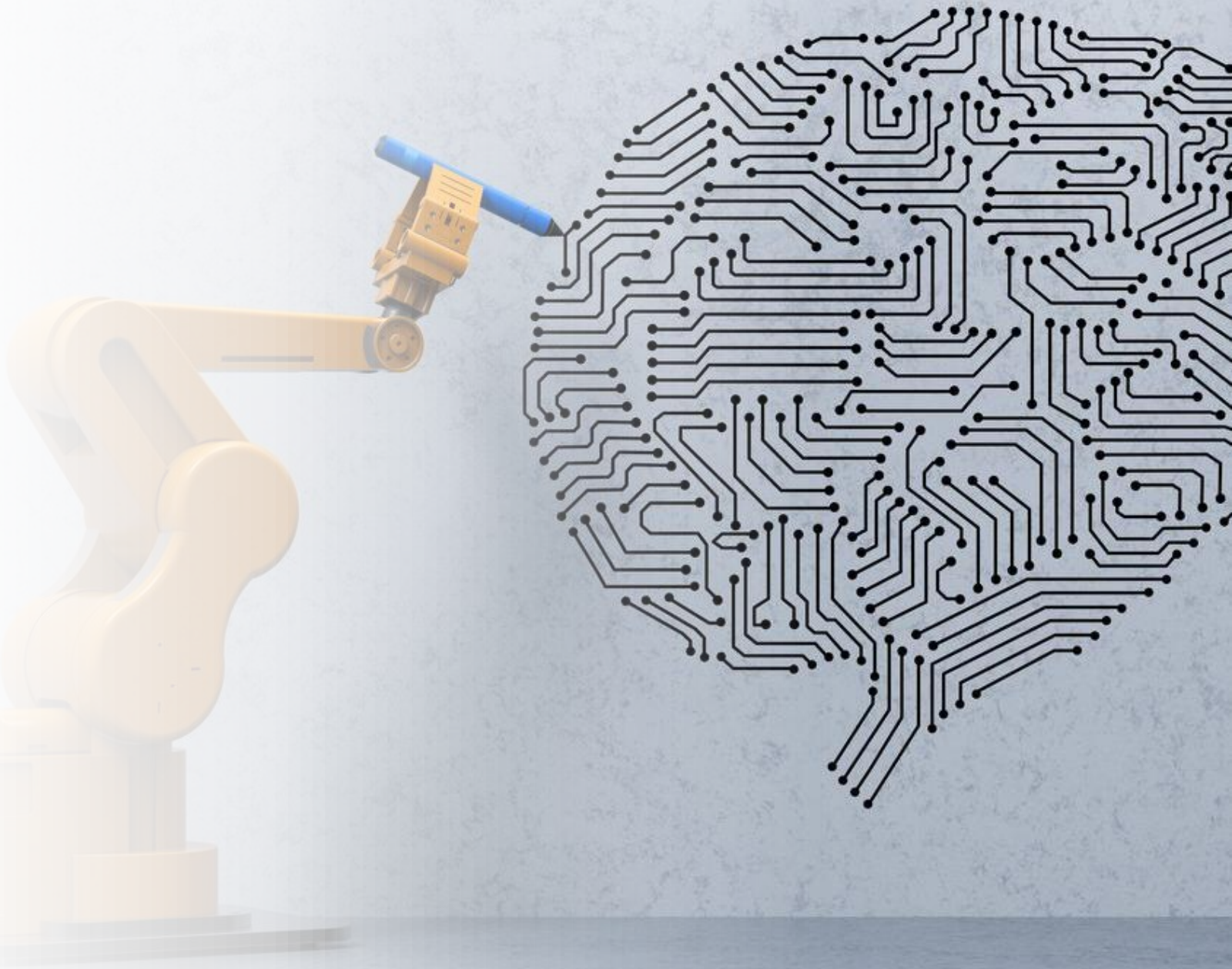
Reconocimiento de Patrones

Clase 5

Selección de características y reducción de dimensionalidad

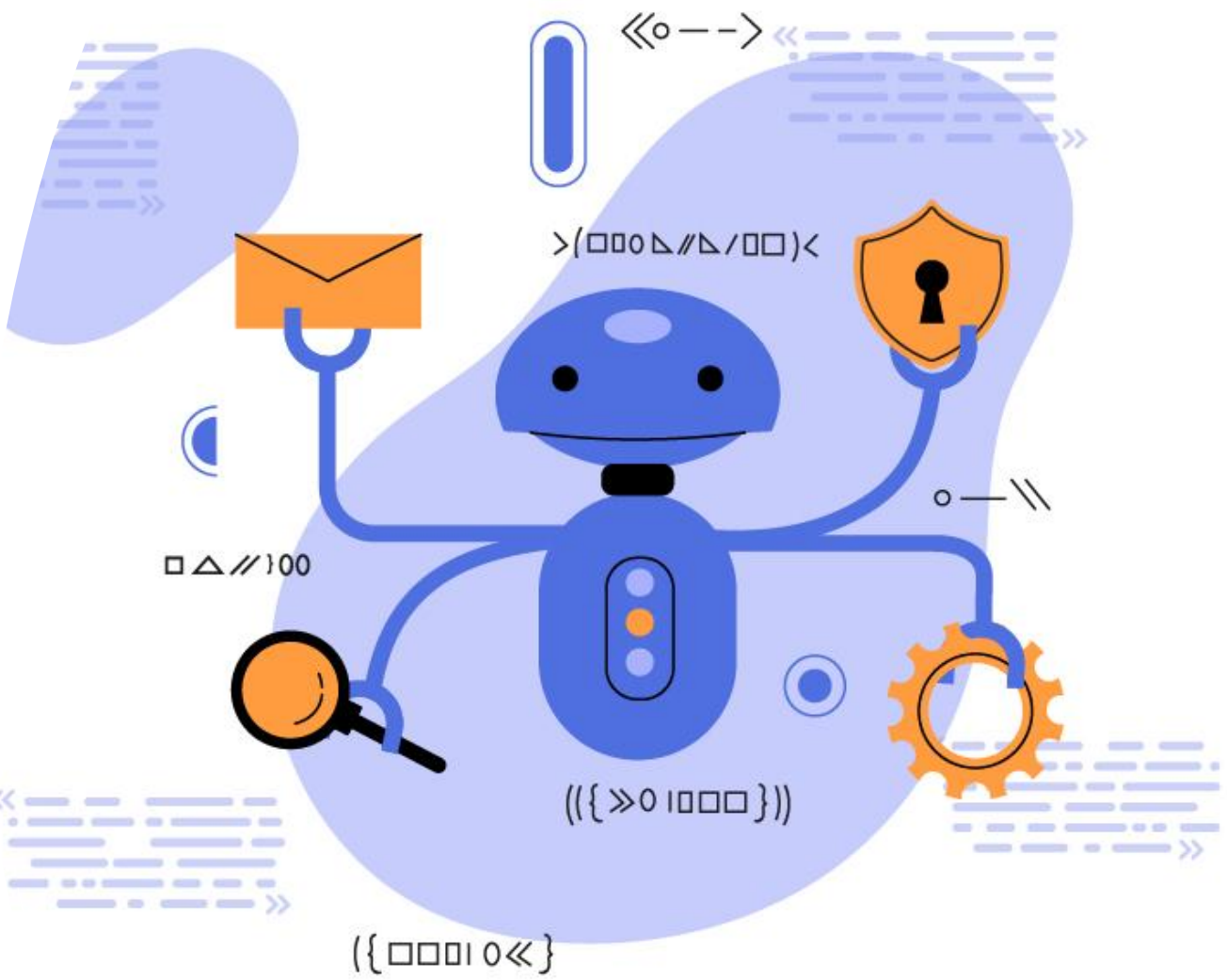


Antes de empezar

- ¿Dudas de la clase anterior?

Selección de características

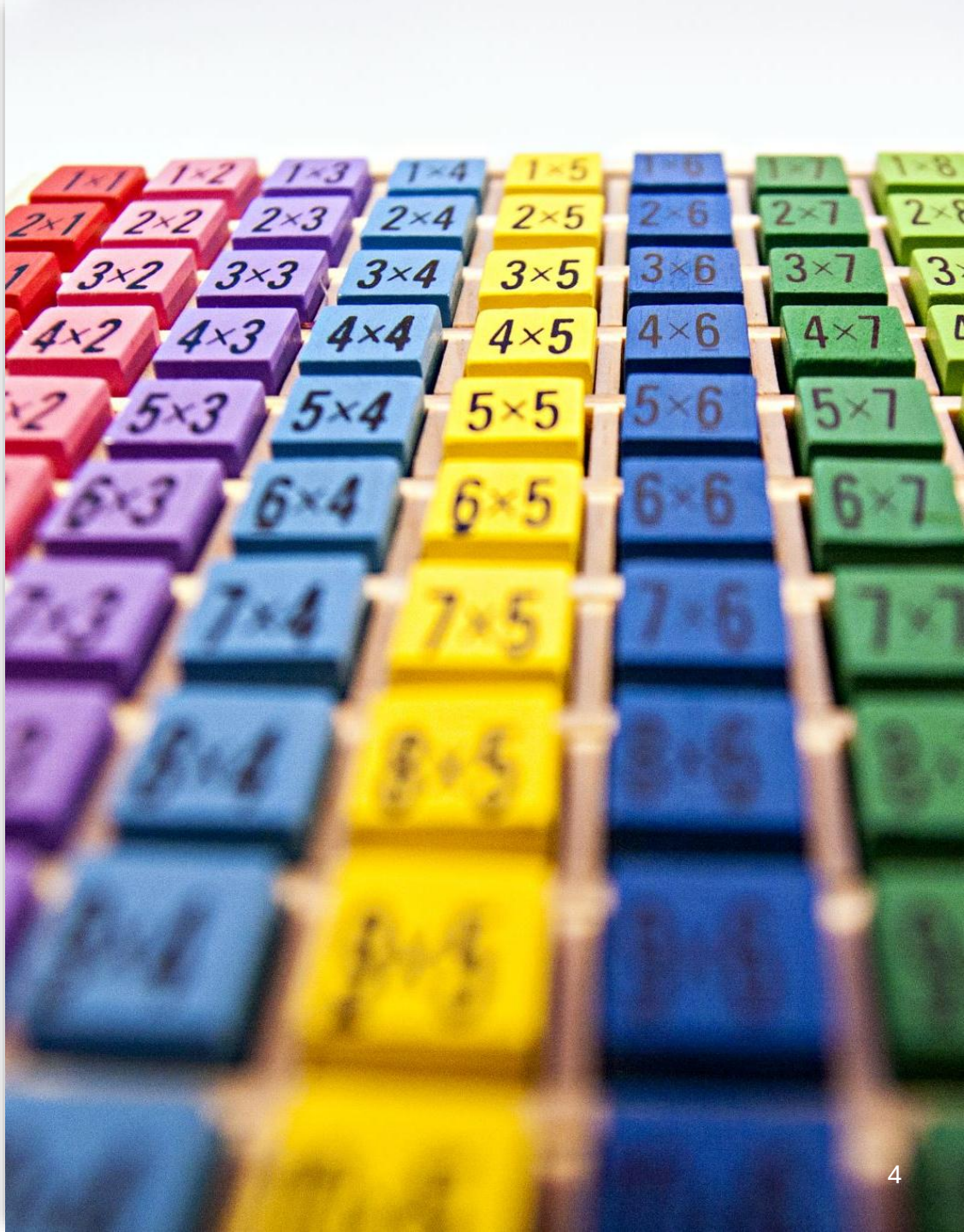
- Métodos de Filtrado
- Métodos Wrapper
- Métodos Embedded



Métodos de filtrado

Estos métodos evalúan las características de forma independiente al algoritmo de aprendizaje. Selección de características **antes de entrenar el modelo.**

1. Calculo de una medida de relevancia.
2. Ordenar características según su puntuación.
3. Seleccionar las k mejores características o aquellas que superan un cierto humbral.



- Correlación de Pearson

Medida estadística que cuantifica la intensidad y dirección de la relación lineal entre dos variables cuantitativas continuas.

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

- r>0: correlación positiva
- r=0: no existe una correlación lineal
- r<0: correlación negativa

- Chi-cuadrado (x^2)

Mide la dependencia entre variables categóricas. Compara las frecuencias observadas en cada categoría con las frecuencias esperadas bajo la hipótesis nula de no asociación.

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

Oi: valor observado

Ei: valor esperado

	Compra sí	Compra no	Total
Hombre	40	60	100
Mujer	70	30	100
Total	110	90	200

Esperado = (Total fila × Total columna) / Total general

	Compra sí	Compra no
Hombre	(100x110)/200=55	(100x90)/200=45
Mujer	(100x110)/200=55	(100x90)/200=45

chi	Compra sí	Compra no
Hombre	$(40 - 55)^2 / 55 = 225 / 55 = 4.09$	$(60 - 45)^2 / 45 = 225 / 45 = 5.00$
Mujer	$(70 - 55)^2 / 55 = 225 / 55 = 4.09$	$(30 - 45)^2 / 45 = 225 / 45 = 5.00$

- $\chi^2 = 4.09 + 5.00 + 4.09 + 5.00 = 18.18$

- gl = (número_filas - 1) × (número_columnas - 1)
- gl = (2 - 1) × (2 - 1) = 1

- De la distribución chi cuadrada con un alfa=0.05, χ^2_crítico = 3.841

Si χ^2_calculado > χ^2_crítico → Rechazar H_0 → Variables DEPENDIENTES → Característica RELEVANTE

Si χ^2_calculado ≤ χ^2_crítico → No rechazar H_0 → Variables INDEPENDIENTES → Característica IRRELEVANTE

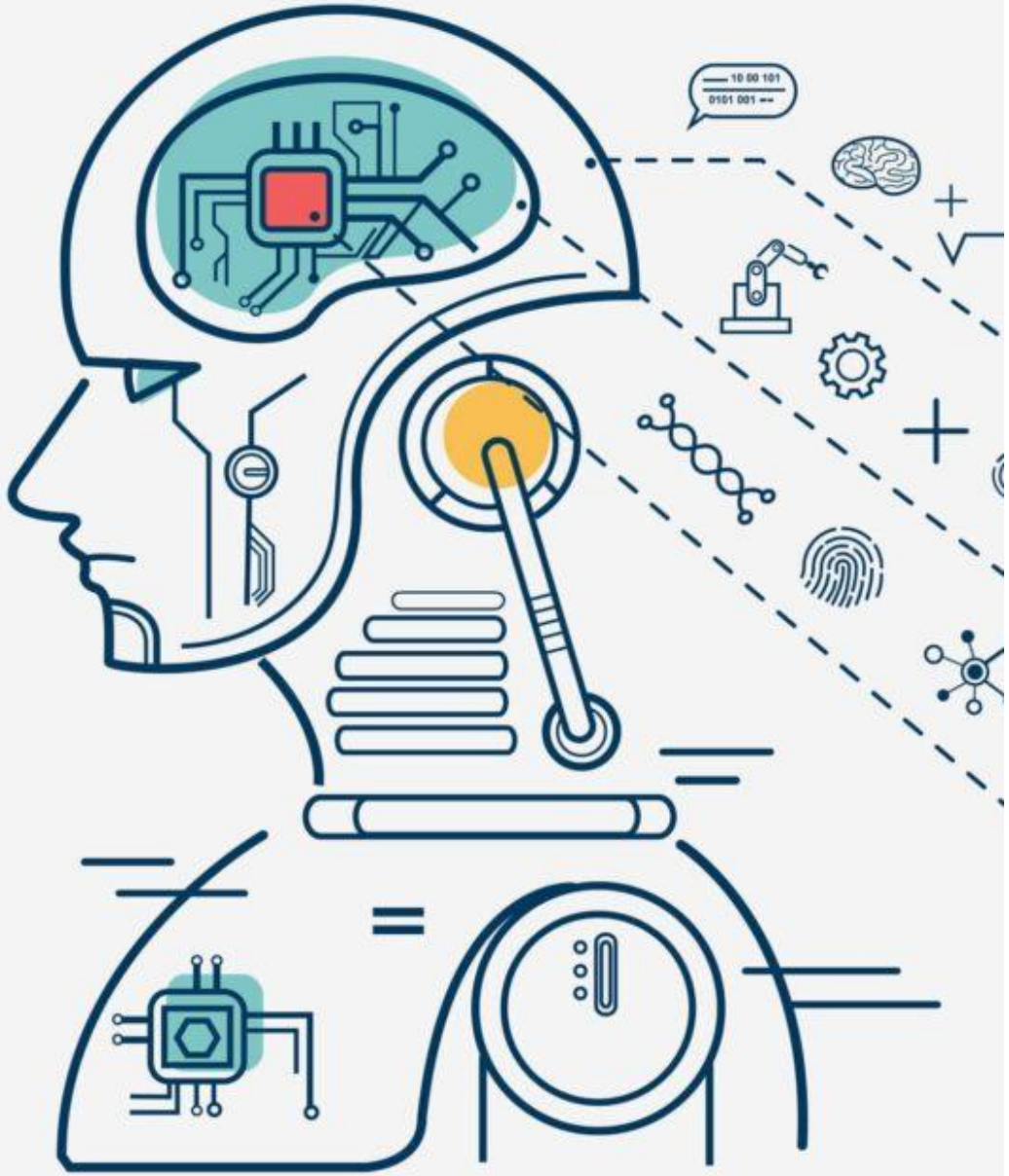
- ANOVA F-score

Mide si las medias de grupos difieren significativamente, se compara la varianza intergrupal con la varianza intragrupal. Un valor F alto indica que las medias de los grupos difieren significativamente, rechazando la hipótesis nula, lo que sugiere que la variable categórica tiene un impacto.

$$F = \frac{varianza\ entre\ grupos}{varianza\ dentro\ de\ los\ grupos}$$

¿Cuándo usar cada una?

Métrica	Tipo de característica	Tipo de objetivo
Correlación de Pearson	Continua	Continua
Chi-cuadrado	Categórica	Categórica
ANOVA F-Score	Continua	Categórica
Información Mutua	Cualquiera	Cualquiera



Métodos de Wrapper

- Evalúan subconjuntos de características **usando un modelo de aprendizaje.** La calidad del subconjunto se mide según el desempeño del clasificador.

1. Generación de subconjuntos.
2. Entrenamiento del modelo con cada subconjunto.
3. Evaluación del rendimiento.
4. Selección del mejor subconjunto.

Forward Selection

- Se empieza con un conjunto vacío de características y va añadiendo una a una hasta encontrar el conjunto optimo. Al llegar a un punto en el que el rendimiento no mejora se detiene el proceso.

- Tenemos 4 características: {A, B, C, D} para clasificar con un modelo

Subconjunto	Características	Accuracy	¿Mejor?
{A}	A	65%	
{B}	B	70%	✓
{C}	C	60%	
{D}	D	55%	

B tiene mayor accuracy: añadir B

S = {B}

Mejor accuracy = 70%

Subconjunto	Características	Accuracy	¿Mejor?
{B,A}	B+A	78%	+8% ✓
{B,C}	B+C	72%	+2%
{B,D}	B+D	71%	+1%

{B, A} tiene mejor accuracy (78%): Añadir A

S = {B, A}

Mejor accuracy = 78%

Subconjunto	Características	Accuracy	¿Mejor?
{B,A,C}	B+A+C	80%	+2% ✓
{B,A,D}	B+A+D	77%	-1%

{B, A, C} mejora (80%): Añadir C

S = {B, A, C}
Mejor accuracy = 80%

Subconjunto	Características	Accuracy	¿Mejor?
{B,A,C,D}	B+A+C+D	79%	-1% ✗

Añadir D empeora el rendimiento: Parar

Características seleccionadas: S = {B, A, C}
Accuracy final: 80%
Orden de selección: B → A → C

Backward Selection

Entrena el modelo con todas las características y elimina iterativamente la característica menos importante del subconjunto.

Tenemos 4 características: {A, B, C, D}

S = {A, B, C, D} (todas las características)

Rendimiento inicial = 82%

Subconjunto	Elimina	Restantes	Accuracy	Cambio
Sin A	A	{B, C, D}	83%	+1%✓
Sin B	B	{A, C, D}	75%	-7%
Sin C	C	{A, B, D}	80%	-2%
Sin D	D	{A, B, C}	81%	-1%

Eliminar A mejora el rendimiento.

S = {B, C, D}

Mejor accuracy = 83%

Subconjunto	Elimina	Restantes	Accuracy	Cambio
Sin B	B	{C, D}	70%	-13%
Sin C	C	{B, D}	80%	-2%
Sin D	D	{B, C}	84%	+1%✓

Eliminar D mejora el rendimiento.

S = {B, C}

Mejor accuracy = 84%

Subconjunto	Elimina	Restantes	Accuracy	Cambio
Sin B	B	{C}	60%	-24%
Sin C	C	{B}	70%	-14%

Eliminar cualquiera empeora significativamente: parar.

Características seleccionadas: S = {B, C}
Accuracy final: 84%
Características eliminadas: A, D
Orden de eliminación: A → D

Exhaustive Feature Selection

Prueba todas las combinaciones posibles de características y selecciona la que da el mejor rendimiento.

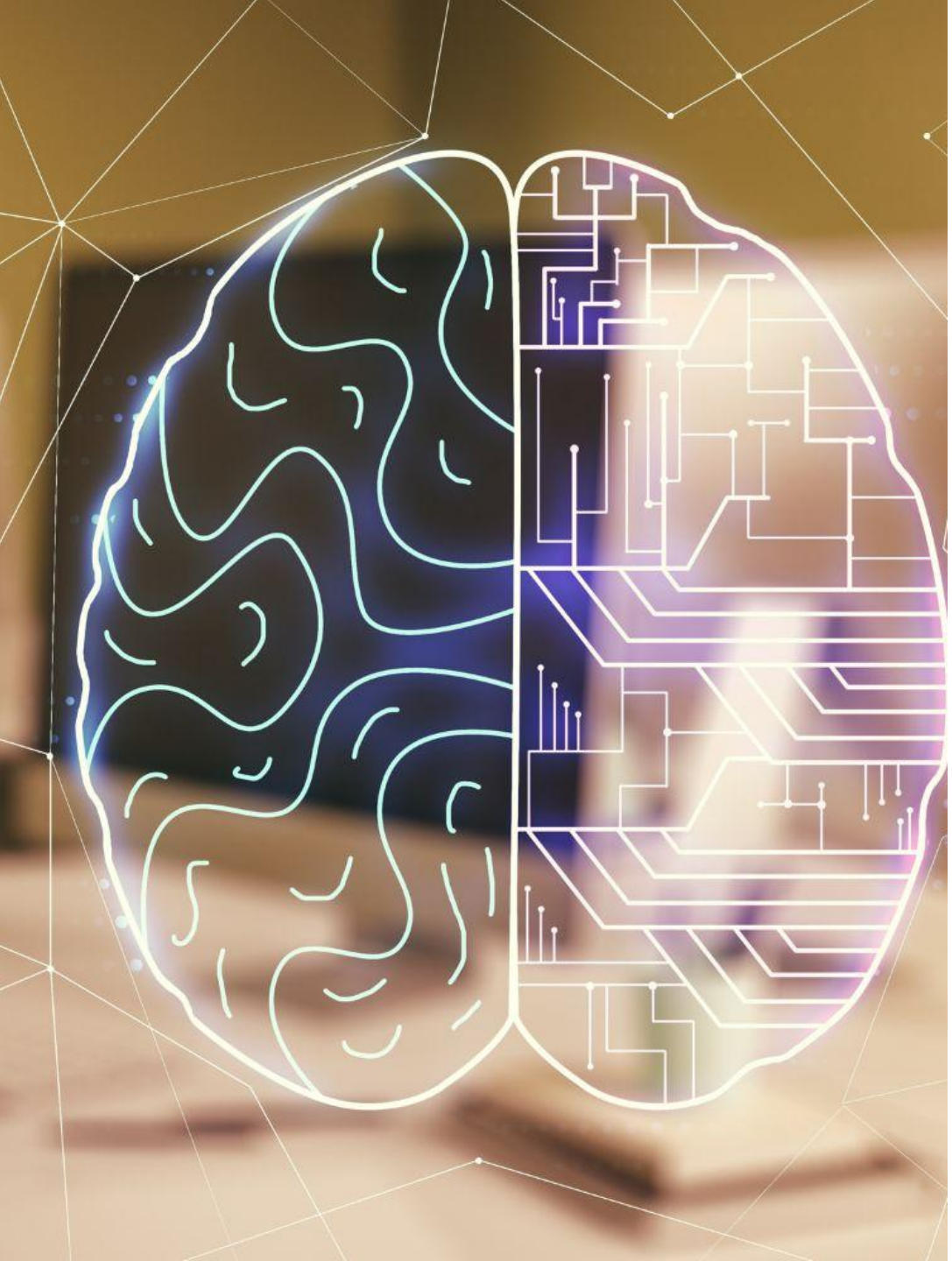
Características: {A, B, C, D}

Mejor subconjunto: {B, C}

Accuracy máxima: 84%

#	Tamaño	Características	Accuracy
1	1	A	65%
2	1	B	70%
3	1	C	60%
4	1	D	55%
5	2	A+B	78%
6	2	A+C	72%
7	2	A+D	68%
8	2	B+C	84%✓
9	2	B+D	71%
10	2	C+D	66%
11	3	A+B+C	80%
12	3	A+B+D	77%
13	3	A+C+D	73%
14	3	B+C+D	83%
15	4	A+B+C+D	82%

	Ventajas	**Desventajas**
Forward Selection	• Bueno cuando se espera que pocas características sean relevantes • Fácil de entender e implementar	• Una vez añadida una característica, no se puede quitar • Problema de características redundantes • No garantiza encontrar el mejor subconjunto global
Backward Selection	• Considera todas las características inicialmente • Bueno cuando se espera que muchas características sean relevantes • Puede capturar interacciones complejas mejor que forward	• Muy costoso computacionalmente. • No puede añadir características una vez eliminadas • Puede ser inviable con miles de características
Exhaustive Feature Selection	• Garantiza el optimo global • Considera todas las interacciones posibles	• Extremadamente costoso computacionalmente. • Inviable para más de ~15-20 características.



Métodos Embedded

La selección ocurre **durante el entrenamiento del modelo**. La selección está integrada en el algoritmo.

El algoritmo de aprendizaje tiene un mecanismo intrínseco para evaluar la importancia de características.

Durante el entrenamiento, asigna pesos o importancias a las características, de tal forma que las menos importantes reciben penalizaciones o son eliminadas

L1 Lasso (Least Absolute Shrinkage and Selection Operator)

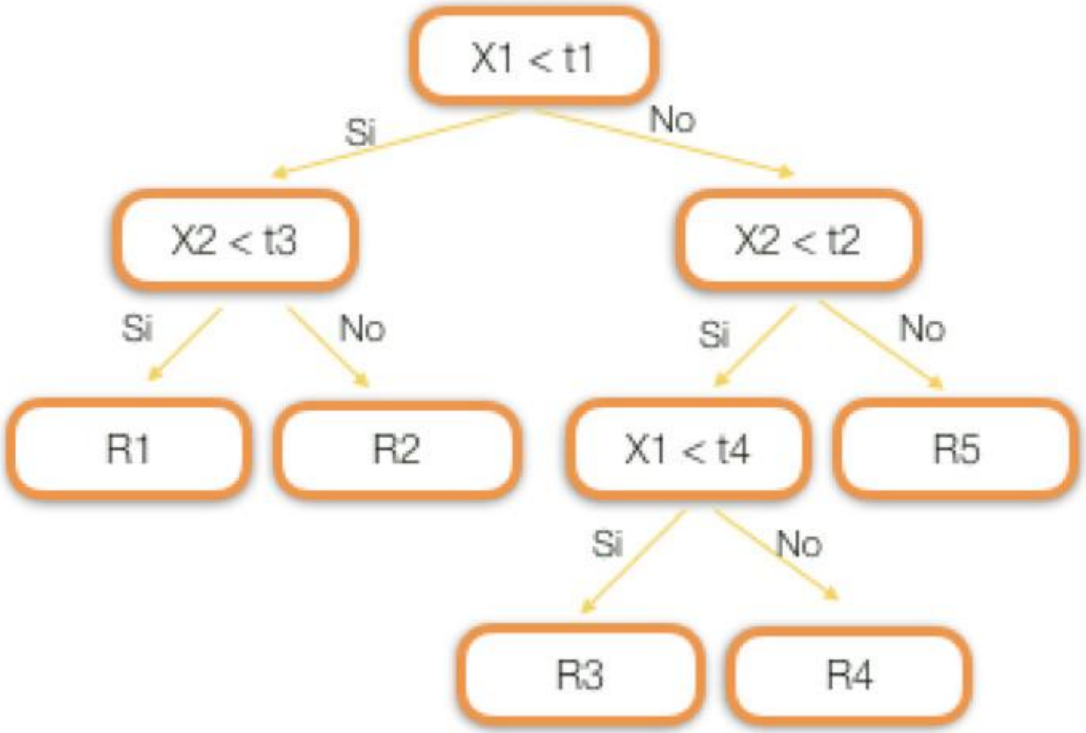
Se añade una penalización a los diferentes parámetros del modelo de aprendizaje, cuanto mayor sea la penalización más características se eliminan.

Su uso eficaz consistirá en equilibrar la penalización para eliminar suficientes características irrelevantes y mantener las importantes.

$$\min_w \left(-\sum y \log(p) + (1-y)\log(1-p) + \lambda \|w\|_1 \right)$$

$$\|w\|_1 = \sum_{i=1}^{p} |w_i|$$

Árbol de Decisión.



Un árbol de decisión es un modelo no lineal que particiona el espacio de características en regiones mediante reglas del tipo:

$$x_j \leq t$$

El árbol busca la variable y el umbral que maximicen la reducción de impureza.

1. Construcción del árbol

En cada nodo, el árbol evalúa TODAS las características

Elige la que MÁS reduce la impureza (Gini o Entropy)

2. Cálculo de importancia

Para cada característica:

Importancia = Σ (muestras_nodo / total) × Reducción_impureza

3. Normalización

Las importancias se normalizan para sumar 1.0

4. Selección

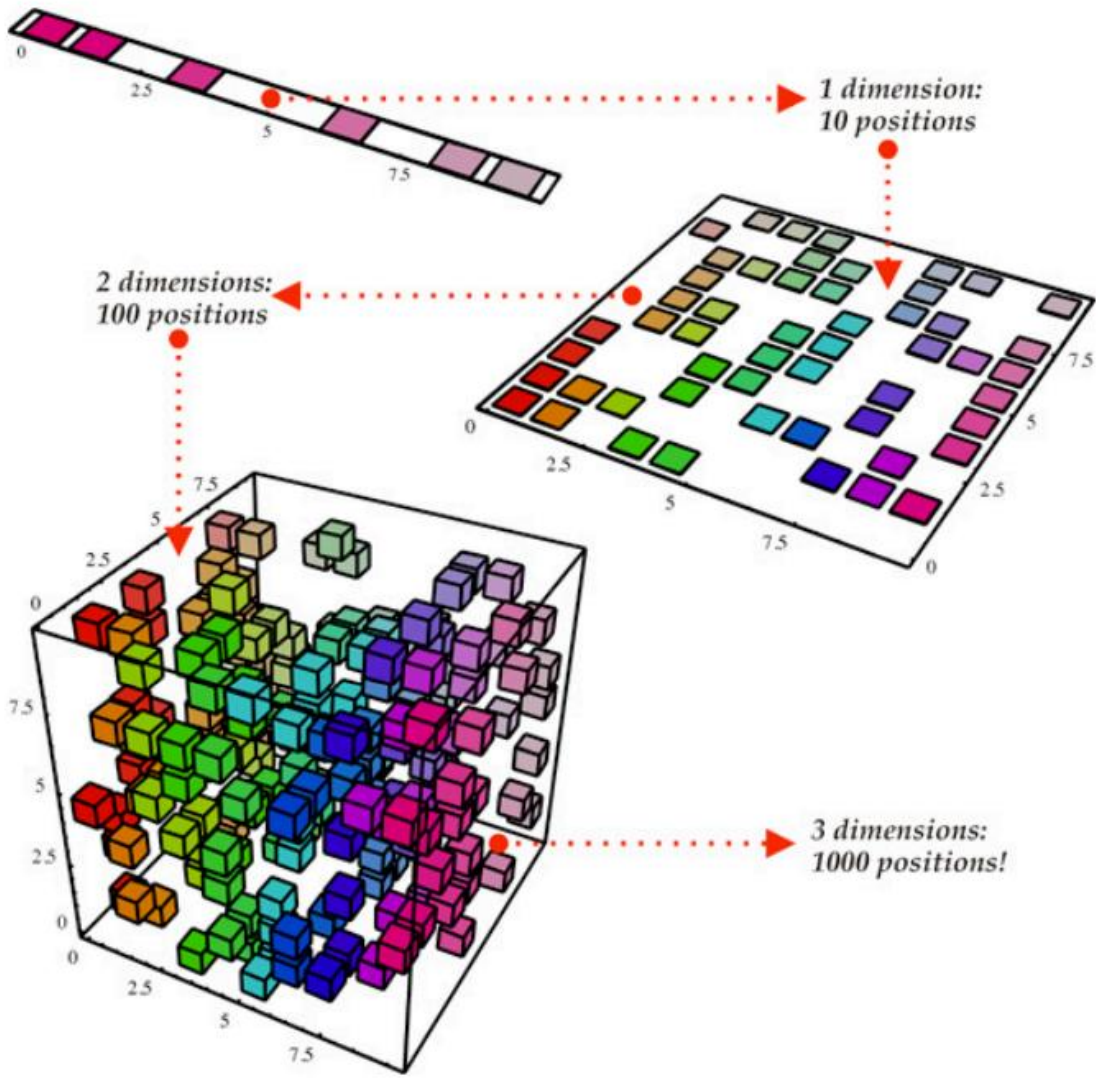
Umbral: Importancia > 0.5

Top-K: Las K mejores

Reducción de dimensionalidad

- Este tipo de técnicas buscan conservar características esenciales de conjuntos de datos complejos al reducir el numero de variables de previsión para una mayor generalización.

- Todos estos métodos transforman espacios de alta dimensión en espacios de bajas dimensiones mediante la extracción o la combinación de variables.



¿Por qué reducir la dimensionalidad?

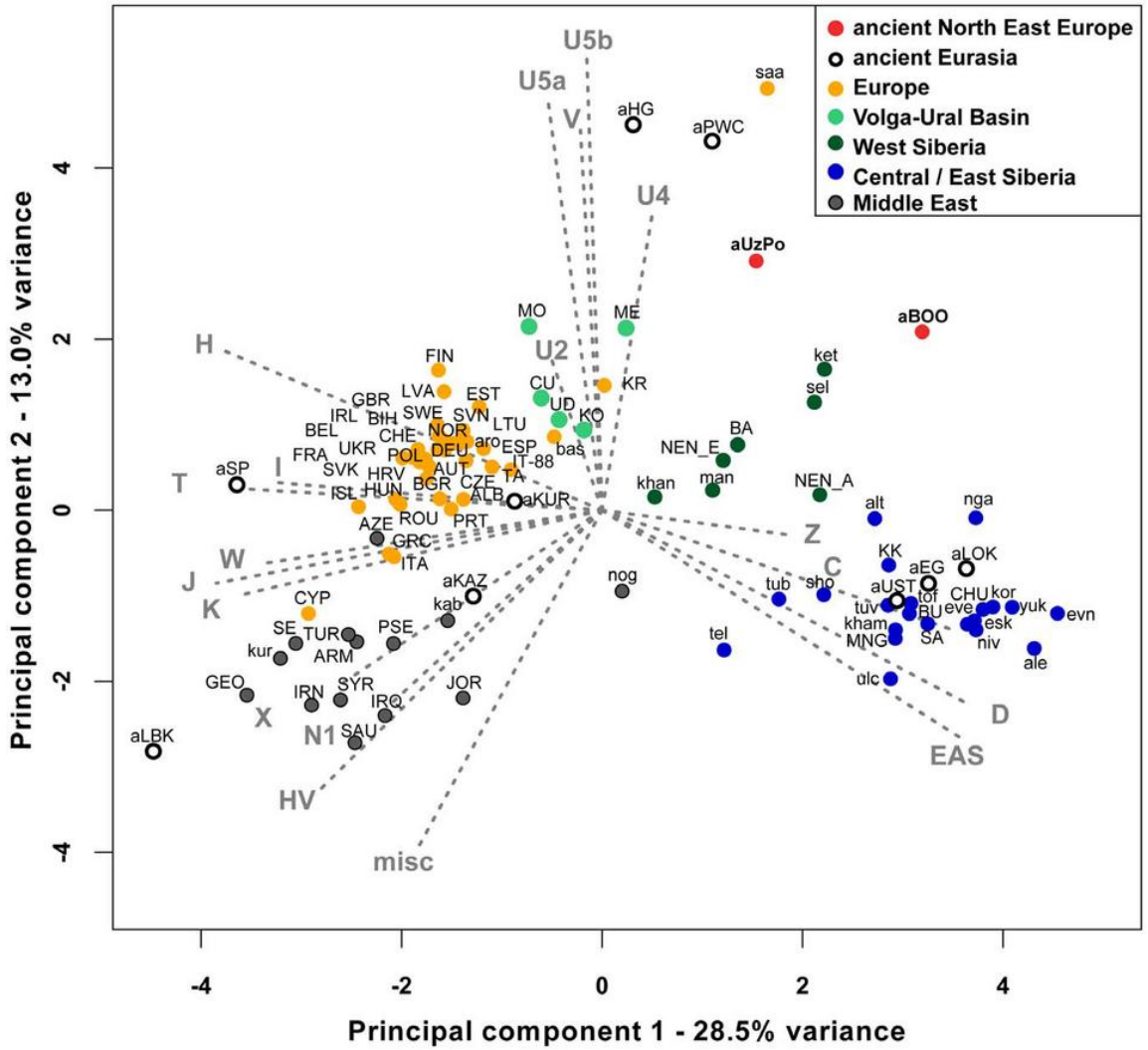
Conjuntos de alta dimensión plantean las siguientes preocupaciones:

- mayor tiempo de calculo
- mayor espacio de almacenamiento
- **degradación del rendimiento de los modelos**

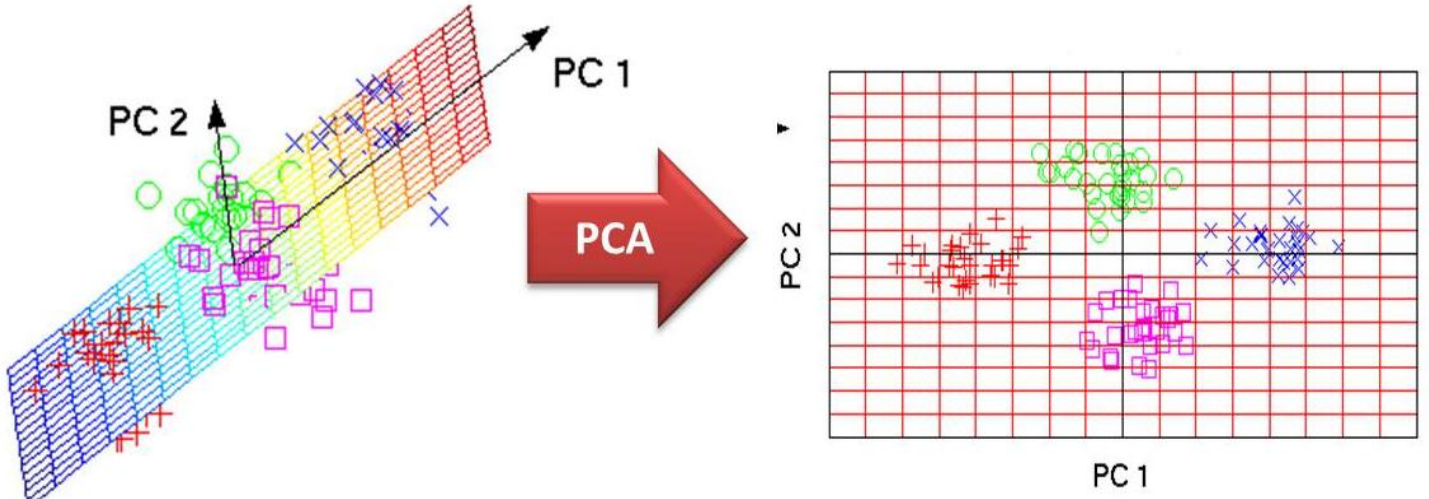
Modelos entrenados con conjuntos de datos de alta dimensionalidad con frecuencia generalizan mal.

Principal Component Analysis (PCA)

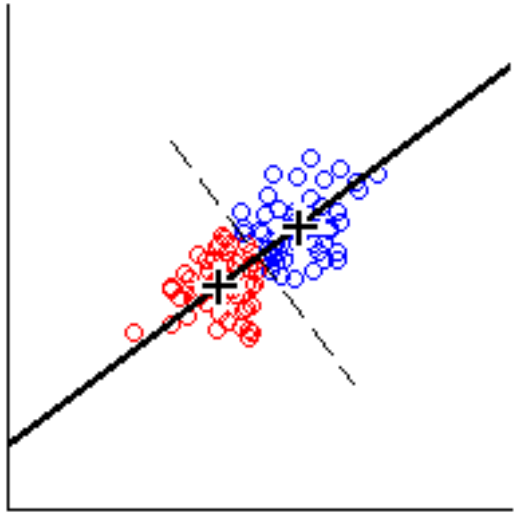
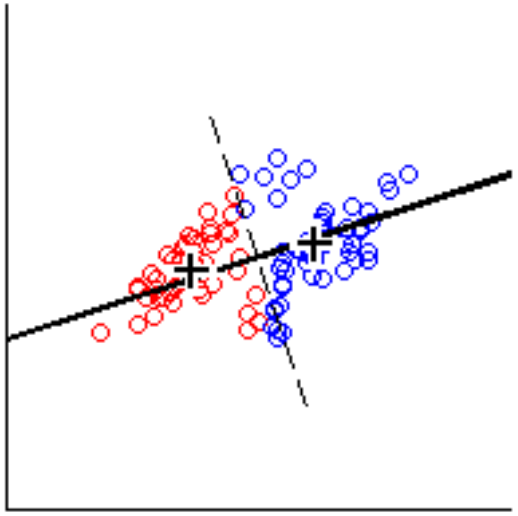
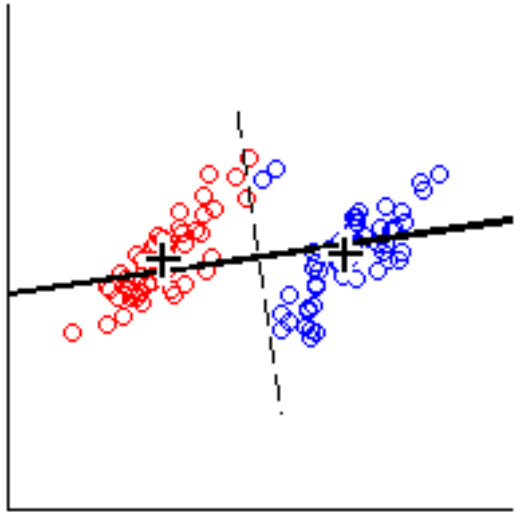
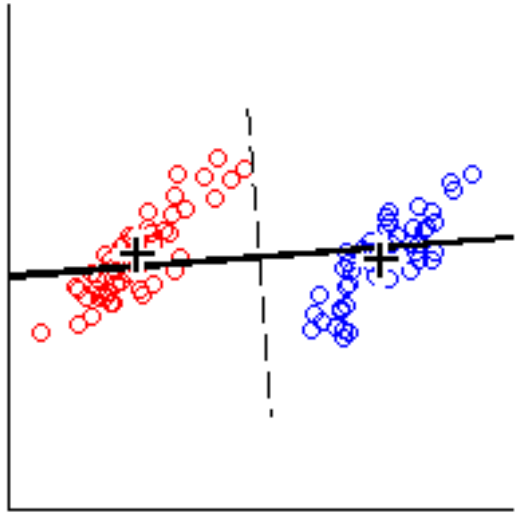
- Es una técnica que tiene como principal objetivo el transformar un conjunto de variables posiblemente correlacionadas en un nuevo conjunto de variables no correlacionadas llamadas **componentes principales**, ordenadas según la cantidad de varianza que explican.



- Extrae las características más informativas de grandes conjuntos de datos mientras conserva la información más relevante del conjunto de datos inicial.



- Proyectar datos con alta dimensionalidad a un espacio de características más pequeño se minimizan los problemas de multicolinealidad (dos o más variables altamente correlacionadas) y ajuste excesivo.

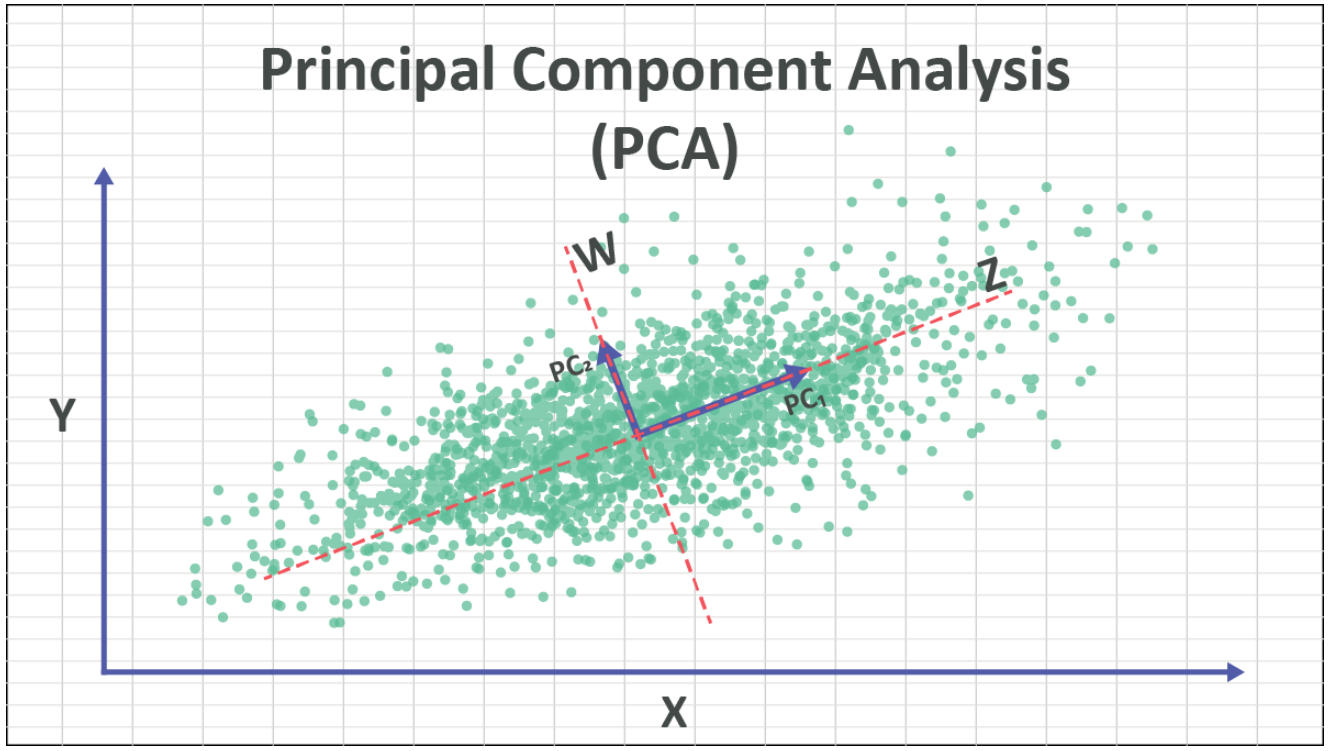





$$a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + a_3\mathbf{v}_3 + \cdots + a_n\mathbf{v}_n$$

- Los llamados componentes principales son combinaciones lineales de las variables originales que tienen la varianza máxima en comparación con otras combinaciones lineales, capturando tanta información del conjunto original como sea posible.

¿En que consiste PCA?

- Se busca encontrar los vectores y valores propios de la matriz de covarianza.
- Los vectores propios proporcionan **la dirección de la varianza** en el diagrama de dispersión y lo valores propios son **los coeficientes de los vectores propios** y **denotan la importancia de estos datos direccionales**.
- Un valor propio alto significa un vector propio critico.



¿Qué son los eigenvectores y eigenvalores?

- Es un vector que al ser transformado por una matriz no cambia de dirección, solo lo escala, el eigenvalor es ese factor de escala

$$A\,v = \lambda\,v$$

A: matriz

v: es el eigenvector

λ: es el eigenvalor

¿Qué son los componentes principales?

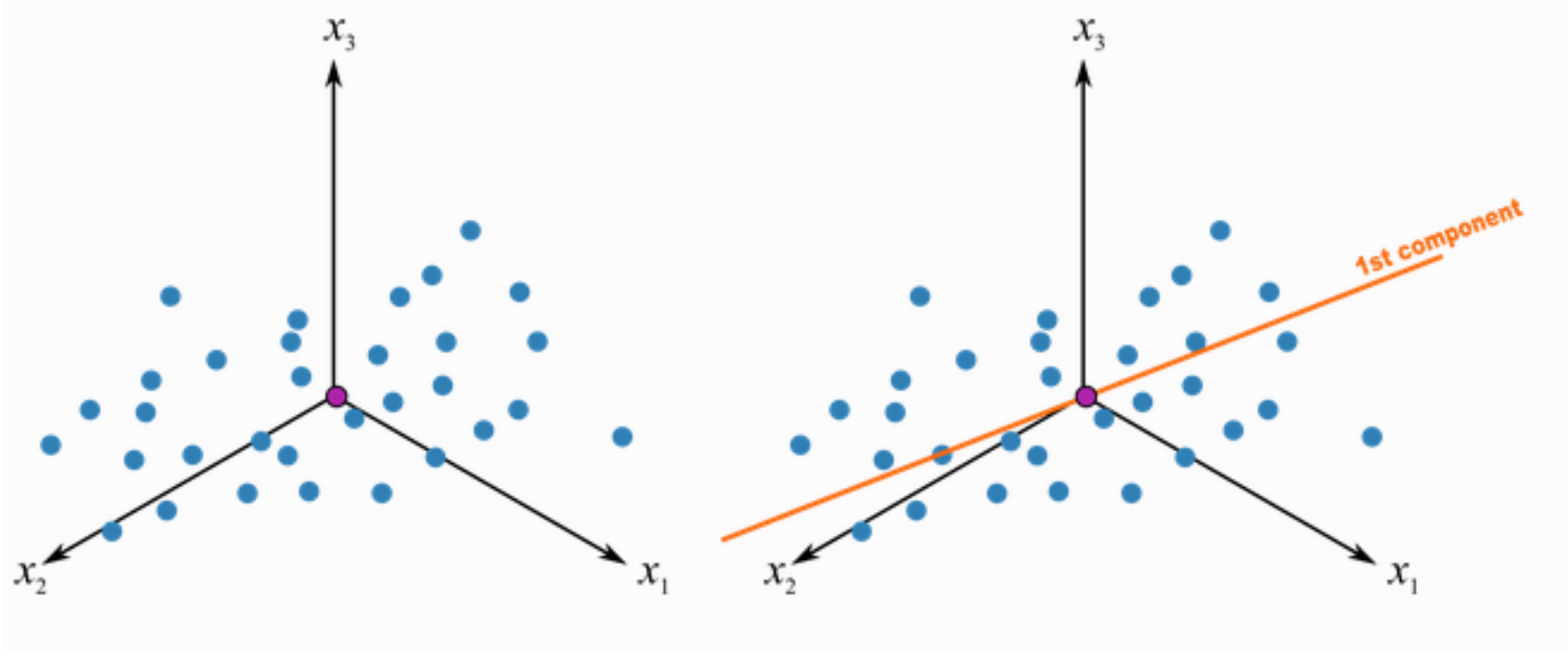
- Cada componente principal es:

$$z_i = w_i^T x$$

z_i : es la nueva variable transformada

w_i : es un vector propio

x : es el vector original



¿Cómo se calcula PCA?

- Supongamos que tenemos:

$$X \in \mathbb{R}^{Nxd}$$

N: número de muestras

d: número de características

1. Centrar los datos

Crucial ya que PCA es sensible a la escala.

$$X_c = X - \mu \qquad X_c = \frac{X - \mu}{\sigma}$$

2. Calcular la matriz de covarianza

Para un dataset X de n observaciones y p variables.

$$\Sigma = \frac{1}{n-1} X^T X$$

La covarianza mide cuanto varían dos variables juntas, si es alta, están correlacionadas.

3. Calcular valores y vectores propios

Se resuelve la ecuación:

$$\Sigma w = \lambda w$$

Obteniendo

λ: valores propios, cuanta varianza explica cada componente.

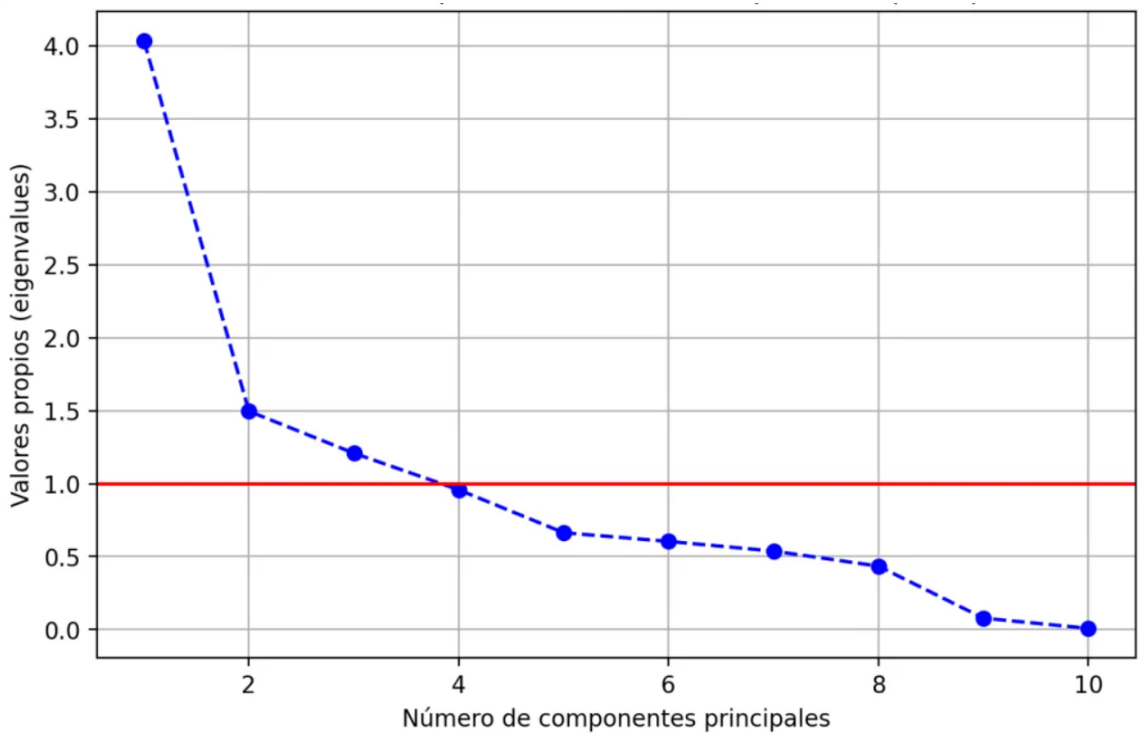
w: vectores propios, definen las direcciones principales.

4. Ordenar por varianza explicada

Se ordenan los pares (λ, w) de mayor a menor eigenvalor. El porcentaje de varianza explicada por cada componente es:

$$Varianza\ explicada = \frac{\lambda_i}{\Sigma_i \lambda_i}$$

Los primeros componentes explican mas varianza



5. Seleccionar k componentes

Se forma la matriz:

$$W_k = [w_1, \cdots, w_k]$$

6. Proyectar los datos

Se proyectan de acuerdo a:

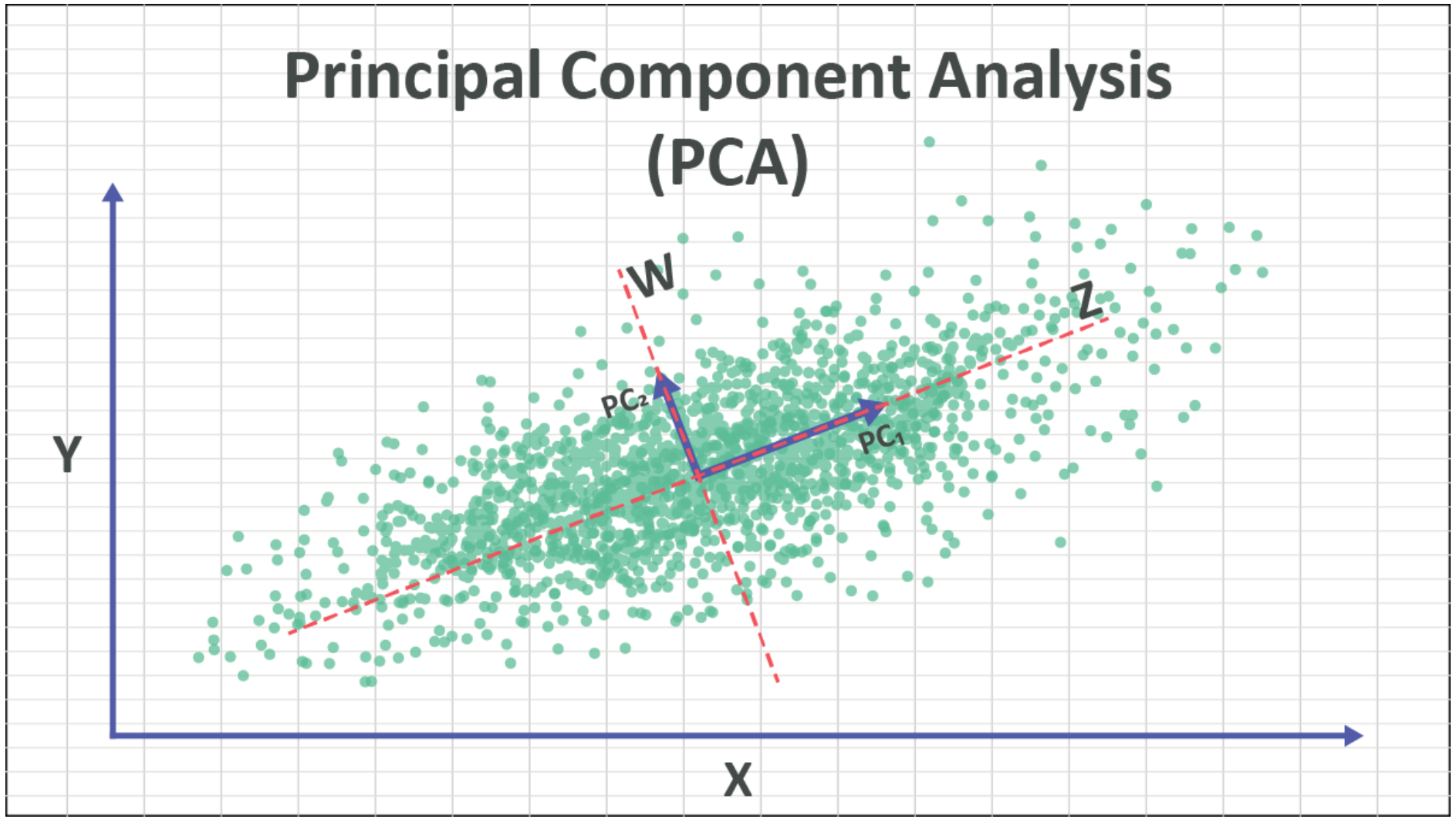
$$Z = X_c W_k, \qquad Z \in \mathbb{R}^{Nxk}, \qquad k < d$$

El resultado son los datos en un nuevo espacio reducido.

¿Qué significan estos componentes?

Primer componente principal: es la dirección en el espacio a lo largo de la cual los puntos de datos tienen la mayor varianza. Cuanto mayor sea la variabilidad capturada en el primer componente, mayor será la información retenida del conjunto de datos original.

Segundo componente principal: representa la siguiente varianza más alta en el conjunto de datos y no debe estar correlacionada con PC1, es decir, es ortogonal a esta.


Principal Component Analysis
(PCA)
W
Z
PC2
PC1
Y
X

Ejemplo.

Supongamos que se tiene la matriz de covarianza:

$$\Sigma = \begin{bmatrix} 4 & 2 \\ 2 & 3 \end{bmatrix}$$

Recordemos: $\Sigma v = \lambda v, \qquad (\Sigma - \lambda I)v = 0$

Para que no exista una solución trivial: $\det(\Sigma - \lambda I) = 0$

$$\Sigma - \lambda I = \begin{bmatrix} 4-\lambda & 2 \\ 2 & 3-\lambda \end{bmatrix}$$

$$\det(\Sigma - \lambda I) = (4-\lambda)(3-\lambda) - 4 = 12 - 4\lambda - 3\lambda + \lambda^2 - 4$$
$$= \lambda^2 - 7\lambda + 8$$

$$\lambda^2 - 7\lambda + 8 = 0$$
$$(\lambda - 1)(\lambda - 8) = 0$$
$$\lambda_1 = 8, \lambda_2 = 1$$

Se sustituye: $(\Sigma - \lambda I)v = 0$

$\lambda_1 = 8$

$\Sigma - 8I = \begin{bmatrix} -4 & 2 \\ 2 & -5 \end{bmatrix}$ $\quad\quad$ $\begin{bmatrix} -4 & 2 \\ 2 & -5 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = 0$

De la primera fila:

$-4w_1 + 2w_2 = 0$

$w_2 = 2w_1$ $\quad\quad$ $v_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

Normalizando: $\quad v_1 = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

$$\lambda_2 = 1$$

$$\Sigma - I = \begin{bmatrix} 3 & 2 \\ 2 & 2 \end{bmatrix} \qquad \begin{bmatrix} 3 & 2 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = 0$$

De la primera fila:

$$3w_1 + 2w_2 = 0$$

$$w_2 = -\frac{3}{2} w_1 \qquad v_2 = \begin{bmatrix} 2 \\ -3 \end{bmatrix}$$

Normalizando: $v_2 = \frac{1}{\sqrt{13}} \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

Ejemplo.

$$\Sigma = \begin{bmatrix} 4 & 1 \\ 2 & 3 \end{bmatrix}$$

$$\Sigma - \lambda I = \begin{bmatrix} 4-\lambda & 1 \\ 2 & 3-\lambda \end{bmatrix}$$

$$\det(\Sigma - \lambda I) = (4-\lambda)(3-\lambda) - 2 = \lambda^2 - 7\lambda + 10$$

$$\lambda_1 = 5, \lambda_2 = 2$$

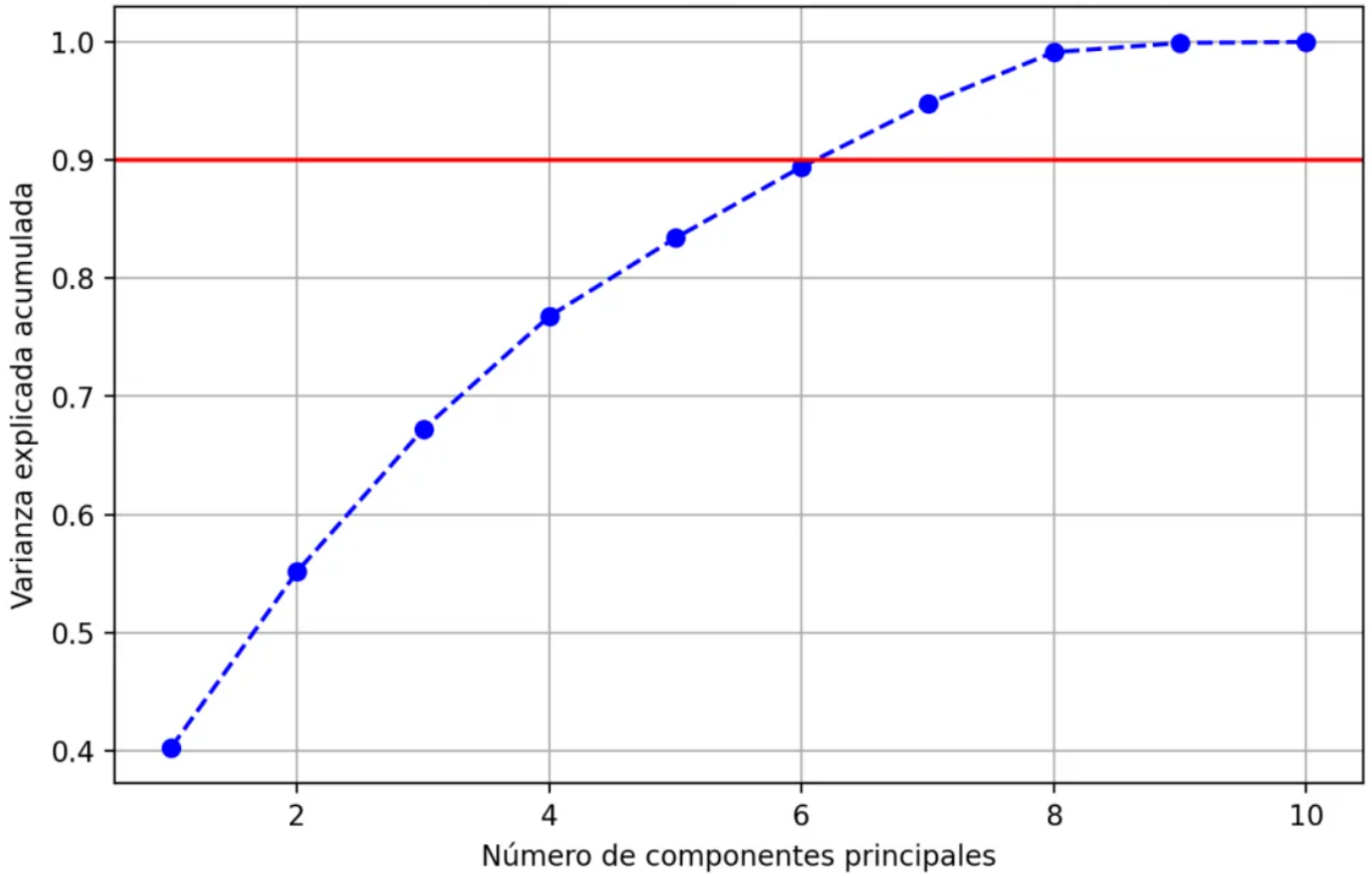
$$v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, v_2 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

¡¡Importante!!

- Vectores propios: nuevas bases
- Valores propios: cuanta información conserva cada componente
- La suma de todos los valores propios= varianza total

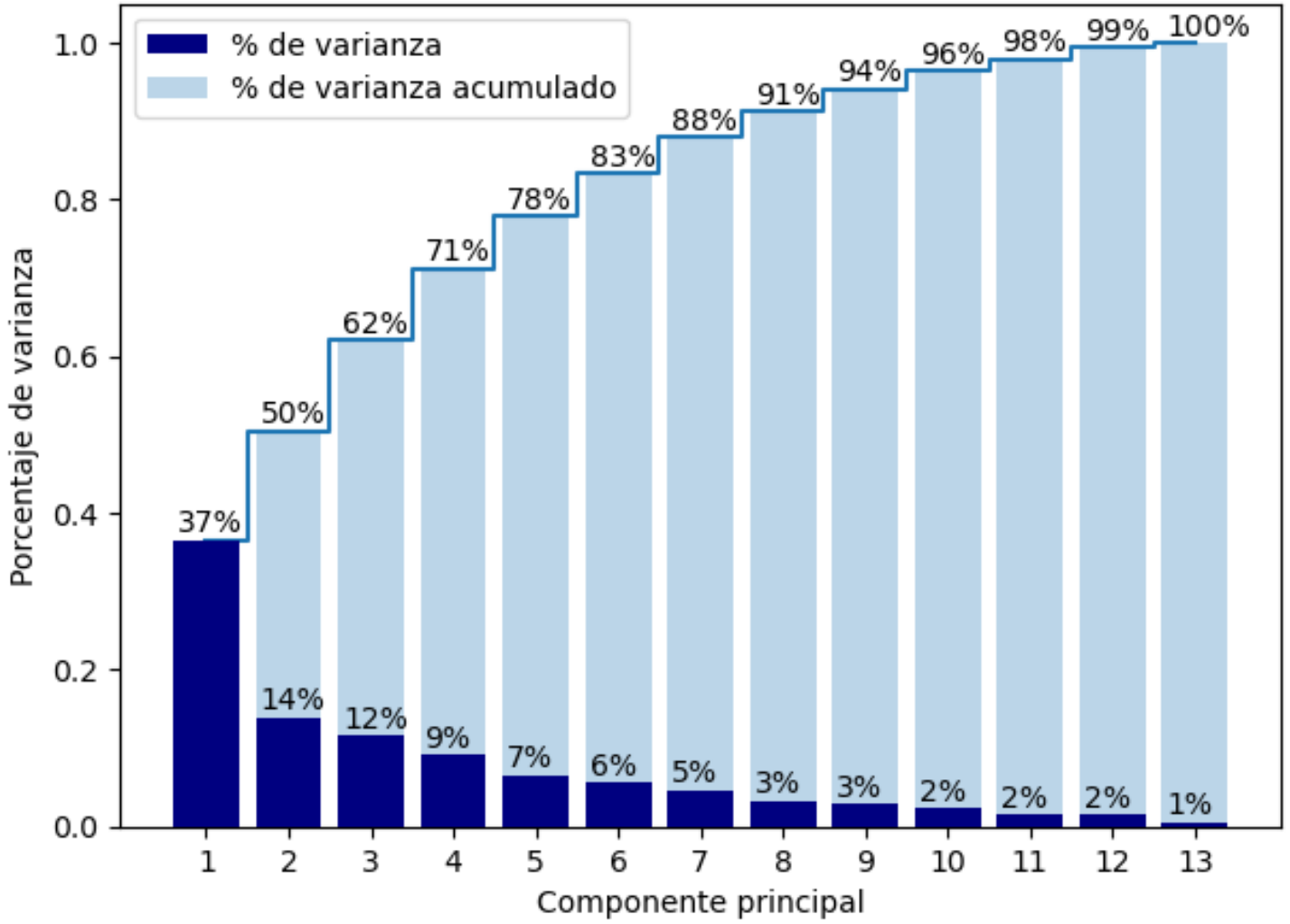
- Varianza explicada acumulada

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$$



¿Cuantas k escoger?

- Regla del codo
- Umbral de varianza
- Criterio de Kaiser



Singular Value Decomposition (SVD)

- Es una técnica que descompone cualquier matriz X (nxp) en tres matrices

$$X = U\Sigma V^T$$

$U(\text{nxn})$: vectores singulares izquierdos (observaciones en el nuevo espacio)

$\Sigma(nxp)$: matriz diagonal con valores singulares en orden descendente

$V^T(pxp)$: vectores singulares derechos (direcciones principales)

¿Cómo se utiliza SVD?

1. Centrar datos

$$X_c = X - \mu$$

2. Aplicar SVD directamente a Xc

$$X_c = U\Sigma V^T$$

3. Los componentes principales son las columnas de V

$$V = \begin{pmatrix} | & \cdots & | \\ v_1 & \cdots & v_p \\ | & \cdots & | \end{pmatrix}$$

- $Varianza\ explicada = \frac{\sigma_i^2}{\Sigma_i \sigma_i^2}$
- $\sigma_i = \sqrt{\lambda_i}$
- Para proyectar los datos: $Z = X_c V_k$

¿Por qué es mejor usar SVD?

- Más estabilidad numérica
- Funciona para matrices no cuadradas
- Evita formar explícitamente $X^T X$
- Más eficiente cuando m>>n o n>>m

Para la próxima clase...

- LDA