0.072550	-0.115290	-0.077765	-0.075146	-0.142673
housing_median_age	-0.105823	0.005737	1.000000	-0.364535	-0.325101	-0.298737	-0.306473	-0.111315	0.114146
total_rooms	0.048909	-0.039245	-0.364535	1.000000	0.929391	0.855103	0.918396	0.200133	0.135140
total_bedrooms	0.076686	-0.072550	-0.325101	0.929391	1.000000	0.876324	0.980167	-0.009643	0.047781
population	0.108071	-0.115290	-0.298737	0.855103	0.876324	1.000000	0.904639	0.002421	-0.026882
households	0.063146	-0.077765	-0.306473	0.918396	0.980167	0.904639	1.000000	0.010869	0.064590
median_income	-0.019615	-0.075146	-0.111315	0.200133	-0.009643	0.002421	0.010869	1.000000	0.687151
median_house_value	-0.047466	-0.142673	0.114146	0.135140	0.047781	-0.026882	0.064590	0.687151	1.000000

Base de datos Housing California

Base de datos Iris

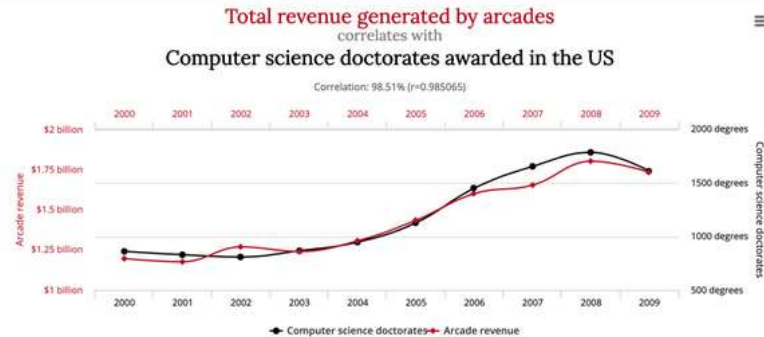
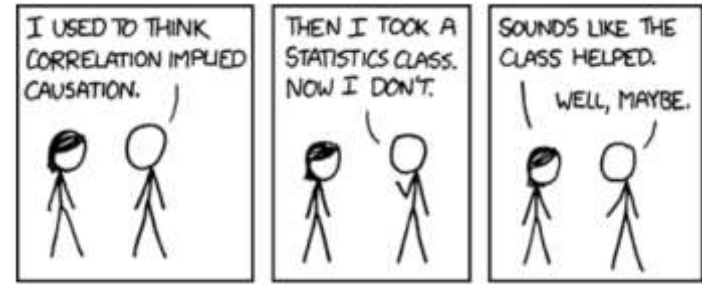




# Buscando relaciones

## Correlación - Causalidad

Es un error común **confundir correlación con causalidad**. Dos fenómenos pueden estar correlacionados y que aún así no tengan realmente una relación.



Tenemos **causalidad** cuando un fenómeno lo causa otro

A veces es difícil determinar si hay **causalidad** y en qué sentido sucede

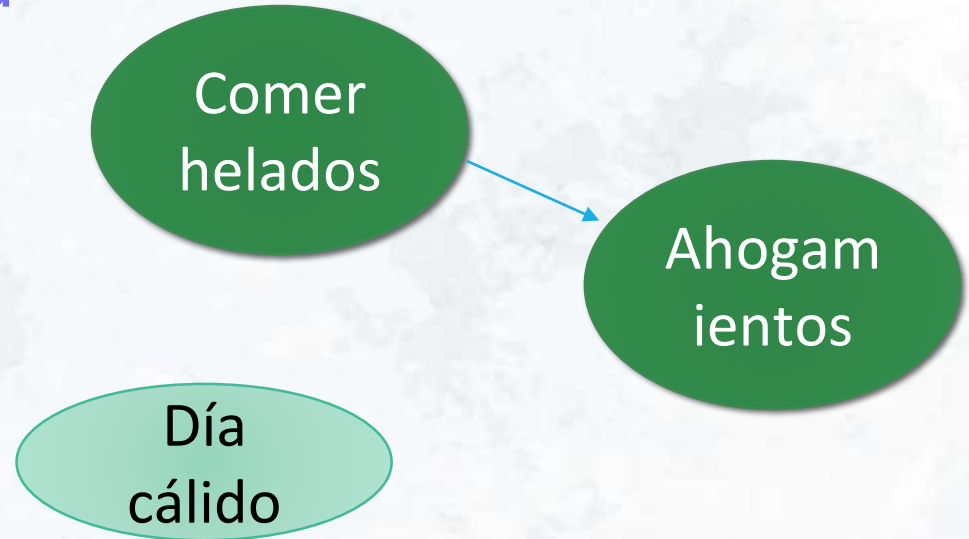
# Buscando relaciones

## Correlación - Causalidad

Correlación no implica Causalidad

Si cuando la gente consume más helados también se ahoga más gente... ¿será que comer helados nos ahoga?

¿O tal vez es que ahogarse le provoca a uno ganas de comer helado?



# Selección y extracción de características

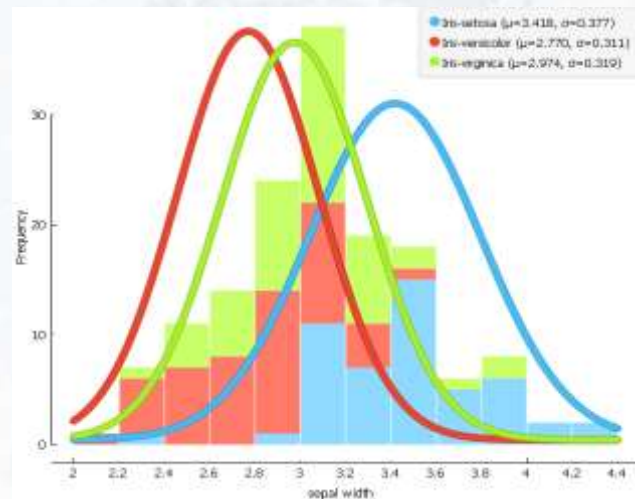
En casos de la **vida real**, **no todas las características disponibles** tienen información relevante para la tarea de análisis que se esté llevando a cabo.

Un **ejemplo** de esto se puede ver en la **base de datos de Iris**. En la cual **para diferenciar entre los tipos de esta flor** no todas las características son necesarias.



Scoring Methods		
<input checked="" type="checkbox"/>	Information Gain	
<input type="checkbox"/>	Information Gain Ratio	
Select Attributes		
<input type="radio"/>	None	
<input type="radio"/>	All	

	#	Info. gain
1	N petal length	1.086
2	N petal width	1.059
3	N sepal length	0.624
4	N sepal width	0.361



# Selección y extracción de características

## Selección:

Existen varias metodologías para la selección de características. Estas metodologías o métodos calculan una medida que indica la importancia de las características según algún criterio fijo. Por medio de esta medida se realiza un **ranking** de importancia.

Estas metodologías se separan en:

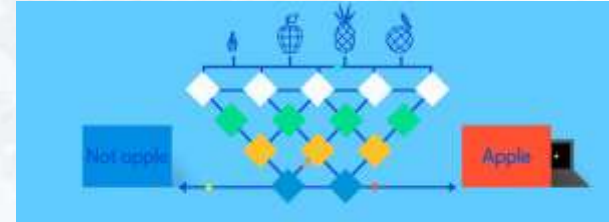
- **Supervisadas:** se requiere de la variable objetivo o target para definir qué características son las importantes.
- **No-supervisadas:** no se requiere de la variable objetivo para definir niveles de importancia de las características bajo estudio.

# Selección y extracción de características

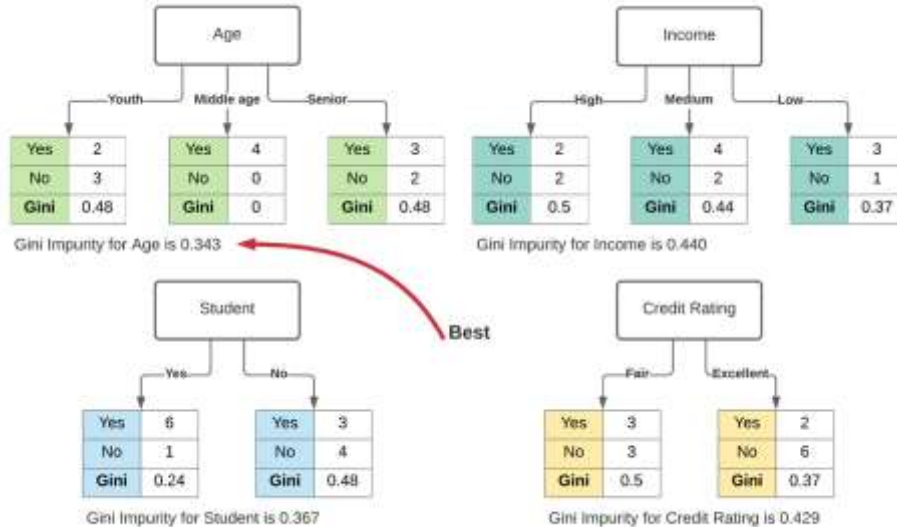
## Selección - SUPERVISADAS:

Dentro de las técnicas supervisadas para la selección de las características más relevantes se encuentran metodologías que:

- Hacen uso de métricas para realizar el ranking. Entre estas medidas, las más usadas son:
  - Gini
  - Entropía/Ganancia de Información
  - ANOVA
  - Chi-Cuadrado
- Metodologías que hacen uso de un método de clasificación (target categórico) o de regresión (target numérico)
  - ReliefF
  - RFE (Recursive Feature Elimination)



# Selección y extracción de características



## Selección - SUPERVISADAS:

- **GINI:** Este es un **coeficiente** obtenido de dos variables/atributos para **medir el nivel de desigualdad entre dos variables**, muy usado en economía.

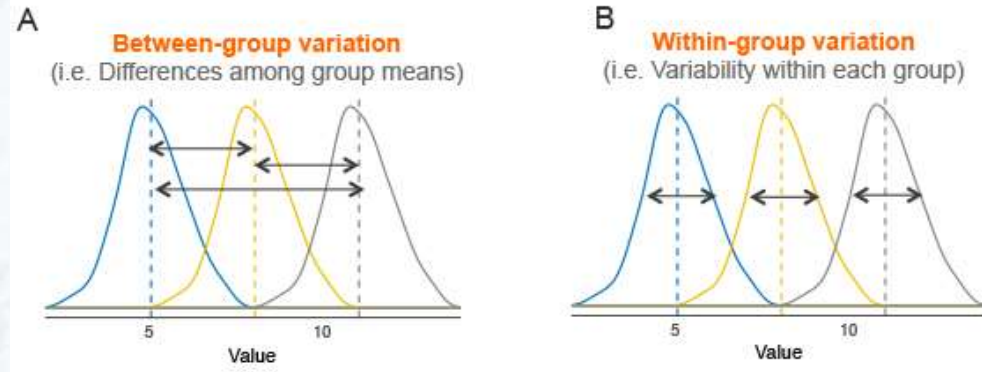
$$Gini(D) = 1 - \sum_{i=1}^k p_i^2$$



# Selección y extracción de características

## Selección - SUPERVISADAS:

- **ANOVA:** Es de los test estadísticos más conocidos, el cual tiene como hipótesis nula que las medias de dos variables son iguales. Adicional, esta técnica hace uso de la Distribución F, y se hace uso de su score para ranquear los atributos por nivel de importancia.



La idea detrás de la prueba **ANOVA** es muy simple: si la variación promedio entre grupos es lo suficientemente grande en comparación con la variación promedio dentro de los grupos, entonces se podría concluir que al menos la media de un grupo no es igual a las demás.

# Selección y extracción de características

The Formula for Chi Square Is

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

where:

$c$  = degrees of freedom

$O$  = observed value(s)

$E$  = expected value(s)

## Selección - SUPERVISADAS:

- **CHI-cuadrado:** También es un popular test estadístico para saber si dos variables (categóricas) son independientes o no. Esta hace uso de la distribución de Chi-cuadrado para determinar lo anterior.

Cuando dos características **son independientes**, el recuento observado se acerca al recuento esperado, por lo que tendremos un valor de Chi-cuadrado más pequeño. Un valor tan alto de Chi-cuadrado indica que la hipótesis de independencia es incorrecta.

En palabras simples, cuanto mayor sea el valor de Chi-cuadrado, la característica dependerá más de la respuesta y se puede seleccionar para el entrenamiento del modelo.

# Selección y extracción de características

## Selección - SUPERVISADAS:

- **Relief:** Dado un ejemplo, Relief busca a sus *dos vecinos más cercanos*, uno de la misma clase y otro de una clase diferente, y actualiza los pesos de los atributos involucrados dependiendo de si sus valores son iguales o no a estos ejemplos.
- La idea es favorecer atributos que tengan valores diferentes en ejemplos parecidos de diferente clase y valores iguales en ejemplos parecidos de la misma clase.

```
for  $i = 1$  to  $n$  do
```

```
  Selecciona aleatoriamente un ejemplo  $E$  de una clase  
  Encuentra el ejemplo de la misma clase más cercano  $P$   
  y el ejemplo de otra clase más cercano  $N$ 
```

```
  for  $A := 1$  to Num. de atributos do
```

```
     $W[A] := W[A] - \text{diff}(A, E, P)/n + \text{diff}(A, E, N)/n$ 
```

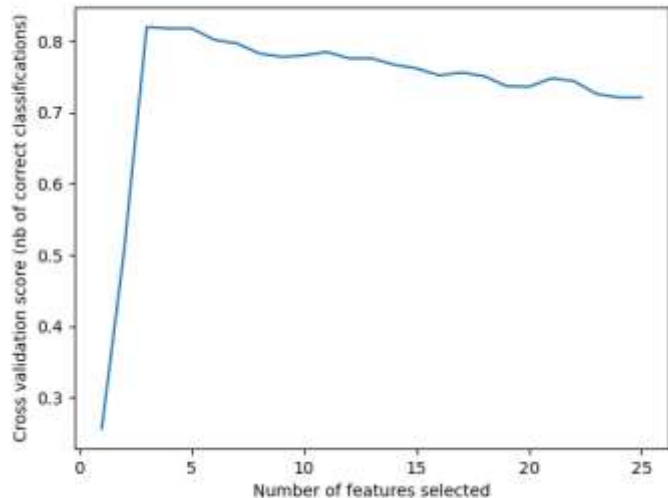
```
  end for
```

```
end for
```

# Selección y extracción de características

## Selección - SUPERVISADAS:

- **Eliminación recursiva de características:** Esta metodología hace uso de un clasificador (target categórico) o de un regresor (target numérico).
  - Se aplica el método en cuestión sobre la base de datos y se calcula **su rendimiento** (acierto, error cuadrático medio, etc.)
  - Se realiza el procedimiento anterior eliminando, de forma aleatoria, atributos.
  - Los atributos son rankeados según la pérdida de rendimiento con respecto al caso donde se usan todos ellos.



# Selección y extracción de características

## Selección - NO SUPERVISADAS:

- ¿Cuáles son los métodos más comunes de selección de características no supervisadas y en qué situaciones son especialmente efectivos?
- ¿Cuáles son las métricas típicas utilizadas para evaluar la calidad de las características seleccionadas por algoritmos no supervisados, como la **reducción de dimensionalidad** o la mejora en la representación de datos?



# Extracción de características

## Reducción de dimensión

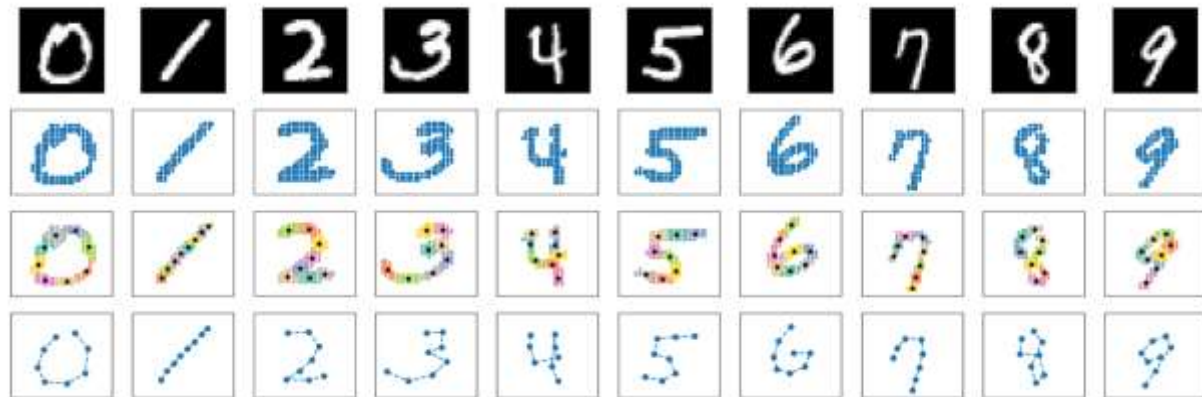
El enfoque de la extracción de características como su nombre lo indica es la extracción de “nuevos” atributos a partir de los ya existentes.

Lo anterior se hace con dos objetivos principales en mente:

- Reducir el número de atributos. Con esto se reduce el efecto de la maldición de la dimensionalidad.
- Tener una “mejor” representación de la base.

Estas metodologías pueden ser:

- No supervisadas
- Supervisadas





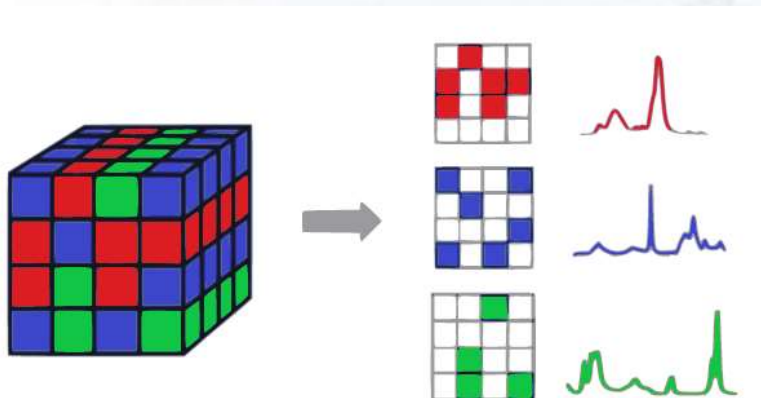
# Extracción de características

## Reducción de dimensión

Muchos problemas de Machine Learning implican millones de características para cada instancia de entrenamiento.

- Extremadamente lento
- Es mucho más difícil encontrar una buena solución.

Este problema a menudo se conoce como **curse of dimensionality**.



En el mundo real, es posible **reducir considerablemente el número de características**, convirtiendo un “intractable problem” en un “tractable problem”.

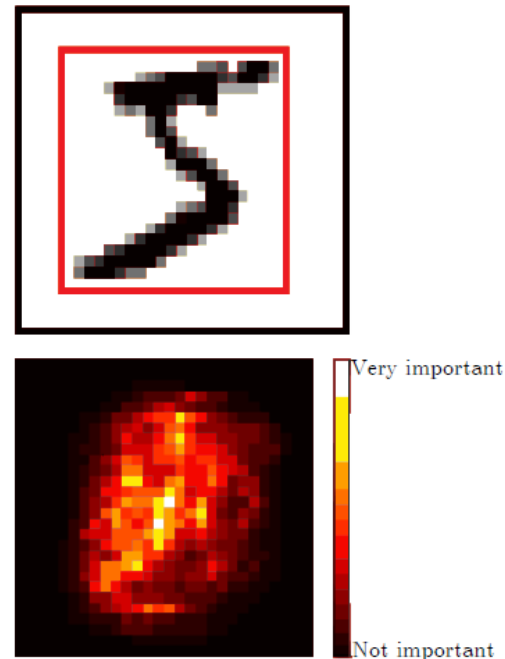
# Extracción de características

## Reducción de dimensión - Ejemplo

Consideremos las imágenes de la base de datos **MNIST**:

- Los píxeles en los bordes de la imagen casi siempre son blancos, por lo que puede eliminar estos píxeles sin perder mucha información.
- La imagen de relevancia, confirma que estos píxeles no son importantes para la tarea de clasificación.

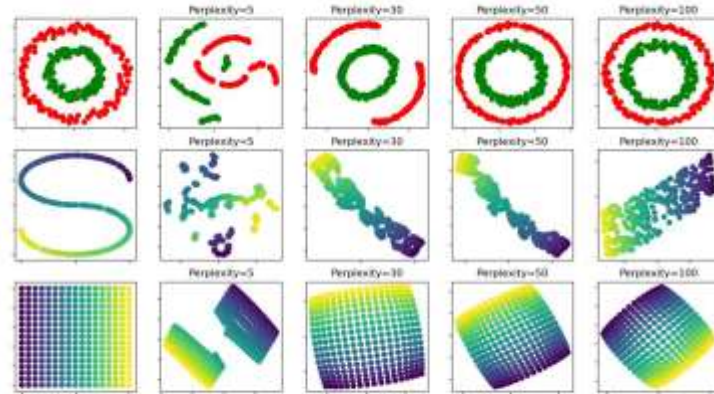
En el proceso de reducción de la dimensionalidad se pierde algo de información.



# Extracción de características

## Reducción de dimensión - Tips

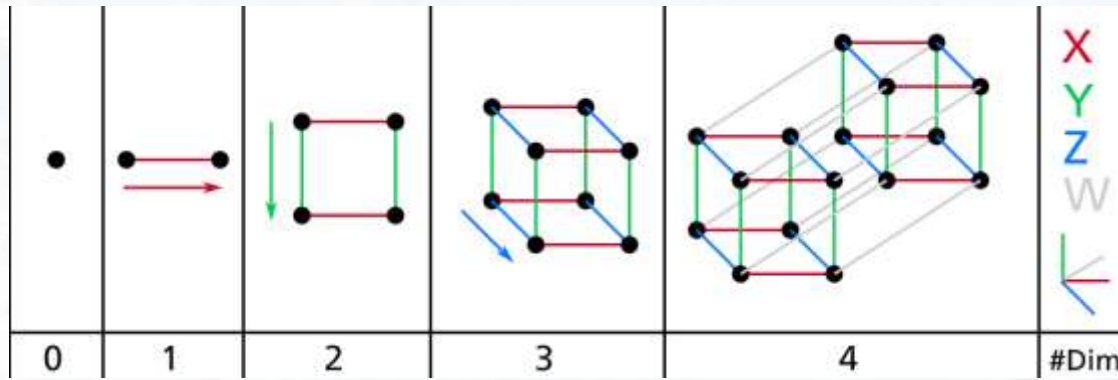
- En algunos casos, reducir la dimensionalidad de los datos de entrenamiento puede filtrar algunos ruidos y detalles innecesarios y, por lo tanto, **generar un mayor rendimiento**.
- Además de acelerar el entrenamiento, la reducción de dimensionalidad también es extremadamente útil para la visualización de datos.



# Extracción de características

## Reducción de dimensión - Curse of Dimensionality

Estamos tan acostumbrados a vivir en **tres dimensiones** que nuestra intuición nos falla cuando tratamos de imaginar un espacio de alta dimensión.

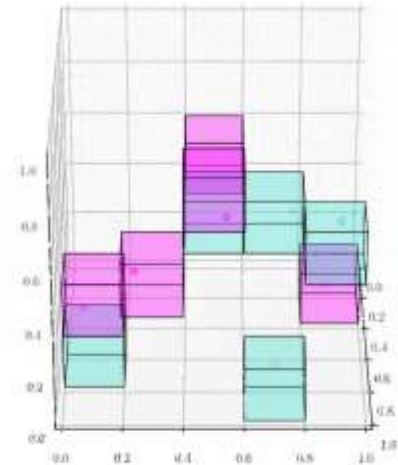
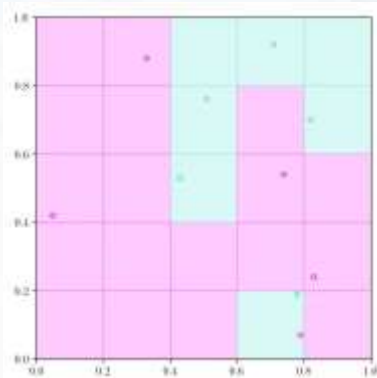
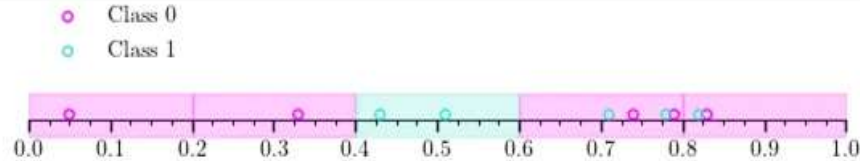


Resulta que muchas cosas se comportan de manera muy diferente en espacios de alta dimensión

# Extracción de características

## Reducción de dimensión - Curse of Dimensionality

Consiste en que a mayor sea la dimensionalidad de los datos, mayor será la cantidad de muestras para tener una buena representación para la tarea a realizar.



# Extracción de características

## Reducción de dimensión - Curse of Dimensionality

**Solución:** Aumentar el tamaño del conjunto de entrenamiento para alcanzar una densidad suficiente de instancias de entrenamiento.

Sin embargo, **el número de instancias de entrenamiento requeridas** para alcanzar una densidad dada **crece exponencialmente con el número de dimensiones**.

- Con solo **100 características**, necesitaría más instancias que átomos en el universo.

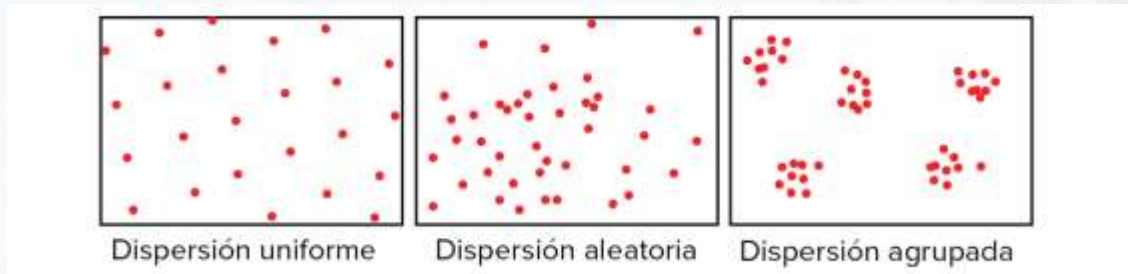




# Extracción de características

## Reducción de dimensión - Enfoques: Proyección

- En los problemas del mundo real, las instancias de entrenamiento no se distribuyen de manera uniforme en todas las dimensiones.



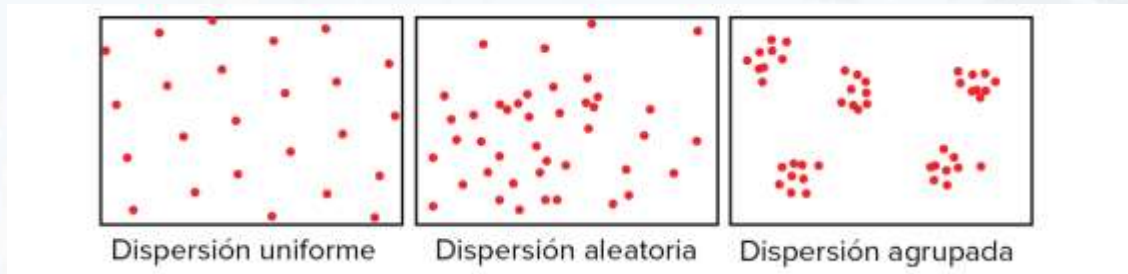
- Muchas características son casi constantes, mientras que otras están altamente correlacionadas.

Todas las instancias de entrenamiento realmente se encuentran dentro (o cerca) de un subespacio de dimensiones mucho más bajas.

# Extracción de características

## Reducción de dimensión - Enfoques: Proyección

- En los problemas del mundo real, las instancias de entrenamiento no se distribuyen de manera uniforme en todas las dimensiones.



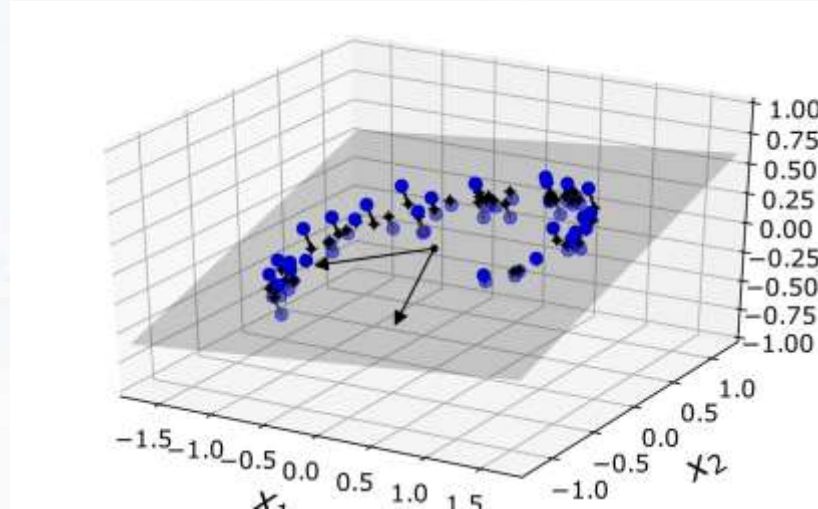
- Muchas características son casi constantes, mientras que otras están altamente correlacionadas.

Todas las instancias de entrenamiento realmente se encuentran dentro (o cerca) de un subespacio de dimensiones mucho más bajas.

# Extracción de características

## Reducción de dimensión - Enfoques: Proyección - Ejemplo I

En la Figura se puede ver un conjunto de datos en 3D representado por círculos.



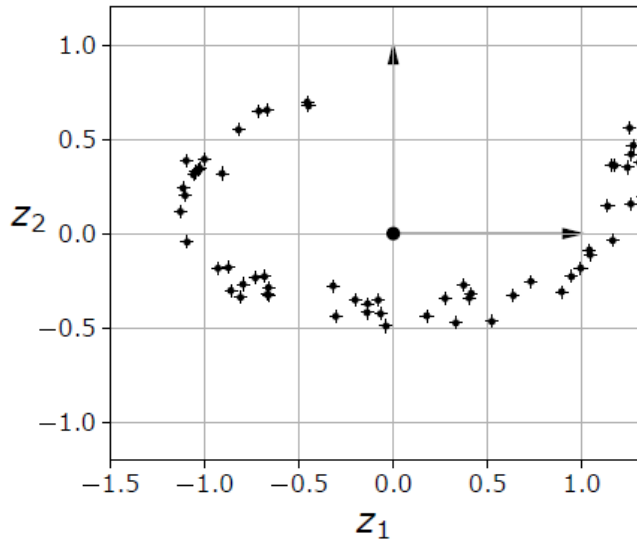
Todas las instancias de entrenamiento se encuentran cerca de un plano: este es un subespacio de menor dimensión (2D) del de alta dimensión (3D).

# Extracción de características

## Reducción de dimensión - Enfoques: Proyección - Ejemplo I

Si proyectamos cada instancia de entrenamiento perpendicularmente en este subespacio, obtenemos el nuevo conjunto de datos 2D:

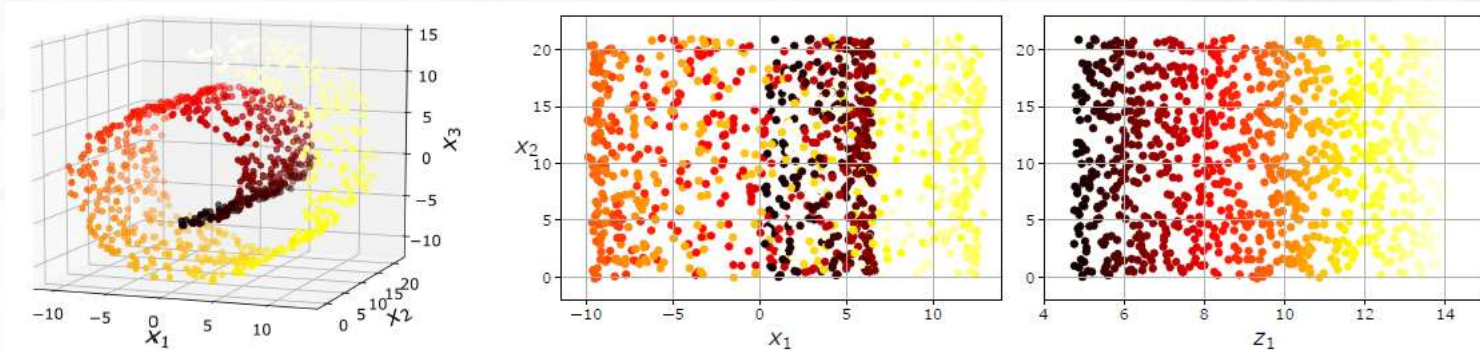
- Tengamos en cuenta que los ejes corresponden a las nuevas características  $z_1$  y  $z_2$ .
- Sin embargo, la proyección no siempre es el mejor enfoque para la reducción de la dimensionalidad.



# Extracción de características

## Reducción de dimensión - Enfoques: Proyección - Ejemplo II

En muchos casos, el subespacio puede girar y girar, como en el famoso conjunto de datos “Swiss roll”:



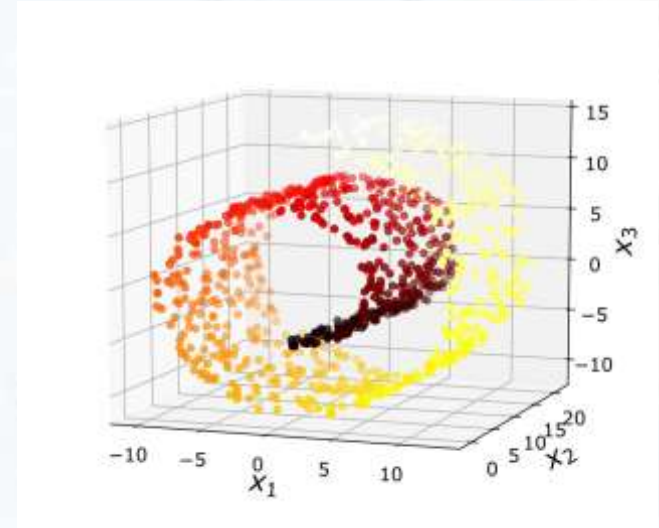
- Proyectar en un plano (por ejemplo, quitando  $x_3$ ) aplastará diferentes capas del rollo suizo (Centro).
- Lo que realmente deseamos es desenrollar el rollo suizo para obtener el conjunto de datos 2D (Derecha).

# Extracción de características

## Reducción de dimensión - Enfoques: Proyección - Manifold Learning

- El “Swiss roll” es un ejemplo de una variedad 2D.
- Un múltiple 2D es una forma 2D que se puede doblar y torcer en un espacio de dimensiones superiores.
- Una variedad d-dimensional es una parte de un espacio n-dimensional (donde  $d < n$ ) que localmente se parece a un hiperplano d-dimensional.

**Manifold assumption:** la mayoría de los conjuntos de datos de alta dimensión del mundo real se encuentran cerca de un manifold de mucha más baja dimensión.

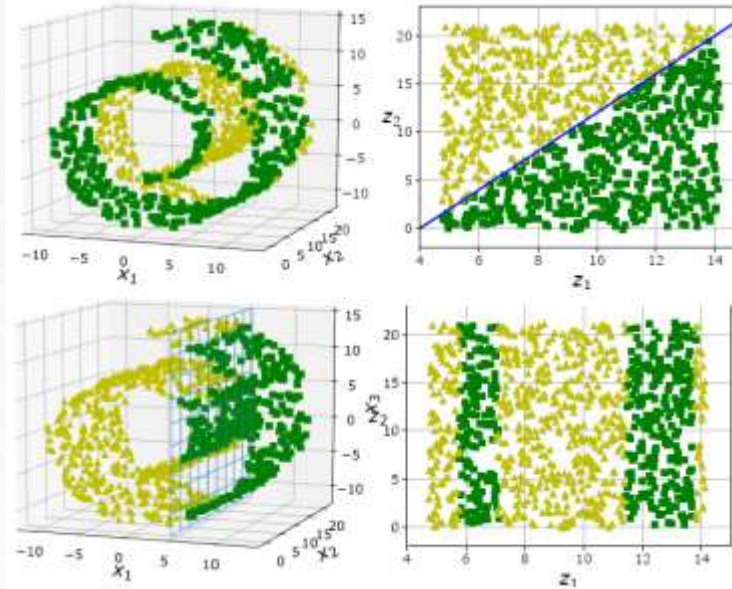




# Extracción de características

## Reducción de dimensión - Enfoques: Proyección - Manifold Learning

**El Implicit assumption:** la tarea en cuestión (por ejemplo, clasificación o regresión) será más simple si se expresa en el espacio de dimensiones inferiores del manifold.



# Extracción de características

## Reducción de dimensión - Enfoques: Proyección - Manifold Learning

- En la **fila superior de la Figura**, el rollo suizo se divide en dos clases: en el espacio 3D (a la izquierda), **el límite de decisión sería bastante complejo**, pero en el espacio múltiple desenrollado 2D (en el derecha), **el límite de decisión es una simple línea recta**.
- De lo contrario, en la **fila inferior de la Figura**, el límite de decisión se encuentra en  $x_1 = 5$ . **Este límite de decisión parece muy simple en el espacio 3D original**, pero se ve más complejo en el múltiple desenrollado.

Si se reduce la dimensionalidad antes de entrenar un modelo, acelerará el entrenamiento, **pero no siempre puede conducir a una solución mejor.**

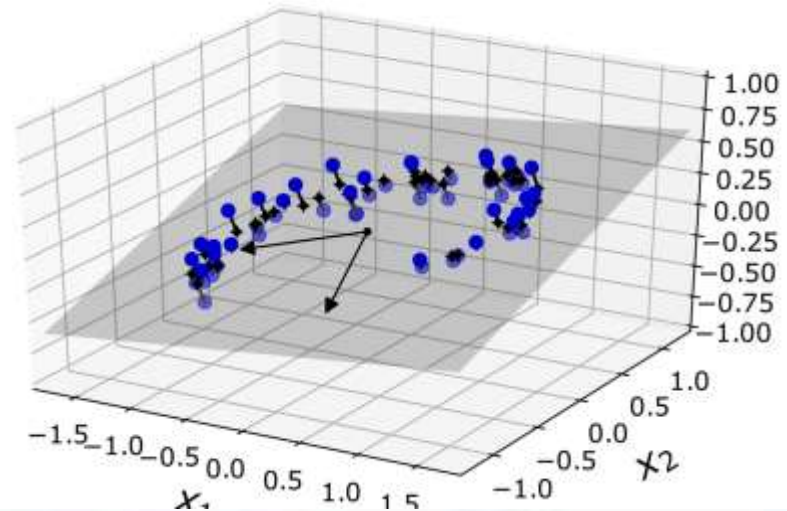
# Extracción de características

## Reducción de dimensión - Análisis de componentes principales (PCA)

PCA es el algoritmo de reducción de dimensionalidad más popular.

### ¿Cómo funciona?:

1. Identifica **el hiperplano** que se encuentra más cerca de los datos.
2. Luego **proyecta los datos** sobre él:

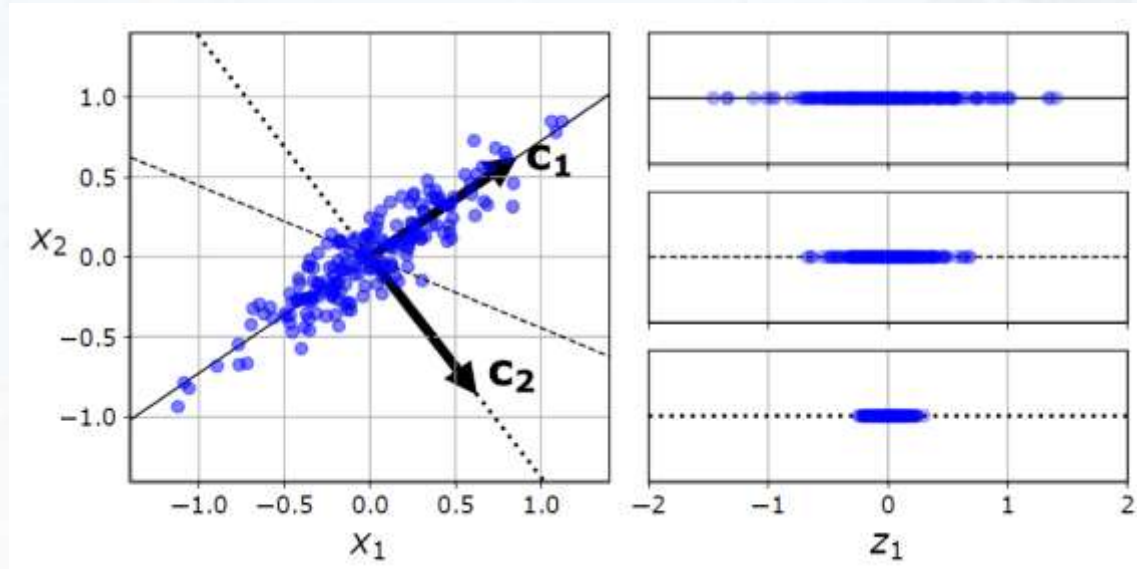


# Extracción de características

## Reducción de dimensión - PCA: Preservando la varianza

Antes de poder **proyectar** el conjunto de entrenamiento en un hiperplano de dimensiones inferiores, primero debemos elegir el hiperplano correcto.

- **Ejemplo:**



# Extracción de características

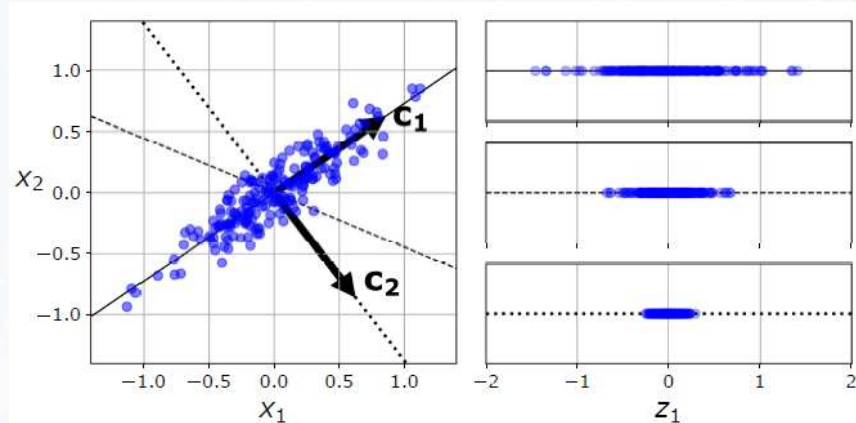
## Reducción de dimensión - PCA: Preservando la varianza

- En la **Figura de la izquierda** tenemos un conjunto de datos 2D, junto con tres ejes diferentes.
- En la **Figura de la derecha** está el resultado de la proyección de los datos en cada uno de estos ejes.
- **Como podemos ver:**
  - La proyección en la **línea continua** conserva la varianza máxima.
  - La proyección en la **línea punteada** conserva muy poca varianza.
  - La proyección en la **línea discontinua** conserva una cantidad intermedia de varianza.

Se selecciona **el eje que preserva la cantidad máxima de variación**, el que **minimiza la distancia cuadrática media** entre el conjunto de datos original y su proyección sobre ese eje.

# Extracción de características

## Reducción de dimensión - PCA: componentes principales

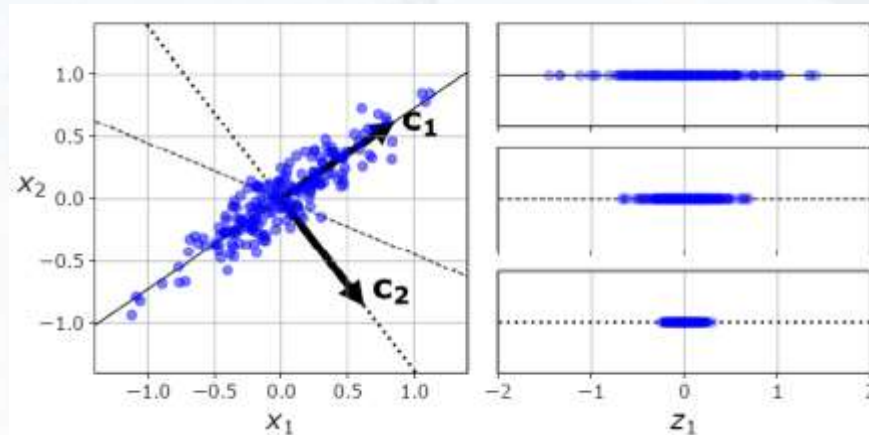


- PCA identifica el eje que representa la mayor cantidad de variación en el conjunto de entrenamiento (línea continua).
- También encuentra un segundo eje, ortogonal al primero, que representa la mayor cantidad de varianza restante (línea punteada).
- PCA encontraría tantos ejes como el número de dimensiones en el conjunto de datos.

# Extracción de características

## Reducción de dimensión - PCA: componentes principales

- El vector unitario que define el  $i$ -ésimo eje se llama  $i$ -ésimo componente principal (PC). En la Figura, la primera PC es  $c_1$  y la segunda PC es  $c_2$ .
- Las dos primeras PC están representadas por las flechas ortogonales en el plano, y la tercera PC sería ortogonal al plano (apuntando hacia arriba o hacia abajo).





# Extracción de características

## Reducción de dimensión - PCA: componentes principales

- **Tip:** La dirección de los componentes principales no es estable. Sin embargo, generalmente seguirán en los mismos ejes.

Entonces, ¿cómo encontrar los componentes principales de un conjunto de entrenamiento?.

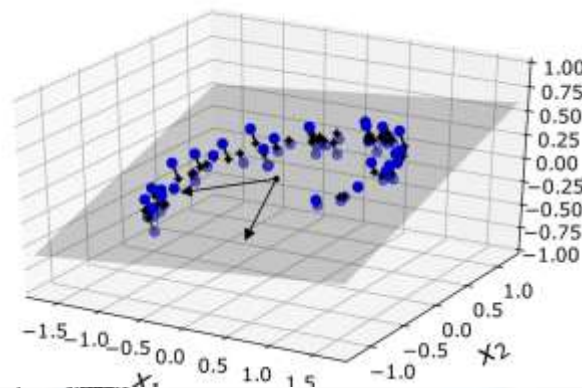
**R/** Existe una técnica estándar de factorización de matriz llamada **Singular Value Decomposition (SVD)** que puede descomponer la matriz del conjunto de entrenamiento **X** en la multiplicación matricial de tres matrices  **$U\Sigma V^T$** , donde **V** contiene todos los componentes principales que estamos buscando:

$$\mathbf{V} = \begin{pmatrix} | & | & \cdots & | \\ c_1 & c_2 & \cdots & c_n \\ | & | & \cdots & | \end{pmatrix}$$

# Extracción de características

## Reducción de dimensión - PCA: proyectando a $d$ -dimensiones

- Se reduce la dimensionalidad a  $d$ -dimensiones proyectándose en el hiperplano definido por los primeros  $d$  componentes principales. Garantizando que la proyección conservará la mayor variación posible.
- En la Figura, el conjunto de datos 3D se proyecta hacia abajo en el plano 2D definido por los dos primeros PCs, preservando una gran parte de la varianza del conjunto de datos.



# Extracción de características

## Reducción de dimensión - PCA: proyectando a $d$ -dimensiones

- Para proyectar el conjunto de entrenamiento en el hiperplano, se calcula la multiplicación matricial de la matriz  $X$  del conjunto de entrenamiento por la matriz  $W_d$ , que contiene los primeros  $d$  componentes principales, como se muestra en la siguiente ecuación:

$$X_{d-proj} = XW_d$$

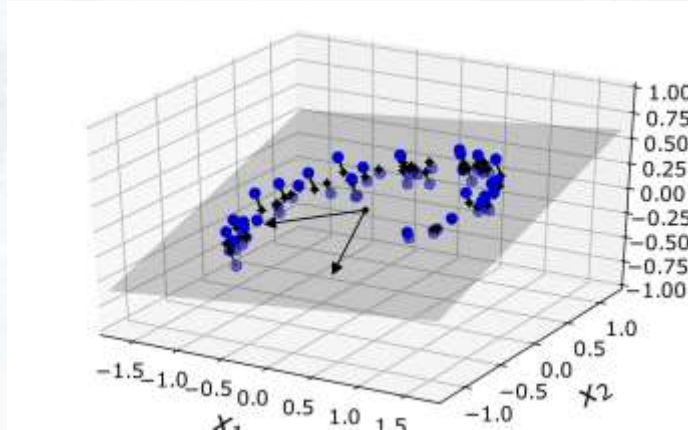
# Extracción de características

## Reducción de dimensión - PCA: explained variance ratio

La relación de varianza explicada de cada componente principal, está disponible a través de la variable *explained variance ratio*.

Esta indica la proporción de la varianza del conjunto de datos que se encuentra a lo largo del eje de cada componente principal.

### Ejemplo:



# Extracción de características

## Reducción de dimensión - PCA: Elegir el # correcto de componentes

Generalmente es preferible elegir el número de dimensiones (componentes) que se suman a una porción suficientemente grande de la varianza (por ejemplo, **95%**).



A menos, que el objetivo sea la visualización de datos; en ese caso, generalmente querrá reducir la dimensionalidad a 2 o 3.

# Extracción de características

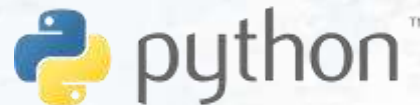
## Reducción de dimensión - PCA: para compresión

- Obviamente, después de la reducción de dimensionalidad, el conjunto de entrenamiento ocupa mucho menos espacio.
- También es posible descomprimir el conjunto de datos reducido de nuevo a las dimensiones originales aplicando la transformación inversa de la proyección PCA. Por supuesto, esto no le devolverá los datos originales, ya que la proyección perdió un poco de información.

$$\mathbf{X}_{recovered} = \mathbf{X}_{d-proj} \mathbf{W}_d^T$$

- La distancia cuadrática media entre los datos originales y los datos reconstruidos se denomina **error de reconstrucción**.

# Extracción de características



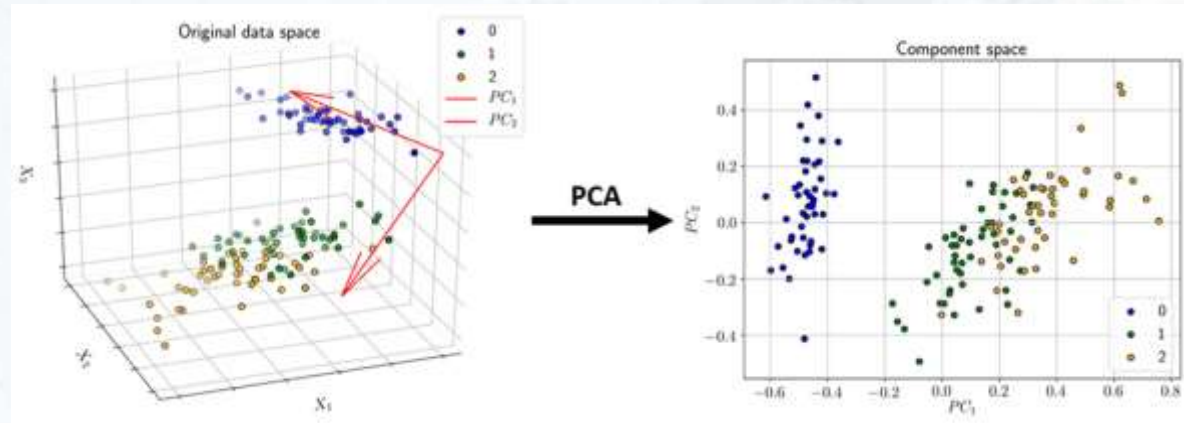
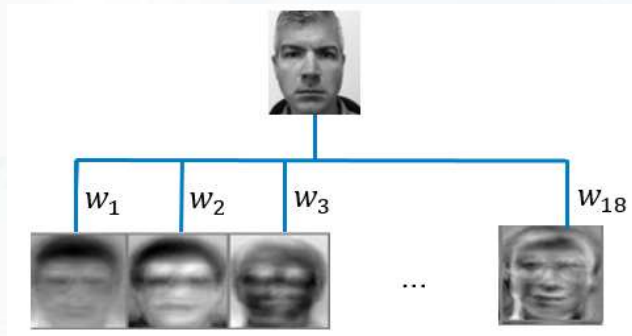
## Reducción de dimensión - PCA: variantes

- **Randomized PCA:** Si configuramos el hiperparámetro *svd\_solver* en “**randomized**”, Scikit-Learn usa un algoritmo estocástico llamado PCA aleatorio que encuentra rápidamente una aproximación de los primeros  $a$  componentes principales.
- **Incremental PCA:** Un problema con las implementaciones anteriores de PCA es que requieren que todo el conjunto de entrenamiento se ajuste a la memoria para que el algoritmo se ejecute. PCA incremental (**IPCA**) divide el conjunto de entrenamiento en mini-lotes y alimentar un algoritmo de IPCA un mini-lote a la vez.



# Extracción de características

## Reducción de dimensión - PCA: Resumen

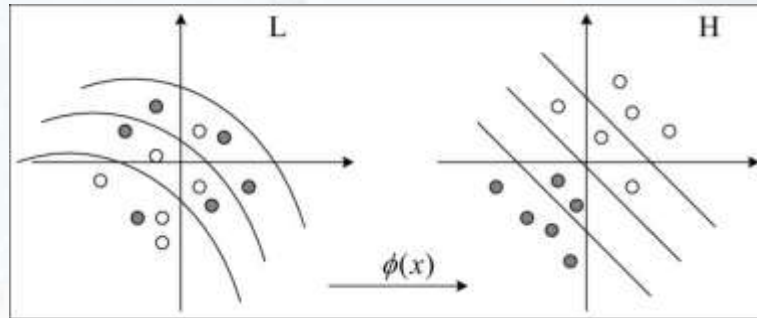


# Extracción de características

## Reducción de dimensión - PCA: Kernel PCA

El *kernel trick*, mapea implícitamente las instancias en un espacio de muy alta dimensión (espacio de características), permitiendo la clasificación y regresión no lineal con SVMs.

- **TIP:** Recuerde que un límite de decisión lineal en el espacio de características de alta dimensión corresponde a un límite de decisión no lineal complejo en el espacio original.



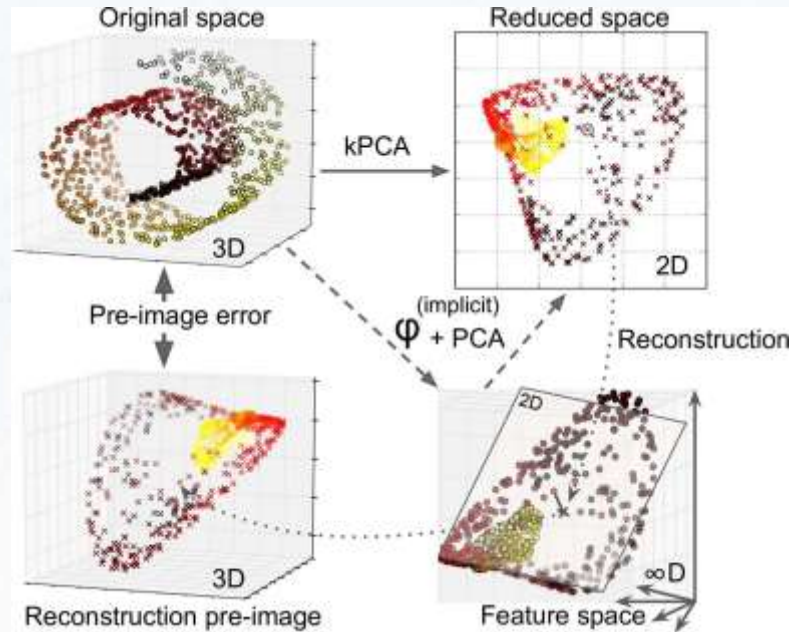
En PCA, se realizan **proyecciones no lineales complejas** para la reducción de la dimensionalidad.

Esto se llama **Kernel PCA (kPCA)**

# Extracción de características

## Reducción de dimensión - PCA: Kernel PCA

Otro enfoque, esta vez completamente sin supervisión, es seleccionar el kernel y los hiperparámetros que producen el error de reconstrucción más bajo. Sin embargo, **la reconstrucción no es tan fácil.**



# Extracción de características

## Reducción de dimensión - PCA: Kernel PCA

### Procedimiento:

1. Esto es mapear los datos en un espacio de características de dimensiones infinitas usando el feature map  $\phi$ , y luego proyectarlo usando **PCA lineal**.
1. Al invertir el paso lineal de PCA para una instancia dada en el espacio reducido, el punto reconstruido estaría en el espacio de características, no en el original.
1. Dado que el espacio de características es de **dimensión infinita**, no podemos calcular el punto reconstruido, ni el error de reconstrucción.
1. Es posible encontrar un punto en el espacio original que se mapearía cerca del punto reconstruido. Esto se llama reconstrucción pre-imagen.

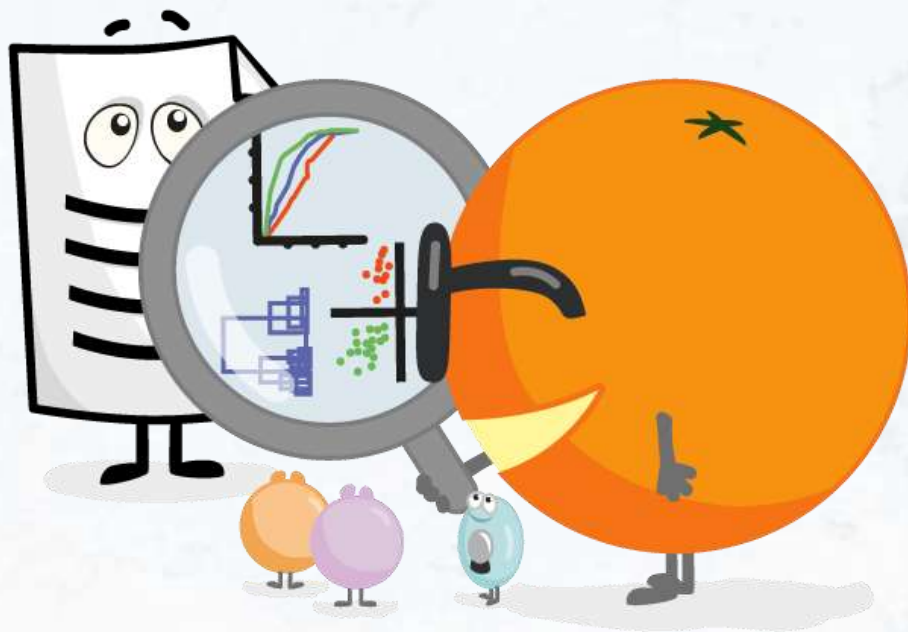
**Se puede seleccionar el kernel y los hiperparámetros que minimizan este error de reconstrucción pre-imagen.**

# Extracción de características

## Reducción de dimensión - OTRAS TÉCNICAS DE REDUCCIÓN DE DIMENSIÓN

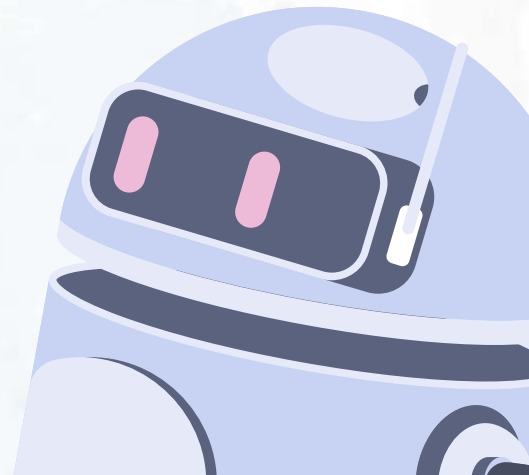
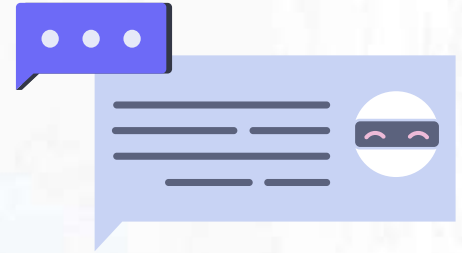
- **Locally Linear Embedding (LLE)** conserva la estructura local de los datos. Primero, encuentra pesos lineales que representan cada punto en función de sus vecinos cercanos. Luego, busca una representación de baja dimensión que conserve estas relaciones lineales. Finalmente, optimiza las coordenadas en la nueva dimensión para minimizar la diferencia entre las representaciones locales y globales.
- **El escalado multidimensional (MDS)** reduce la dimensionalidad al intentar preservar las distancias entre las instancias.
- **Isomap** crea un gráfico conectando cada instancia a sus vecinos más cercanos, luego reduce la dimensionalidad mientras intenta preservar las distancias geodésicas entre las instancias.
- **t-Distributed Stochastic Neighbor Embedding (t-SNE)** reduce la dimensionalidad al tratar de mantener instancias similares cercanas e instancias distintas lejos. Se utiliza principalmente para la visualización, en particular para visualizar grupos de instancias en espacios de alta dimensión.

# Ejercicios prácticos



# Gracias !

dfcollazosh@unal.edu.co



UNAL

Departamento de Eléctrica, Electrónica y Computación  
Facultad de Ingeniería y Arquitectura  
Sede Manizales



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA