Reto: PCA y t-SNE con el dataset Heart Disease (Kaggle)

El dataset Heart Disease UCI contiene información clínica de pacientes (edad, colesterol, presión arterial, frecuencia cardiaca, tipo de dolor de pecho, azúcar en sangre, etc.), y una variable `target` que indica presencia o ausencia de enfermedad cardíaca. El objetivo de este reto es aplicar técnicas de reducción de dimensionalidad para explorar patrones ocultos y mejorar la comprensión de los datos: - PCA (Análisis de Componentes Principales) → técnica lineal. - t-SNE (t-distributed Stochastic Neighbor Embedding) → técnica no lineal.

Parte 1: Preparación de los datos

1. Descarga el dataset de Kaggle: Heart Disease UCI.
2. Carga el archivo `heart.csv` en tu entorno de Python.
3. Separa las características (`X`) y la variable objetivo (`y = target`).
4. Escala las variables (`StandardScaler`) para que todas tengan media 0 y varianza 1.
5. Para este reto, selecciona al menos 5 características relevantes: Ejemplo: `age`, `trestbps` (presión arterial en reposo), `chol` (colesterol sérico), `thalach` (frecuencia cardiaca máxima) y `oldpeak` (depresión del ST inducida por ejercicio).

Qué se espera: Matriz X_scaled normalizada con las características elegidas y vector y con 0/1.

Parte 2: PCA (2D y 3D)

1. Aplica PCA con 2 componentes principales sobre las características seleccionadas.
2. Visualiza un scatterplot 2D coloreando por `y`.
3. Calcula la varianza explicada acumulada en 2D.
4. Aplica PCA con 3 componentes principales y visualiza en 3D.
5. Calcula la varianza acumulada en 3D.

Preguntas de reflexión sobre PCA:

- ¿Qué significa que el PCA 2D capture, por ejemplo, 45% de la varianza?
- ¿Por qué es importante escalar los datos antes de aplicar PCA?
- ¿Qué diferencias observas entre PCA 2D y PCA 3D en cuanto a la separación de las clases?
- ¿Crees que PCA es útil para clasificación en este dataset? ¿Por qué?
- Si usáramos 10 componentes en lugar de 2 o 3, ¿qué cambiaría en la varianza explicada y en la visualización?

Parte 3: t-SNE (2D y 3D)

1. Aplica t-SNE con 2 dimensiones (perplexity=20, learning_rate=200) sobre las características seleccionadas.
2. Visualiza los resultados en scatterplot 2D coloreando por `y`.
3. Aplica t-SNE con 3 dimensiones y visualiza en 3D.
4. Cambia perplexity a 5, 30 y 50 y compara visualizaciones.
5. Calcula la métrica trustworthiness para evaluar preservación de vecindarios.

Preguntas de reflexión sobre t-SNE:

- ¿Qué papel juega el parámetro `perplexity` en t-SNE y cómo afecta la visualización?
- ¿Por qué t-SNE es más costoso en tiempo que PCA?
- ¿Qué diferencias principales observas entre la distribución de puntos en PCA vs t-SNE?

- ¿Por qué se dice que t-SNE no es recomendable como entrada para un clasificador supervisado?
- ¿En qué escenarios del mundo real preferirías usar t-SNE en lugar de PCA?

Parte 4: Comparación PCA vs t-SNE

1. Construye una tabla comparativa con métricas: Varianza acumulada (PCA), Trustworthiness, Tiempos, Exactitud KNN (2D y 3D).
2. Resume en un párrafo ventajas y limitaciones de cada técnica.

Entregable del reto

- 4 gráficas (PCA 2D, PCA 3D, t-SNE 2D, t-SNE 3D). - 1 tabla comparativa de métricas. - Respuestas a las 10 preguntas de reflexión.