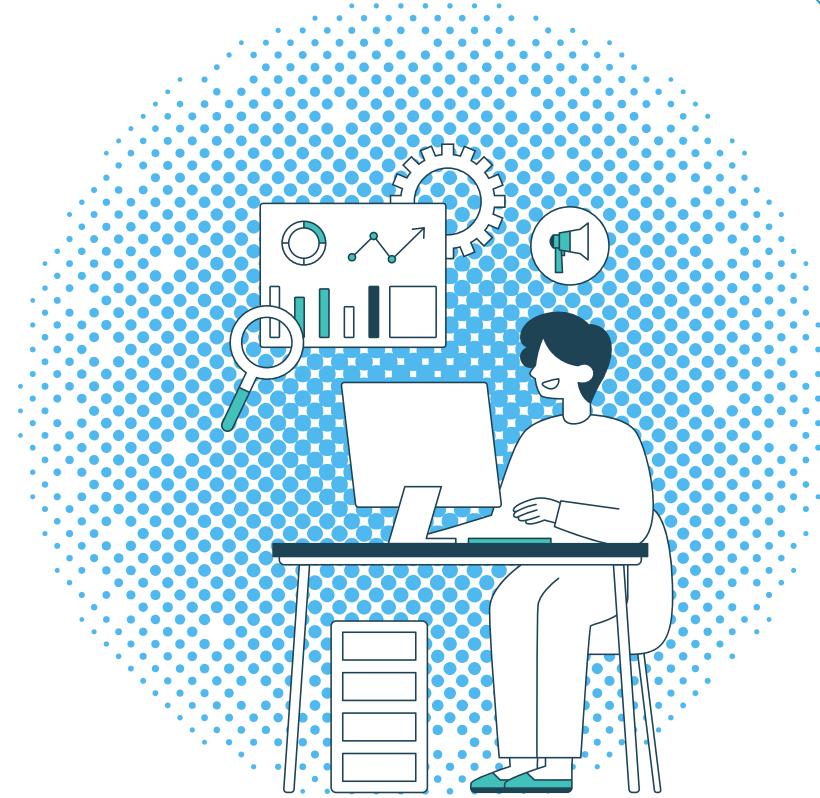


Análisis de datos

Carrera de Especialización en
Inteligencia Artificial



Programa de la materia

1 Introducción al análisis de datos

Flujo de trabajo típico. Aplicaciones. Configuración del entorno. Introducción a las principales bibliotecas.

2 Análisis exploratorio de datos

EDA, estadística descriptiva y visualización.

3 Preprocesamiento de datos y Feature Engineering

Tipos de variables. Codificación. Manejo de datos faltantes. Creación de nuevos features.

4 Pruebas estadísticas y validación de hipótesis

Pruebas de normalidad y de correlación. ANOVA y test de dependencia.

5 Reducción de dimensionalidad

Métodos para reducir la dimensionalidad y técnicas para selección de features.

6 Taller práctico

Análisis de datos completo de un dataset. Buenas prácticas. Discusión grupal.

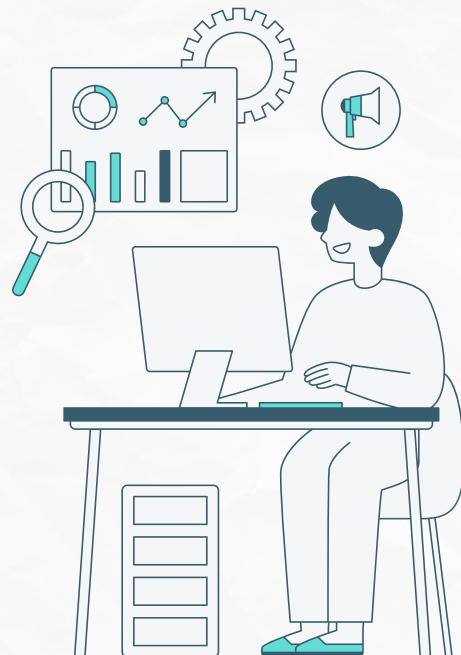
7 Presentación de trabajos finales

Exposición por grupos y devolución de los docentes.

8 Automatización del análisis de datos

Herramientas para automatizar el EDA y otras tareas del flujo de trabajo.

Creación de nuevos features



- Se trata de generar nuevas variables a partir de otras existentes.
- El objetivo es captar patrones o relaciones ocultas.

Ejemplo:

- $\text{IMC} = \text{peso} * (\text{altura})^2$
- Reemplazar las variables peso y altura por la métrica IMC, que está más relacionada con la salud que las originales.

Criterios para crear features

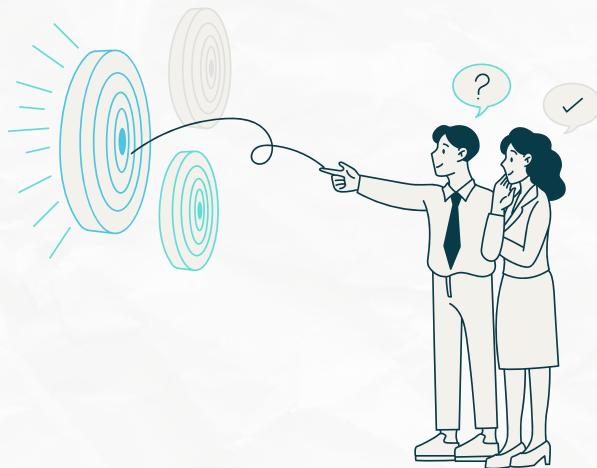


- Tienen que ser relevantes para el objetivo.
- Parten de variables que interactúan o dependen entre sí (correlacionadas).
- Tienen que complementar o superar lo que aportan las variables originales.

Ejemplos de casos de uso:

- Dataset con pocas filas.
- Patrones no lineales.
- El conocimiento del dominio lo pide (deuda/ingreso, nro. de ítems por compra, etc).

(Algunas) técnicas para crear features



- Matemáticas: A/B , A^*B , $A-B$, etc.
- Agregados: promedios, totales por grupo, sumas, etc.
- Temporales: extraer día, hora, mes, de un timestamp.
- Texto: TF-IDF (importancia de una palabra), generación de embeddings.

Dimensionalidad



- Se llama **dimensionalidad** al número de variables (features) en un dataset.
- Los datos con alta dimensionalidad pueden ser difíciles de analizar y visualizar.

Para qué reducir la dimensionalidad?

- Mejorar la eficiencia computacional.
- Reducir el riesgo de *overfitting*.
- Ayuda a la visualización (?)

La maldición de la dimensionalidad

A medida que aumentan las dimensiones, la complejidad de los datos crece exponencialmente, y:

- los puntos de datos se vuelven escasos
- las distancias entre puntos pierden significado
- el costo computacional aumenta exponencialmente.



Selección vs Extracción

- **Selección de *features*:** elegir un subconjunto de features importantes.
Ejemplo: remover features altamente correlacionadas.
- **Extracción de *features*:** transformar datos en un espacio de menos dimensiones.
Ejemplo: aplicar técnicas como PCA, t-SNE, UMAP.

Análisis de los componentes principales (PCA)

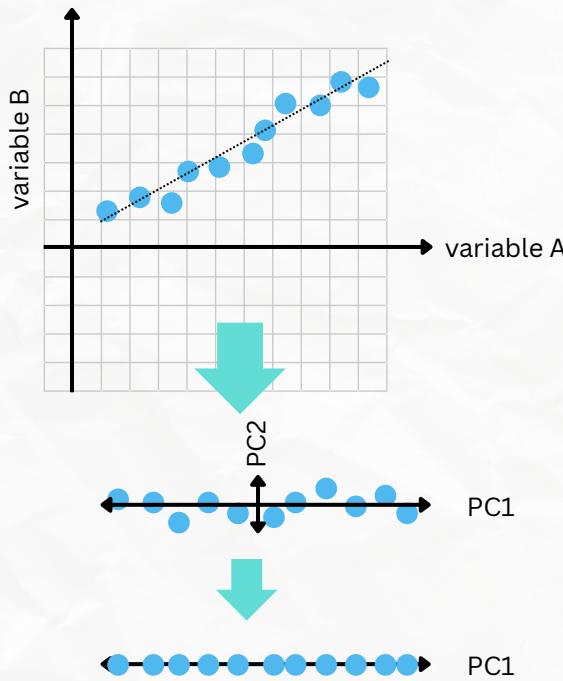
- Es un método lineal.
- Encuentra nuevos ejes de máxima varianza.
- Los componentes no están correlacionados.

Pasos:

- Estandarizar los datos.
- Computar la matriz de covarianza.
- Encontrar los autovalores y autovectores.
- Seleccionar los primeros k componentes.



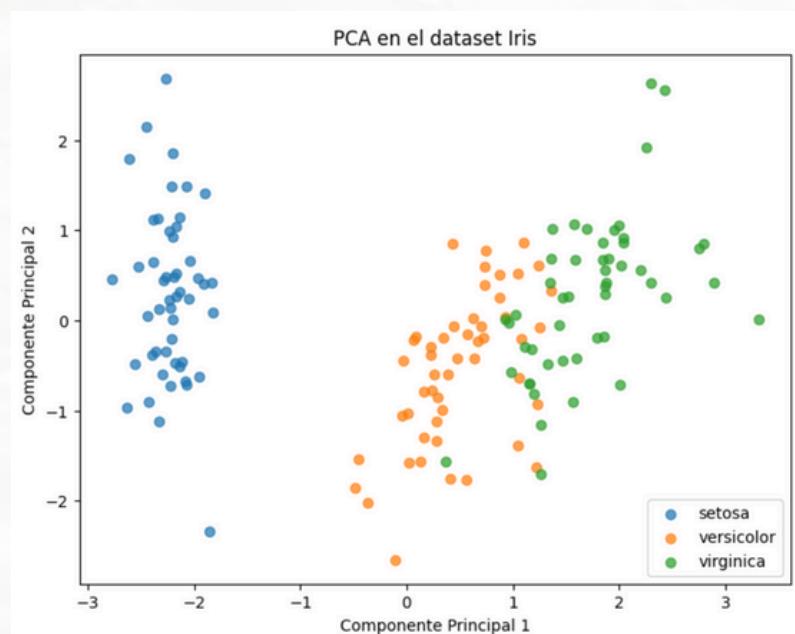
¿Por qué es importante la varianza?



- Recordemos: la varianza mide la dispersión de los datos alrededor de la media.
- En un dataset con muchos features, cada feature contribuye a la variación de los datos en mayor o menor medida.
- Retener la máxima varianza implica mantener las componentes que explican la porción más grande de esa variación.

Análisis de los componentes principales (PCA)

- Nro de Variables (antes de PCA) = 4
- Resultado PCA: solo 2 componentes concentran el 95% de la varianza



Ejemplo práctico

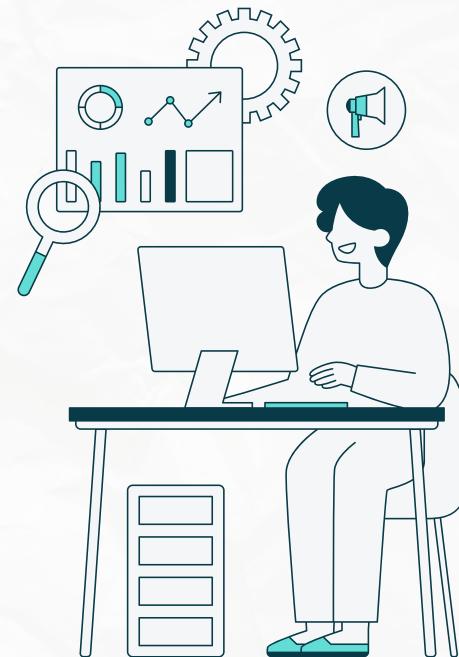


Análisis Discriminante Lineal (LDA)

- Es un método lineal.
- Es una técnica supervisada.
- Maximiza la separabilidad de las clases.
- Es útil para clasificación.

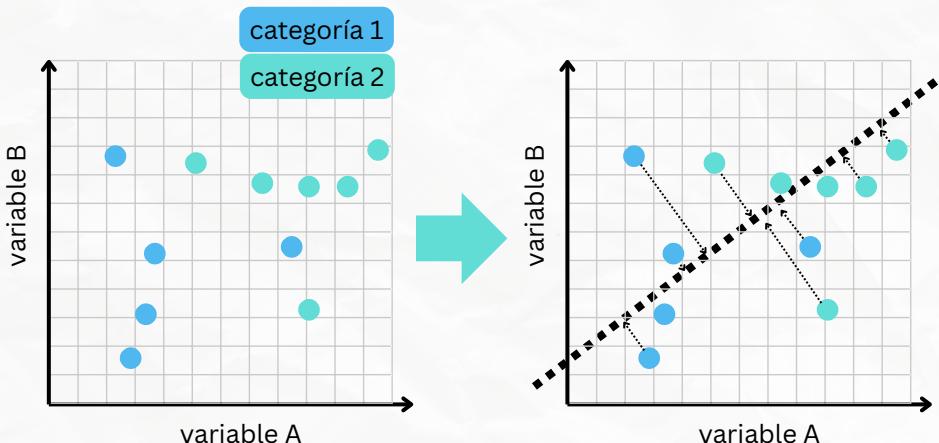
Comparación con PCA

- PCA captura la varianza.
- LDA se enfoca en la separación de clases.



¿Cómo funciona LDA?

- Busca maximizar distancia entre medias + minimizar la varianza (scatter) de cada categoría.



Crea un nuevo eje y proyecta los datos de manera que se maximice la separación de las categorías

En gral, el algoritmo devuelve:

- LD1, el eje que muestra la mayor variación entre categorías.
- LD2, el segundo mejor.
- Etc.

t-Distributed Stochastic Neighbor Embedding (t-SNE)

- Es un método no lineal.
- Bueno para visualización.
- Preserva la estructura local. No preserva bien la global.
- Computacionalmente caro.
- Es bueno para datasets pequeños.

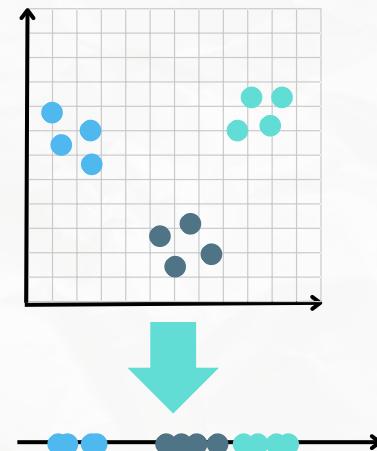
⚠️ No es para reducción de *features*, solo para visualización.

Funcionamiento de t-SNE

- Busca proyectar los datos en un espacio de dimensiones reducidas de manera tal que se mantengan los clústers de puntos del espacio de alta dimensión.

Pasos simplificados (Ej., 2D a 1D):

- 1) Determina la "similitud" entre cada punto y todos los demás.
- 2) Guarda todas las similitudes en una matriz.
- 3) Proyecta todos los puntos en forma aleatoria en un eje (1D).
- 4) Calcula matriz de similitudes.
- 5) Mueve los puntos poco a poco para que la matriz de 1D se parezca a la de 2D.



Uniform Manifold Approximation and Projection (UMAP)

- Es un método no lineal.
- Bueno para clustering.
- Preserva estructuras locales y globales.

Comparado con t-SNE:

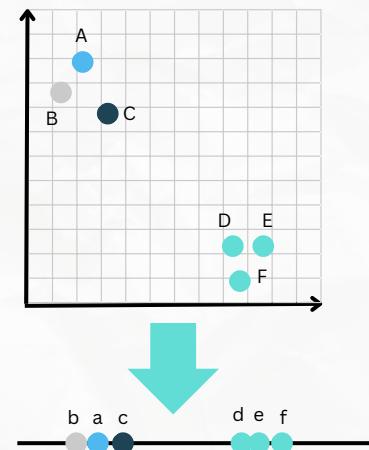
- Es más rápido.
- Escala bien para grandes datasets.
- Retiene más información global.

UMAP paso a paso

Crea una proyección en un espacio de dimensiones reducidas que preserva clústers presentes en los datos originales y la relación entre ellos.

Funcionamiento (Ej: 2D a 1D)

- 1) Calcula distancias entre todos los puntos.
- 2) Arma un mapa de vecinos y similitudes.
- 3) Inicializa el espacio de dimensión reducida (1D).
- 4) Calcula similitudes en el espacio reducido.
- 5) Ajusta una función para minimizar la diferencia de similitudes entre los espacios 2D y 1D.



¿Qué significa retener “información global o local”?

Mantener estructura global: conservar la relación general entre los puntos en el conjunto de datos, como por ejemplo la varianza.

Mantener estructura local: preservar las relaciones entre puntos cercanos.

Ej: estamos monitoreando datos de sensores de una máquina y queremos captar anomalías sutiles.

¿Cuándo usar cada una?

Técnica	Cuándo usarla	Consideraciones	Escenarios de uso comunes
PCA	Cuando se necesita un método simple y lineal que retenga la máxima varianza.	<ul style="list-style-type: none">Asume relaciones lineales.	<ul style="list-style-type: none">Preprocesamiento antes de ML.Reducir ruido.Comprimir datos.
LDA	Tenemos ls datos etiquetados y queremos reducir dimensiones para facilitar la separación entre clases.	<ul style="list-style-type: none">Es para métodos supervisados.Reduce los datos un máx de $C-1$ ($C =$ núm clases).	<ul style="list-style-type: none">Tareas de clasificación.Extracción de features para datos etiquetados.
t-SNE	Queremos visualizar datos en muchas dimensiones	<ul style="list-style-type: none">Costoso computacionalmente.No supervisado.	<ul style="list-style-type: none">EDAVisualizar embeddings, vectores de palabras, datos de genómica.
UMAP	Se necesita reducir dim. o visualizar pero también preservar la estructura global y local.	<ul style="list-style-type: none">Bueno para relaciones no lineales y datos complejos.Más rápido que t-SNE.	<ul style="list-style-type: none">Visualización (alternativa a t-SNE).Preprocesamiento antes de ML para datos complejos.

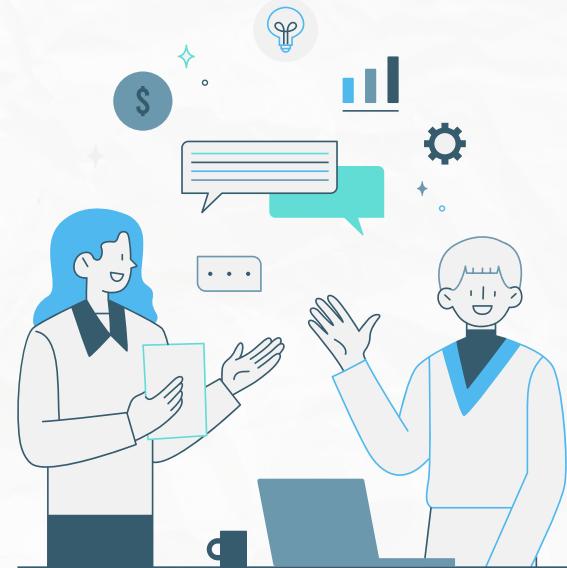
Siempre hay un “pero”...



- Al reducir dimensiones se pierde información.
- Técnicas como PCA transforman los features originales y después cuesta interpretarlos.
- La efectividad de la técnica depende del dataset y del problema en cuestión.

Aplicaciones

- Compresión de imágenes en reconocimiento facial.
- Segmentación de clientes (clustering).
- Análisis de datos genómicos.



¿Cómo queda armado el rompecabezas?

Transformación de features:

- Manejo de valores faltantes
- Manejo de outliers
- Normalización
- Codificación

Selección de features:

- T-tests
- ANOVA
- Chi- cuadrado correlación

Creación de nuevos features cuando sea necesario
(con funciones matemáticas, de combinación, temporales, etc.)

Extracción de features:

- PCA
- LDA
- t-SNE
- UMAP

Feature Engineering

Ejemplo práctico

