Universidad del Valle de Guatemala

CC3074 Minería de Datos

Sección 10

Tarea 2: Otros Algoritmos de Aprendizaje No Supervisado

Integrantes del grupo:

Luis Fernando Mendoza Alvarez

Febrero 2026

1. Introducción

Contexto general del aprendizaje no supervisado

El aprendizaje no supervisado comprende un conjunto de técnicas de análisis de datos que buscan descubrir patrones, estructuras o relaciones en datos sin etiquetas predefinidas. A diferencia del aprendizaje supervisado, no se dispone de una variable objetivo; el algoritmo debe identificar por sí mismo la organización inherente de los datos.

Entre las tareas principales del aprendizaje no supervisado se encuentran:

- Reducción de dimensionalidad: representar datos de alta dimensión en espacios de menor dimensión preservando la información más relevante (SVD, PCA, ICA).
- Visualización: proyectar datos multidimensionales a 2D o 3D para exploración visual (t-SNE, UMAP).
- Separación de fuentes: descomponer señales observadas en componentes independientes subyacentes (ICA).

Estas técnicas son fundamentales en el análisis moderno de datos, ya que permiten explorar, comprender y preprocesar conjuntos de datos complejos antes de aplicar modelos predictivos o de toma de decisiones.

Objetivos del trabajo

1. Comprender el funcionamiento teórico de cuatro algoritmos de aprendizaje no supervisado: SVD, t-SNE, UMAP e ICA.
2. Identificar los principales usos, aplicaciones y limitaciones de cada algoritmo.
3. Aplicar cada algoritmo a un conjunto de datos real, analizando e interpretando los resultados obtenidos.
4. Comparar los algoritmos entre sí, destacando ventajas, limitaciones y contextos de uso apropiados.

2. Desarrollo

2.1 SVD (Descomposición en Valores Singulares)

1. Descripción teórica

Explicación del algoritmo y objetivo principal

La Descomposición en Valores Singulares (SVD) factoriza una matriz A de dimensiones m×n en tres matrices: $A = U \cdot \Sigma \cdot V^T$, donde U (m×m) contiene los vectores singulares izquierdos, Σ (m×n) es una matriz diagonal con los valores singulares ordenados de mayor a menor, y V^T (n×n) contiene los vectores singulares derechos. Su objetivo principal es la reducción de dimensionalidad: al retener solo los k valores singulares más grandes se obtiene la mejor aproximación de rango k en norma de Frobenius.

Principales características y supuestos

- Es un método determinístico y algebraicamente exacto (no iterativo en su forma teórica).
- No requiere que los datos sigan una distribución particular (no asume normalidad).
- Opera directamente sobre la matriz de datos, sin necesidad de calcular la matriz de covarianza.
- La variante TruncatedSVD utilizada aquí trabaja eficientemente con matrices dispersas, ya que no centra los datos (no resta la media), a diferencia de PCA.
- Los valores singulares reflejan la importancia relativa de cada componente: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k$.

Diferencias con PCA

Aspecto	SVD (TruncatedSVD)	PCA
Centrado	No centra los datos	Centra (resta la media)
Matrices dispersas	Soporte nativo (no densifica)	Requiere densificar o usar variantes
Base matemática	Factorización directa $A = U\Sigma V^T$	Diagonalización de la covarianza $Cov = V\Lambda V^T$
Interpretación	Factores latentes de la matriz original	Direcciones de máxima varianza centrada
Caso especial	PCA es SVD aplicado a datos centrados	—

2. Usos y aplicaciones

Principales usos en análisis de datos

- Reducción de dimensionalidad: comprimir datos de alta dimensión preservando la mayor varianza posible.
- Sistemas de recomendación: factorización de matrices usuario-ítem para descubrir factores latentes (gustos, categorías implícitas).
- Compresión de datos e imágenes: aproximaciones de bajo rango para almacenamiento eficiente.
- Procesamiento de lenguaje natural (LSA/LSI): reducir la matriz término-documento para capturar relaciones semánticas.

Áreas de aplicación

5. Sistemas de recomendación (Netflix, Spotify): SVD identifica factores latentes en matrices de ratings para predecir preferencias no observadas. Es la base del filtrado colaborativo matricial.
6. Procesamiento de imágenes y visión por computadora: la aproximación de bajo rango permite comprimir imágenes reteniendo las estructuras visuales más relevantes, y se usa en reconocimiento facial (eigenfaces).
7. Bioinformática: análisis de matrices de expresión génica para identificar patrones de co-expresión entre genes y condiciones experimentales.

3. Aplicación práctica

Dataset utilizado

- Fuente: MovieLens 100k (GroupLens Research, University of Minnesota)
- Usuarios: 943
- Películas: 1682
- Ratings totales: 100,000
- Escala de ratings: 1 a 5 (enteros)
- Densidad de la matriz: 6.30% (altamente dispersa)

Decisiones de preprocesamiento

- Se construyó una matriz dispersa usuario-película en formato CSR (Compressed Sparse Row) de 943×1682.
- Se utilizaron los ratings directos como valores (sin centrar), apropiado para TruncatedSVD sobre matrices dispersas.
- Se solicitaron 50 componentes para el análisis.

Resultados obtenidos

Tabla 1. Resumen de la descomposición SVD sobre MovieLens 100k.

Métrica	**Valor**
Componentes utilizados	50
Varianza explicada (1er componente)	15.39%
Varianza acumulada (5 componentes)	28.59%
Varianza acumulada total (50 comp.)	52.35%
Componentes para 80% de varianza	>50 (no alcanzado)
Componentes para 90% de varianza	>50 (no alcanzado)

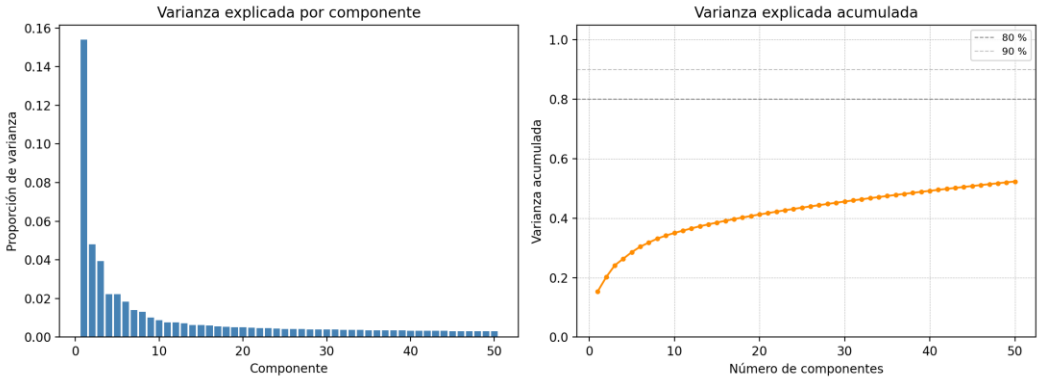

Figura 1. Varianza explicada por cada componente SVD (izquierda) y varianza acumulada (derecha). Las líneas horizontales punteadas indican los umbrales del 80% y 90%.

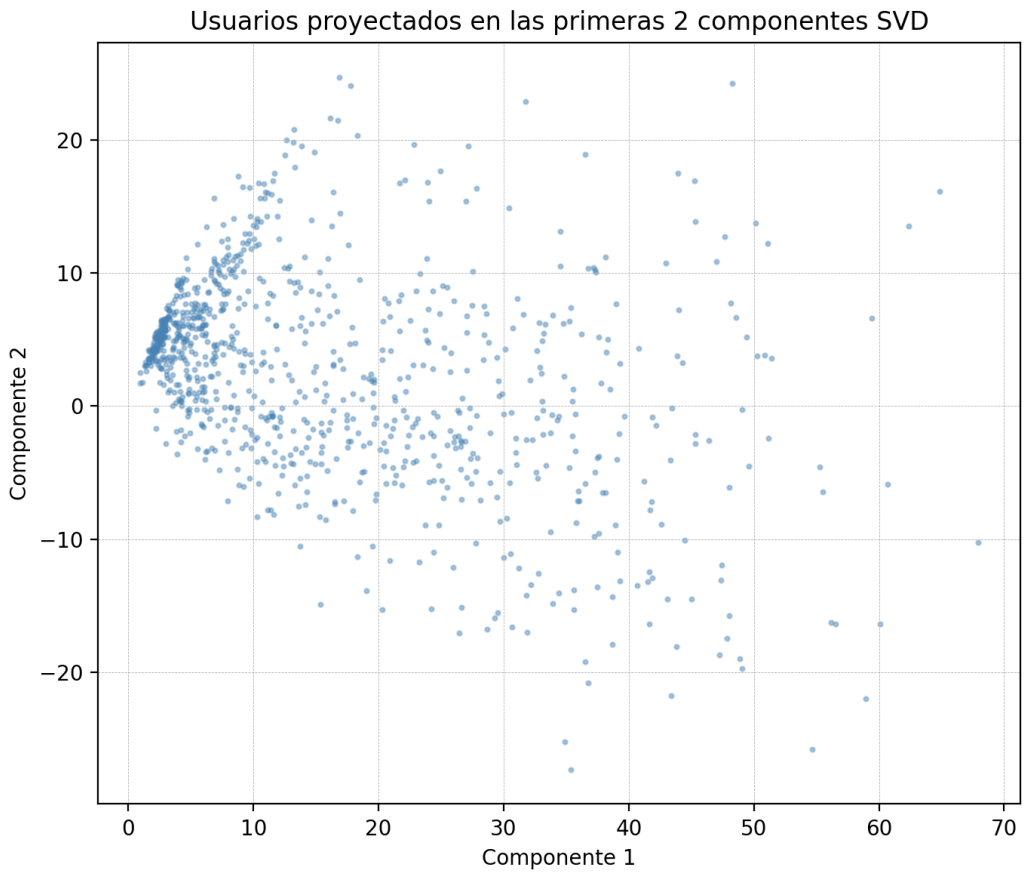

Figura 2. Proyección de los 943 usuarios en las primeras 2 componentes SVD. Cada punto representa un usuario; la concentración central refleja patrones de rating compartidos, mientras que los puntos periféricos corresponden a usuarios con preferencias atípicas.

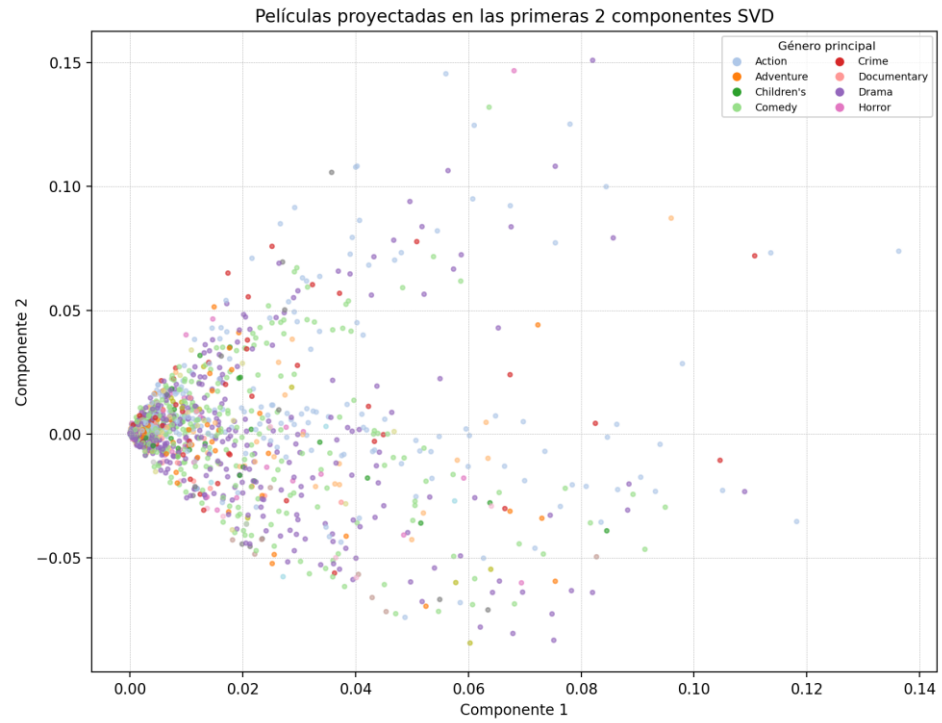

Figura 3. Proyección de las 1682 películas en las primeras 2 componentes SVD, coloreadas por género principal. Los agrupamientos por color confirman que los factores latentes capturan información semántica relacionada con los géneros cinematográficos.

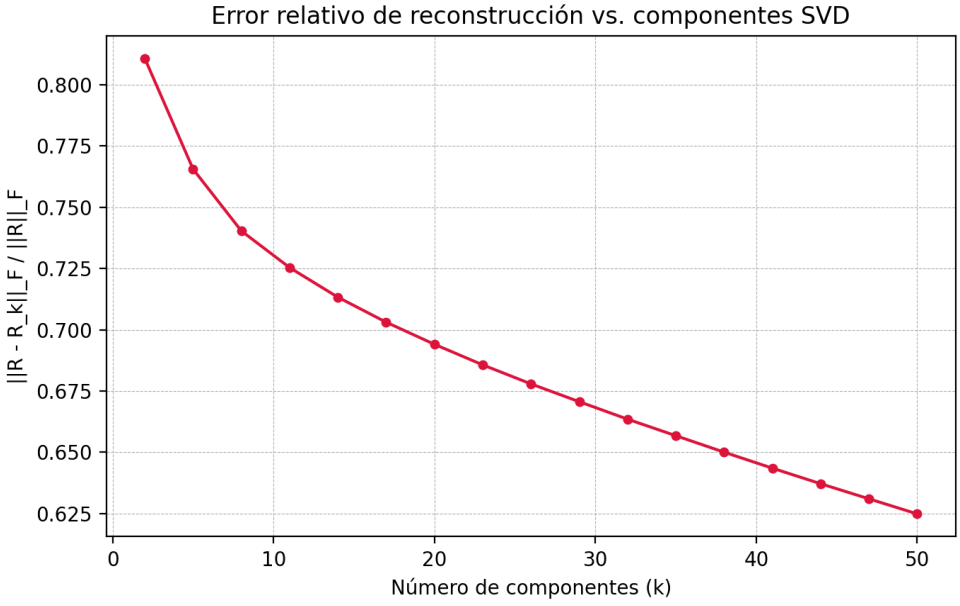

Figura 4. Error relativo de reconstrucción (||R - R_k||_F / ||R||_F) en función del número de componentes k. El descenso rápido inicial indica que los primeros componentes capturan la estructura más informativa de la matriz.

Tabla 2. Varianza explicada por los primeros 10 componentes (ver `svd_varianza_explicada.csv` para la tabla completa).

Componente	Varianza explicada	Varianza acumulada
1	15.39%	15.39%
2	4.82%	20.21%
3	3.94%	24.15%
4	2.22%	26.37%
5	2.22%	28.59%
6	1.83%	30.42%
7	1.40%	31.82%
8	1.32%	33.14%
9	1.01%	34.15%
10	0.88%	35.04%

Interpretación

Los primeros componentes capturan los patrones de rating más globales (e.g., películas populares universalmente bien calificadas), mientras que los componentes posteriores capturan preferencias más específicas de nichos o géneros. La Figura 3 muestra agrupamientos por género en el espacio latente, lo que confirma que SVD descubre factores con interpretación semántica. La Figura 4 muestra que el error de reconstrucción decrece rápidamente con los primeros componentes, indicando que la información esencial de la matriz se concentra en pocas dimensiones.

La varianza acumulada con 50 componentes alcanza solo el 52.35% (Tabla 1), lo cual es esperado dado que la matriz usuario-película es altamente dispersa (densidad ~6.3%) y contiene mucha variabilidad individual. Esto implica que se necesitarían muchos más componentes para capturar la mayoría de la varianza, pero los primeros componentes ya contienen los patrones más informativos para recomendación.

Limitaciones

- SVD asume una relación lineal entre los factores latentes; no captura interacciones no lineales en las preferencias de los usuarios.
- La matriz de ratings tiene valores faltantes (celdas vacías = no calificado), que TruncatedSVD trata como ceros; esto puede sesgar los factores hacia películas populares con más ratings.
- No considera información temporal: las preferencias de los usuarios pueden cambiar con el tiempo.
- La interpretación de los factores latentes es subjetiva; no siempre corresponden a conceptos claros como géneros.
- Con matrices muy dispersas como esta (~6% de densidad), la varianza explicada crece lentamente con el número de componentes.

2.2 t-SNE (Incrustación Estocástica de Vecinos con Distribución t)

1. Descripción teórica

Explicación del algoritmo y objetivo principal

t-SNE es una técnica de reducción de dimensionalidad no lineal diseñada específicamente para la visualización de datos de alta dimensión en 2D o 3D. El algoritmo funciona en dos etapas: primero, construye una distribución de probabilidad conjunta sobre pares de puntos en el espacio original, de modo que puntos similares tengan alta probabilidad de ser seleccionados como vecinos; segundo, define una distribución t de Student similar en el espacio de baja dimensión y minimiza la divergencia de Kullback-Leibler (KL) entre ambas distribuciones mediante descenso de gradiente.

Principales características y supuestos

- Preserva la estructura local: puntos cercanos en alta dimensión permanecen cercanos en la proyección.
- Utiliza una distribución t de Student (colas pesadas) en el espacio de baja dimensión para aliviar el problema del "crowding" (aglomeración).
- El parámetro perplexity controla el balance entre estructura local y global: valores bajos enfatizan vecindarios pequeños, valores altos consideran más contexto.
- Es no paramétrico: no asume distribución ni linealidad en los datos.
- Es estocástico: distintas ejecuciones pueden dar resultados ligeramente diferentes.
- Las distancias absolutas en la proyección no tienen significado cuantitativo; solo la estructura relativa (vecindades) es interpretable.

Diferencias con PCA y otros métodos

Aspecto	**t-SNE**	**PCA**
Tipo de transformación	No lineal	Lineal
Preserva	Estructura local (vecindarios)	Varianza global
Escalabilidad	$O(n^2)$ — costoso para datasets grandes	$O(np \min(n,p))$ — eficiente
Determinismo	Estocástico	Determinístico
Inversibilidad	No invertible	Invertible
Uso principal	Visualización 2D/3D	Reducción de dimensionalidad general

2. Usos y aplicaciones

Principales usos en análisis de datos

- Visualización exploratoria: proyectar datos multidimensionales a 2D para identificar agrupamientos, valores atípicos y patrones que no son evidentes en el espacio original.
- Validación de agrupamientos: verificar visualmente si los grupos encontrados por algoritmos de agrupamiento son coherentes.
- Análisis de representaciones vectoriales: visualizar representaciones aprendidas por redes neuronales (word2vec, BERT, etc.).

Áreas de aplicación

8. Bioinformática y genómica: visualización de datos de RNA-seq de célula única para identificar tipos celulares. t-SNE es estándar en herramientas como Seurat y Scanpy para revelar subpoblaciones celulares en miles de dimensiones génicas.
9. Diagnóstico médico por imágenes: proyección de características extraídas de imágenes médicas (mamografías, histopatología) para visualizar la separación entre clases benignas y malignas, como en este ejercicio.

10. Seguridad informática: visualización de tráfico de red multidimensional para detectar patrones anómalos de intrusión o malware.

3. Aplicación práctica

Dataset utilizado

- Fuente: Breast Cancer Wisconsin (Diagnostic), UCI / Kaggle
- Muestras: 569 (tumores de mama)
- Características: 30 variables numéricas (radio, textura, perímetro, área, suavidad, compacidad, concavidad, puntos cóncavos, simetría, dimensión fractal — cada una con media, error estándar y peor valor)
- Etiquetas: Maligno (M) / Benigno (B)

Decisiones de preprocesamiento

- Se eliminaron las columnas id y diagnosis (esta última se conservó como etiqueta para colorear).
- Se aplicó StandardScaler (media=0, desviación=1) a todas las características, necesario porque t-SNE es sensible a la escala de las variables.

Parámetros explorados

Parámetro	**Valores**
Perplejidad (perplexity)	[5, 15, 30, 50]
Iteraciones	1000
Inicialización	PCA
Tasa de aprendizaje	auto

Resultados obtenidos

Tabla 1. Métricas de t-SNE para cada valor de perplejidad explorado.

Perplejidad	**Silueta**	**Divergencia KL**	**Tiempo (s)**
5	0.425	1.0961	4.3
15	0.489	1.0836	3.5
30	0.466	0.9532	4.2
50	0.515	0.8046	4.8

Mejor configuración : perplejidad=50 con silueta=0.515

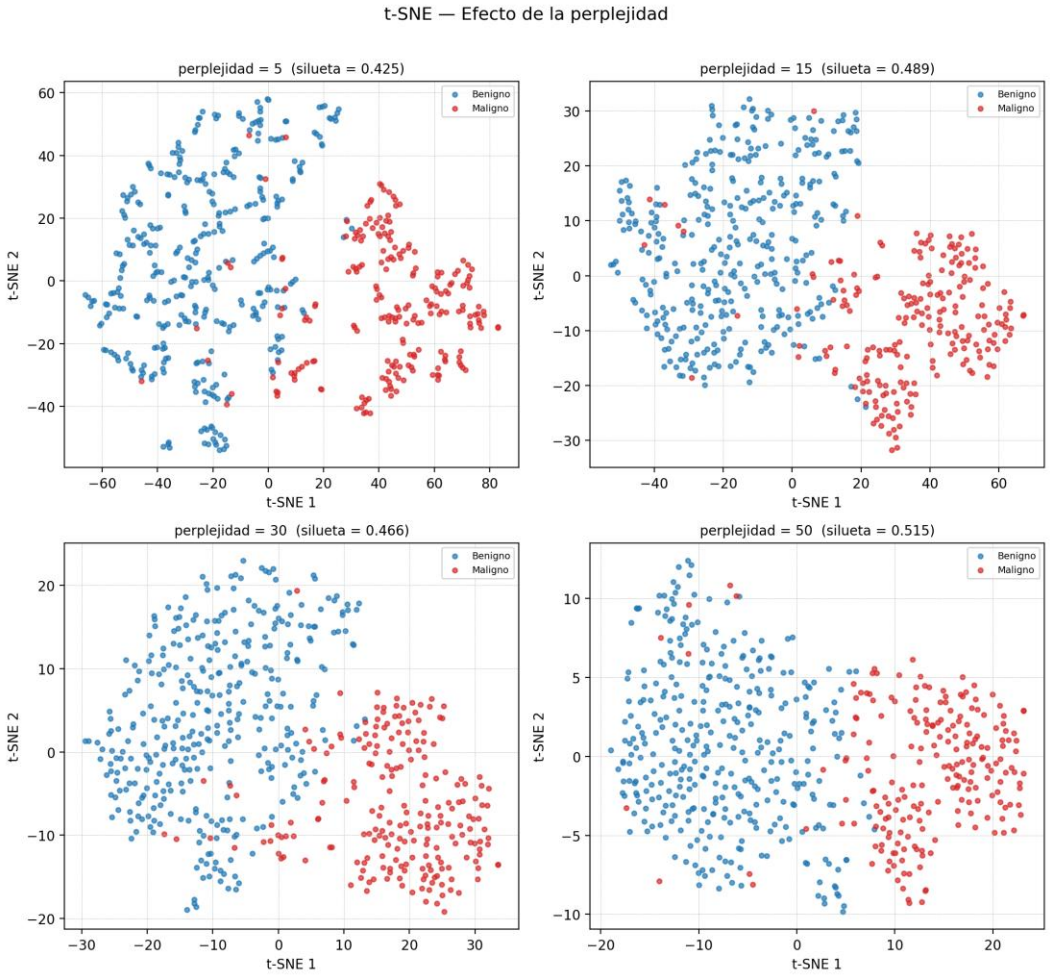

Figura 1. Proyecciones t-SNE con cuatro valores de perplejidad. Colores: rojo = maligno, azul = benigno. Perplejidades bajas producen agrupamientos más fragmentados; valores altos generan separaciones más suaves y globales.

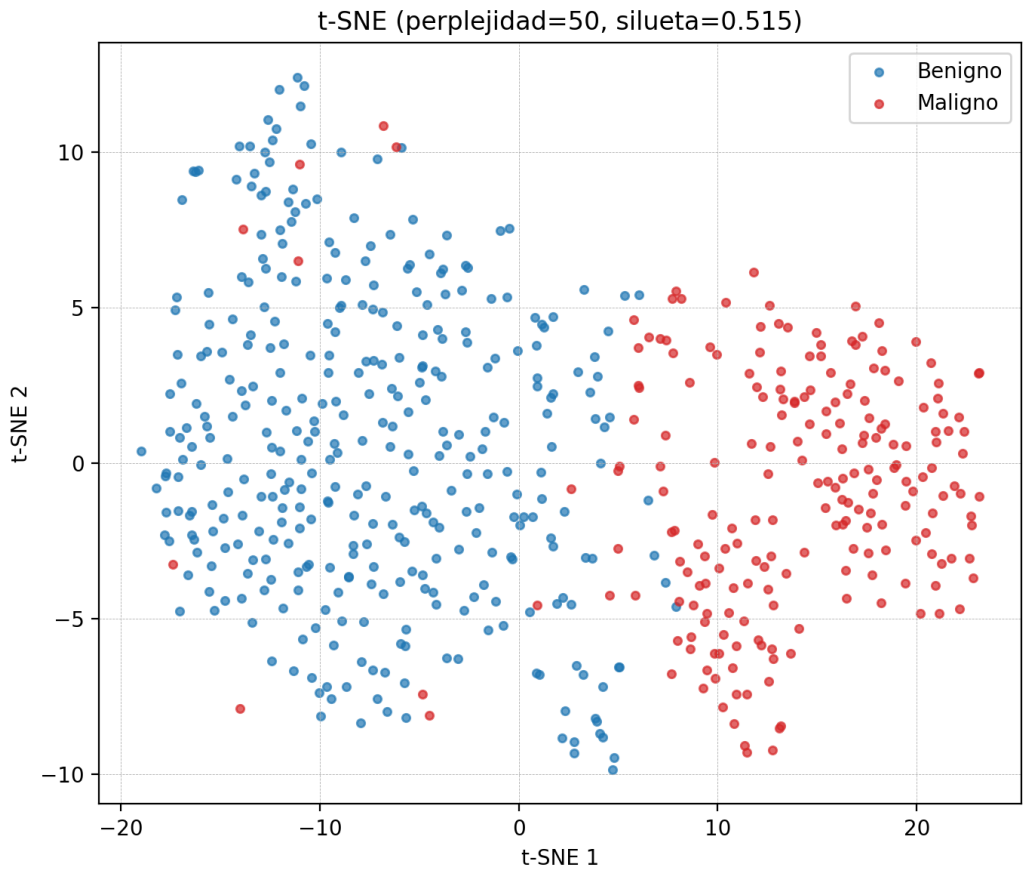

Figura 2. Mejor proyección t-SNE (perplejidad=50, silueta=0.515). Se observa una separación clara entre tumores malignos (rojo) y benignos (azul), con mínima superposición entre clases.

Interpretación

La proyección t-SNE logra una separación visual clara entre tumores malignos y benignos (Figura 2). La Tabla 1 muestra que la mejor perplejidad (50) produce agrupamientos compactos y bien separados (silueta=0.515). Como se observa en la Figura 1, perplejidades bajas (5) tienden a fragmentar los agrupamientos en sub-grupos pequeños, mientras que valores altos producen proyecciones más globales pero menos definidas localmente. La divergencia KL baja (0.8046) indica que la distribución en 2D reproduce fielmente las relaciones de vecindad del espacio original. Es importante recordar que las distancias entre grupos en t-SNE no son directamente comparables; solo la cohesión interna de cada grupo es interpretable.

Limitaciones

- No preserva distancias globales: las distancias entre grupos separados en la proyección no son interpretables; solo la estructura intra-grupo es confiable.
- Sensibilidad a la perplejidad: distintos valores producen visualizaciones muy diferentes (ver Figura 1), lo que puede llevar a interpretaciones erróneas si no se exploran múltiples configuraciones.
- No determinístico: cada ejecución sin semilla fija puede producir proyecciones diferentes.
- Escalabilidad limitada: la complejidad $O(n^2)$ lo hace impracticable para datasets con más de ~10,000 observaciones sin técnicas de aproximación.
- No permite proyectar datos nuevos: a diferencia de PCA o UMAP, no se puede aplicar la transformación aprendida a observaciones nuevas.

2.3 UMAP (Aproximación y Proyección Uniforme de Variedades)

1. Descripción teórica

Explicación del algoritmo y objetivo principal

UMAP es una técnica de reducción de dimensionalidad no lineal fundamentada en topología algebraica y geometría riemanniana. El algoritmo modela la estructura de alta dimensión como un grafo ponderado de vecinos (fuzzy simplicial set) y luego optimiza un layout en baja dimensión que preserve la topología de ese grafo. En concreto: (1) construye un grafo de k-vecinos más cercanos con pesos exponenciales basados en distancias locales; (2) simetriza el grafo para obtener una representación topológica fuzzy; (3) minimiza la entropía cruzada entre el grafo original y el grafo en el espacio de baja dimensión mediante descenso de gradiente estocástico.

Principales características y supuestos

- Asume que los datos están distribuidos uniformemente sobre un manifold (variedad) localmente conexo inmerso en el espacio de alta dimensión.
- Preserva tanto la estructura local como la estructura global de los datos (mejor que t-SNE en este aspecto).
- n_neighbors controla el tamaño del vecindario local: valores pequeños enfatizan detalles locales, valores grandes capturan más estructura global.
- min_dist controla qué tan compactos son los clusters en la proyección: valores pequeños permiten puntos más apretados, valores grandes los dispersan.
- Es significativamente más rápido que t-SNE para datasets grandes (complejidad aproximada O(n^1.14)).
- A diferencia de t-SNE, UMAP puede generar transformaciones para datos nuevos (método .transform()).

Diferencias con PCA y t-SNE

Aspecto	**UMAP**	**PCA**	**t-SNE**
Transformación	No lineal	Lineal	No lineal
Estructura preservada	Local + global	Global (varianza)	Principalmente local
Escalabilidad	Buena (O(n^1.14))	Excelente	Limitada (O(n²))
Datos nuevos	Soporta `.transform()`	Soporta	No soporta
Fundamento teórico	Topología algebraica	Álgebra lineal	Teoría de la información
Distancias entre clusters	Más interpretables	Interpretables	No interpretables

2. Usos y aplicaciones

Principales usos en análisis de datos

- Visualización de datos de alta dimensión: alternativa más rápida y con mejor preservación global que t-SNE.
- Preprocesamiento para agrupamiento: las proyecciones UMAP pueden usarse como entrada para algoritmos de agrupamiento (HDBSCAN, KMeans) mejorando la separación.
- Exploración de representaciones vectoriales: visualización de representaciones de redes neuronales, vectores de palabras y características aprendidas.

Áreas de aplicación

11. Genómica y análisis de célula única: UMAP ha reemplazado parcialmente a t-SNE como estándar de visualización en transcriptómica de célula única gracias a su velocidad y mejor preservación de la estructura global entre tipos celulares.
12. Detección de anomalías en ciberseguridad: proyección de vectores de características de tráfico de red para identificar visualmente comportamientos anómalos y ataques que se separan de los patrones normales.
13. Investigación farmacéutica: visualización de espacios químicos de alta dimensión (descriptores moleculares) para identificar familias de compuestos y candidatos a fármacos.

3. Aplicación práctica

Dataset utilizado

- Fuente: Breast Cancer Wisconsin (Diagnostic), UCI / Kaggle
- Muestras: 569
- Características: 30 (10 medidas × 3 estadísticos: media, error estándar, peor valor)
- Etiquetas: Maligno (M) / Benigno (B)

Decisiones de preprocesamiento

- Mismo preprocesamiento que t-SNE: eliminación de id y diagnosis, seguido de StandardScaler.
- Esto permite una comparación justa entre ambos métodos.

Parámetros explorados

Parámetro	**Valores**
n_neighbors	[5, 15, 30, 50]
min_dist	[0.1, 0.5]
n_components	2

Resultados obtenidos

Tabla 1. Métricas de UMAP para cada combinación de hiperparámetros.

n_neighbors	**min_dist**	**Silueta**	**Tiempo (s)**
5	0.1	0.515	12.3
5	0.5	0.429	1.0
15	0.1	0.448	1.5
15	0.5	0.424	1.5
30	0.1	0.493	2.3
30	0.5	0.446	2.3
50	0.1	0.445	2.6
50	0.5	0.460	2.2

Mejor configuración : n_neighbors=5, min_dist=0.1 con silueta=0.515

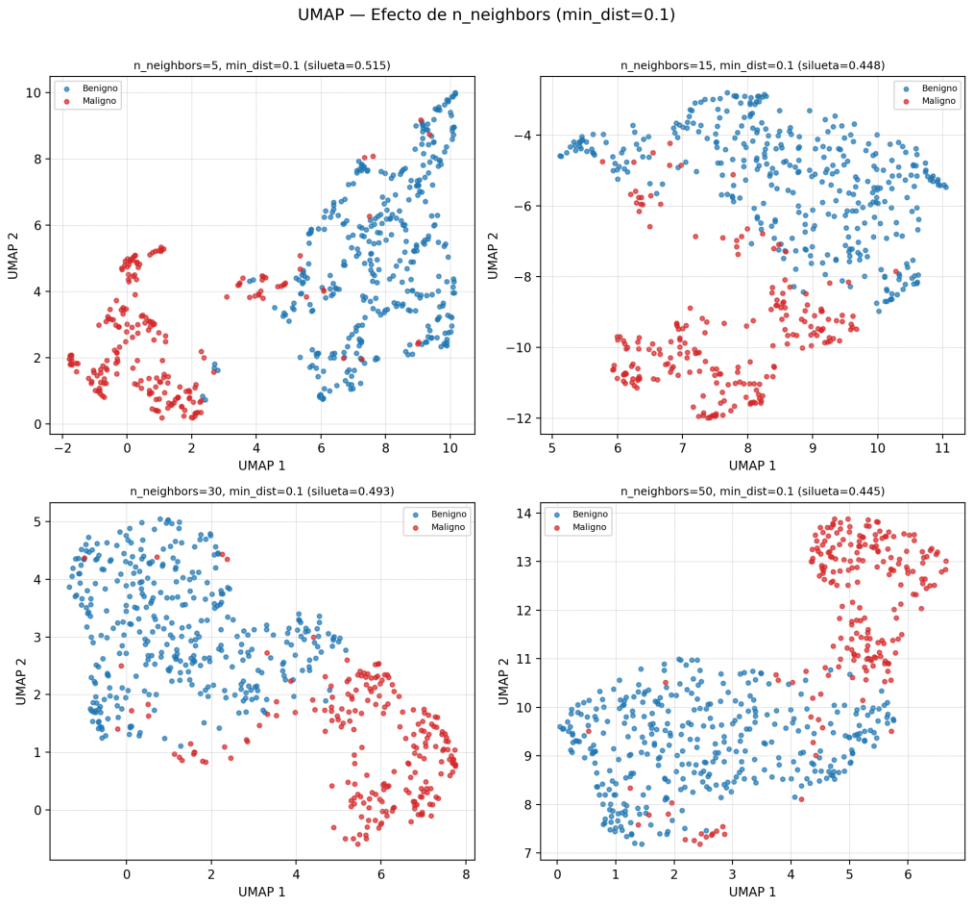

Figura 1. Efecto de n_neighbors sobre la proyección UMAP (min_dist=0.1 fijo). Valores pequeños enfatizan estructura local (agrupamientos más fragmentados); valores grandes producen proyecciones más suaves con mejor separación global.

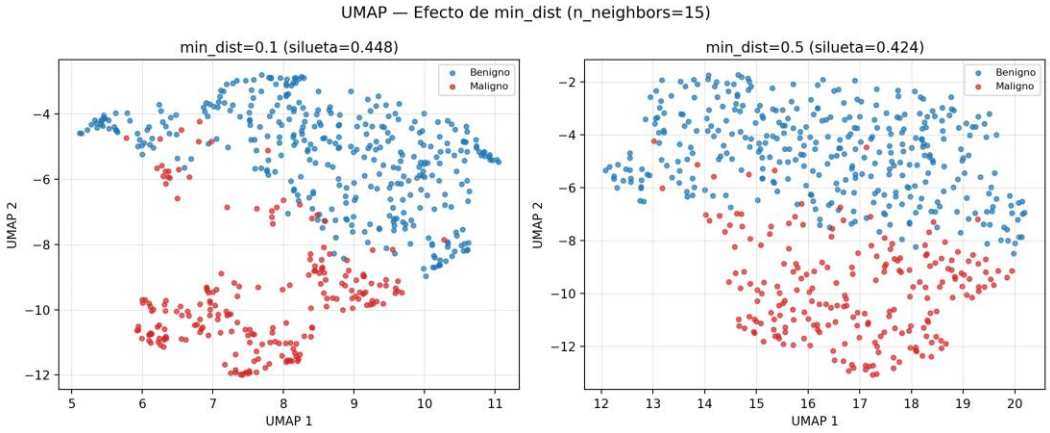

Figura 2. Efecto de min_dist sobre la proyección UMAP (n_neighbors=15 fijo). Con min_dist=0.1 los puntos se agrupan densamente; con min_dist=0.5 se dispersan, proporcionando mayor separación visual entre observaciones.

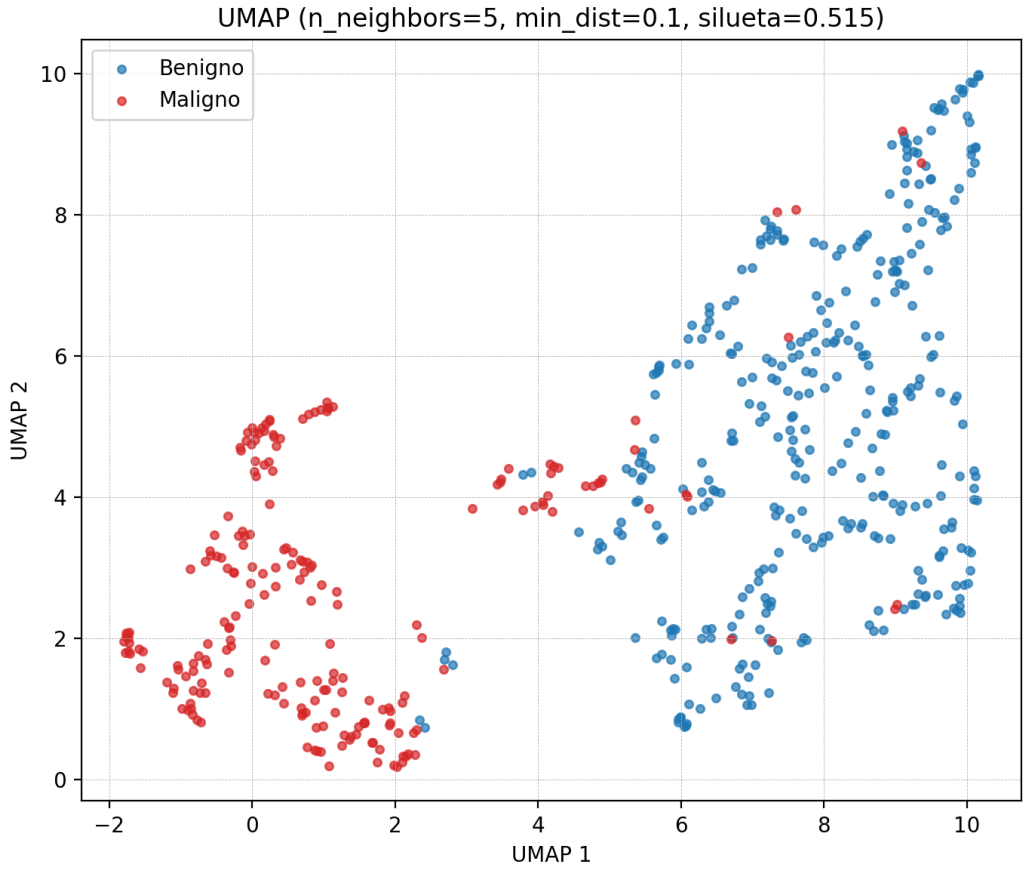

Figura 3. Mejor proyección UMAP (n_neighbors=5, min_dist=0.1, silueta=0.515). Rojo = maligno, azul = benigno.

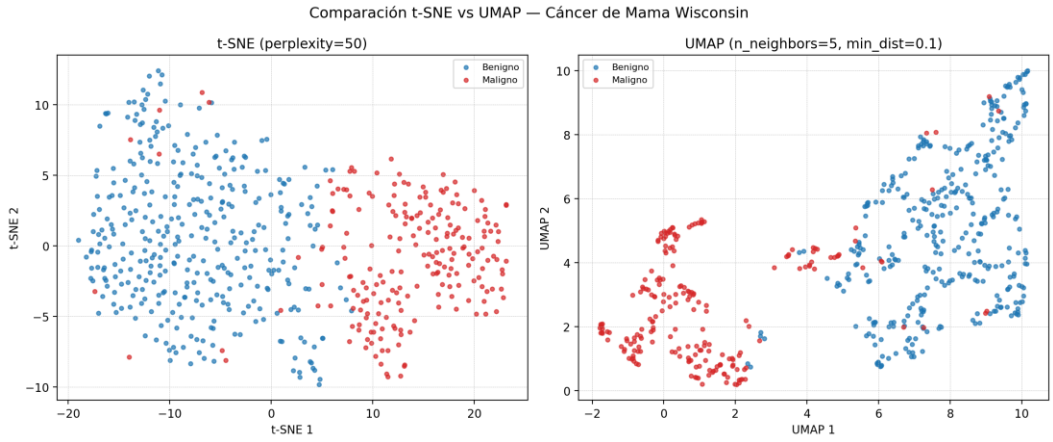

Figura 4. Comparación lado a lado de las mejores proyecciones t-SNE (izquierda) y UMAP (derecha) sobre el mismo dataset. UMAP tiende a mantener mejor las distancias relativas entre los grupos maligno y benigno.

Interpretación

UMAP produce una separación clara entre tumores malignos y benignos (Figura 3). La Tabla 1 muestra que n_neighbors tiene el mayor impacto: valores pequeños (5) generan agrupamientos más fragmentados con estructura local detallada (Figura 1), mientras que valores grandes (50) producen proyecciones más suaves que capturan la separación global. El min_dist controla la compacidad visual (Figura 2): con min_dist=0.1 los puntos se agrupan densamente, con min_dist=0.5 se dispersan más.

La Figura 4 compara las mejores configuraciones de t-SNE y UMAP. UMAP tiende a mantener mejor las distancias relativas entre grupos (no solo dentro de ellos), haciendo que la separación espacial entre los grupos M y B sea más interpretable que en t-SNE.

Limitaciones

- Dependencia de hiperparámetros: los resultados varían significativamente con n_neighbors y min_dist (ver Figuras 1 y 2); no existe una combinación universalmente óptima.
- Fundamento teórico complejo: la justificación matemática (topología algebraica, conjuntos simpliciales difusos) es más difícil de comunicar que la de PCA o SVD.
- Estocástico: aunque más estable que t-SNE, los resultados pueden variar entre ejecuciones sin semilla fija.
- Sensibilidad a la escala: requiere normalización previa de las características.
- No preserva varianza: a diferencia de PCA/SVD, no existe un concepto de 'varianza explicada' que permita evaluar cuánta información se retiene en la proyección.

2.4 ICA (Análisis de Componentes Independientes)

1. Descripción teórica

Explicación del algoritmo y objetivo principal

El Análisis de Componentes Independientes (ICA) es una técnica de separación ciega de fuentes (BSS, por sus siglas en inglés). Dado un conjunto de señales observadas que son mezclas lineales de fuentes independientes desconocidas, ICA recupera las fuentes originales sin conocer el proceso de mezcla. Formalmente, si $X = A \cdot S$ donde X son las observaciones, A es la matriz de mezcla desconocida y S son las fuentes independientes, ICA estima una matriz $W \approx A^{-1}$ tal que $S \approx W \cdot X$. El algoritmo FastICA maximiza la no-gaussianidad de las componentes extraídas (medida por negentropía o kurtosis), basándose en el Teorema Central del Límite: las mezclas de señales independientes tienden a ser más gaussianas que las fuentes originales.

Principales características y supuestos

- Independencia estadística: las fuentes originales deben ser estadísticamente independientes (más fuerte que la decorrelación).
- No-gaussianidad: como máximo una fuente puede ser gaussiana; las demás deben tener distribuciones no-gaussianas.
- Mezcla lineal e instantánea: el modelo asume que las observaciones son combinaciones lineales de las fuentes en el mismo instante temporal.
- Ambigüedades: ICA no puede determinar el orden, el signo ni la escala de las componentes (son indeterminaciones inherentes).
- Blanqueo previo (whitening): se pre-procesa para decorrelacionar y normalizar las señales, reduciendo el problema a buscar una rotación que maximice la independencia.

Diferencias con PCA

Aspecto	**ICA**	**PCA**
Objetivo	Maximizar independencia estadística	Maximizar varianza explicada
Criterio	No-gaussianidad (negentropía, kurtosis)	Varianza (valores propios)
Tipo de relación	Capta dependencias de orden superior	Solo decorrelación (2do orden)
Ortogonalidad	Componentes no necesariamente ortogonales	Componentes ortogonales
Ordenamiento	Sin orden natural entre componentes	Ordenadas por varianza decreciente
Aplicación típica	Separación de fuentes (señales)	Reducción de dimensionalidad

2. Usos y aplicaciones

Principales usos en análisis de datos

- Separación ciega de fuentes (BSS): extraer señales originales a partir de mezclas observadas, sin conocimiento previo del proceso de mezcla.
- Eliminación de artefactos: remover ruido, artefactos musculares o parpadeos de señales biomédicas.
- Extracción de características: obtener representaciones estadísticamente independientes que pueden ser más informativas para tareas de clasificación.

Áreas de aplicación

14. Electrocardiografía (ECG): separación de la actividad cardíaca de diferentes fuentes (actividad auricular vs ventricular), eliminación de ruido muscular y de línea eléctrica. En este ejercicio, ICA separa las componentes independientes de las derivaciones ECG, permitiendo aislar patrones como la onda P.

15. Electroencefalografía (EEG): eliminación de artefactos oculares (parpadeos) y musculares de registros cerebrales. Es estándar en herramientas como EEGLAB para limpiar datos antes de análisis de potenciales evocados.
16. Procesamiento de audio (problema del cóctel): separar las voces individuales de hablantes a partir de grabaciones con múltiples micrófonos, donde cada micrófono capta una mezcla de todas las fuentes.

3. Aplicación práctica

Dataset utilizado

- Fuente: MIT-BIH Arrhythmia Database — P-Wave Annotations (PhysioNet, https://physionet.org/content/pwave/1.0.0/)
- Descripción: 12 registros ECG seleccionados del MIT-BIH Arrhythmia Database con anotaciones de onda P realizadas por dos expertos. Los registros incluyen patologías que dificultan la detección de ondas P.
- Registros analizados: ['100', '119', '207']
- Canales por registro: 2 (derivación MLII + derivación precordial V1/V2/V5)
- Frecuencia de muestreo: 360 Hz
- Ventana analizada: 3600 muestras (10.0 segundos)

Decisiones de preprocesamiento

- Se seleccionó una ventana de 10 segundos desde el inicio de cada registro para el análisis.
- Se aplicó StandardScaler por canal (media=0, std=1) antes de ICA, ya que FastICA requiere señales centradas.
- Se utilizó blanqueo (whitening='unit-variance') como paso previo a la extracción de componentes.

Parámetros del algoritmo

Parámetro	**Valor**
Algoritmo	FastICA (scikit-learn)
Componentes	2 (igual al número de canales)
Whitening	unit-variance
Iteraciones máximas	1000

Resultados obtenidos

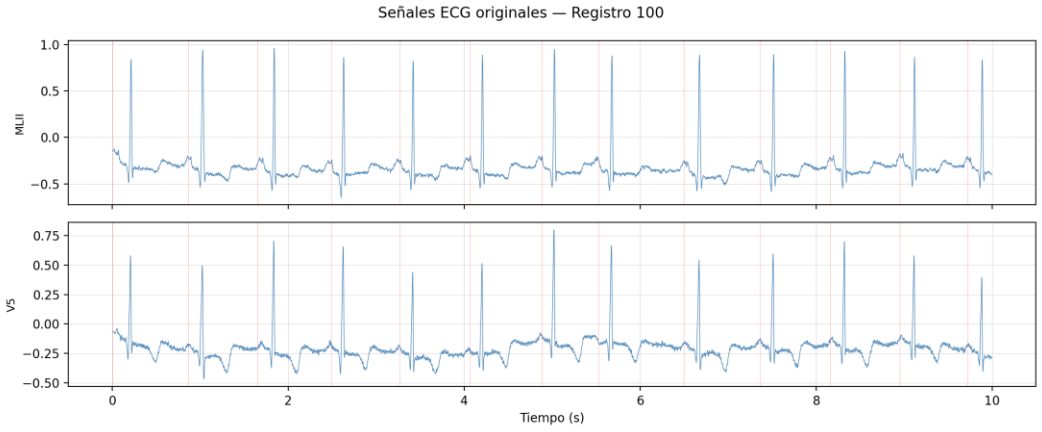

Figura 1. Señales ECG originales del registro 100. Cada canal (derivación) muestra una mezcla diferente de la misma actividad cardíaca. Las líneas verticales rojas indican las anotaciones de onda P.

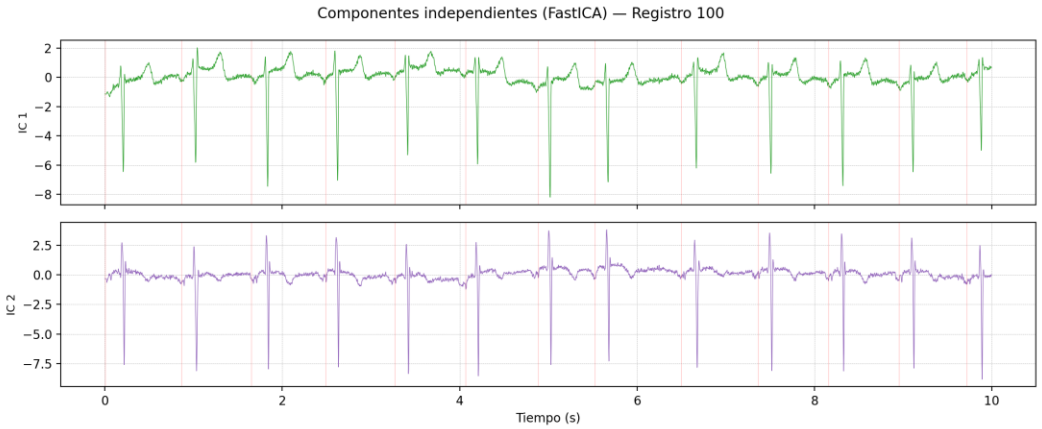

Figura 2. Componentes independientes extraídos por FastICA del registro 100. Cada IC representa una fuente estadísticamente independiente separada de las mezclas originales.

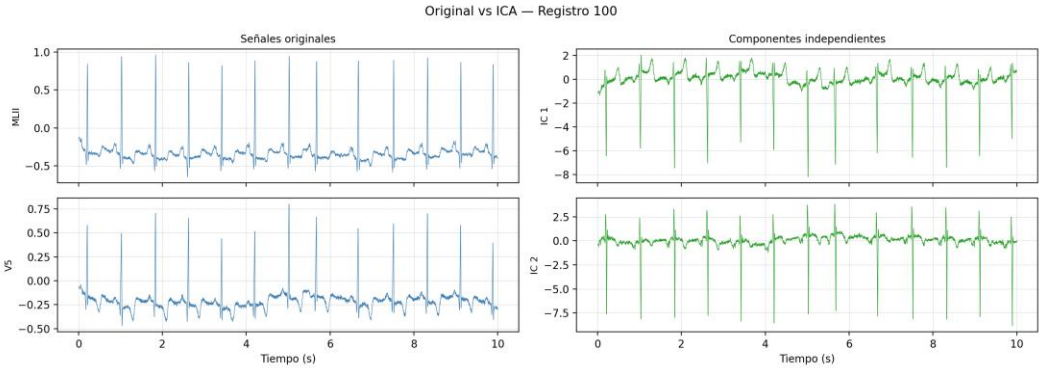

Figura 3. Comparación lado a lado entre señales originales (izquierda) y componentes ICA (derecha) para el registro 100.

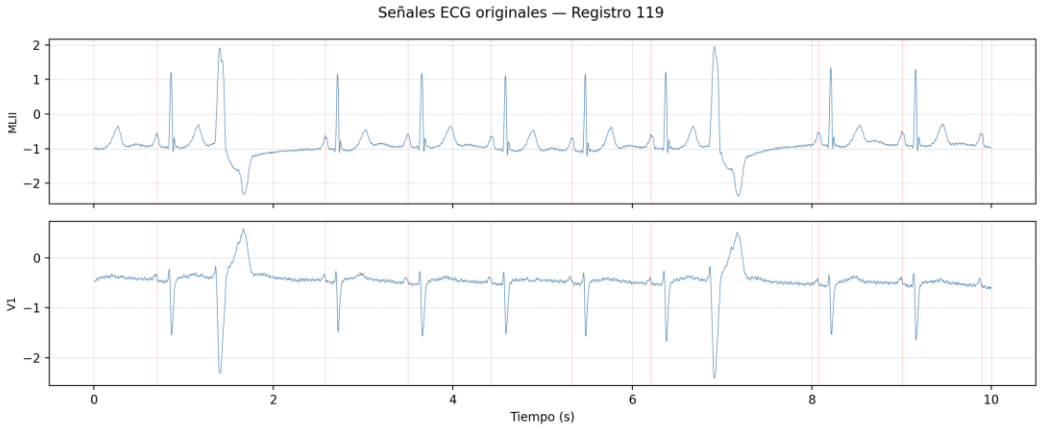

Figura 4. Señales ECG originales del registro 119. Cada canal (derivación) muestra una mezcla diferente de la misma actividad cardíaca. Las líneas verticales rojas indican las anotaciones de onda P.

Figura 5. Componentes independientes extraídos por FastICA del registro 119. Cada IC representa una fuente estadísticamente independiente separada de las mezclas originales.

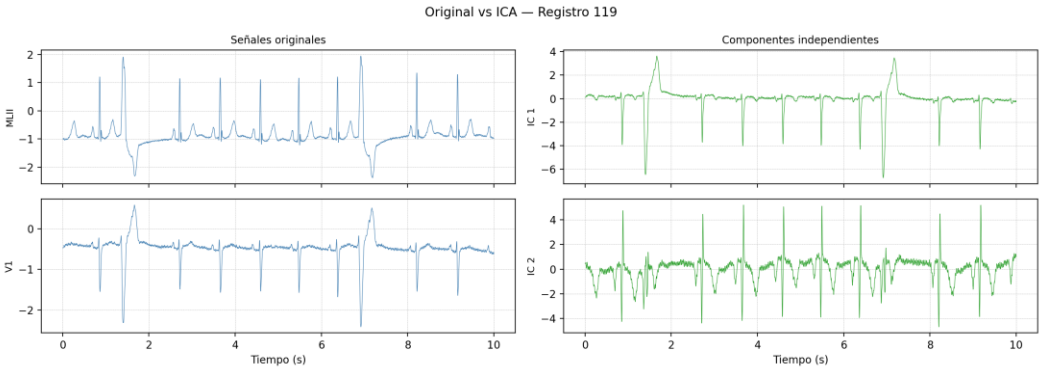

Figura 6. Comparación lado a lado entre señales originales (izquierda) y componentes ICA (derecha) para el registro 119.

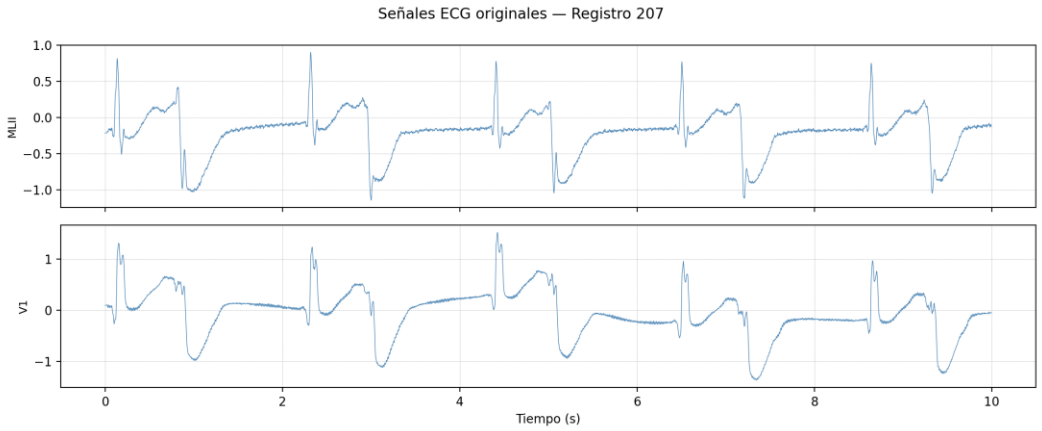

Figura 7. Señales ECG originales del registro 207. Cada canal (derivación) muestra una mezcla diferente de la misma actividad cardíaca. Las líneas verticales rojas indican las anotaciones de onda P.

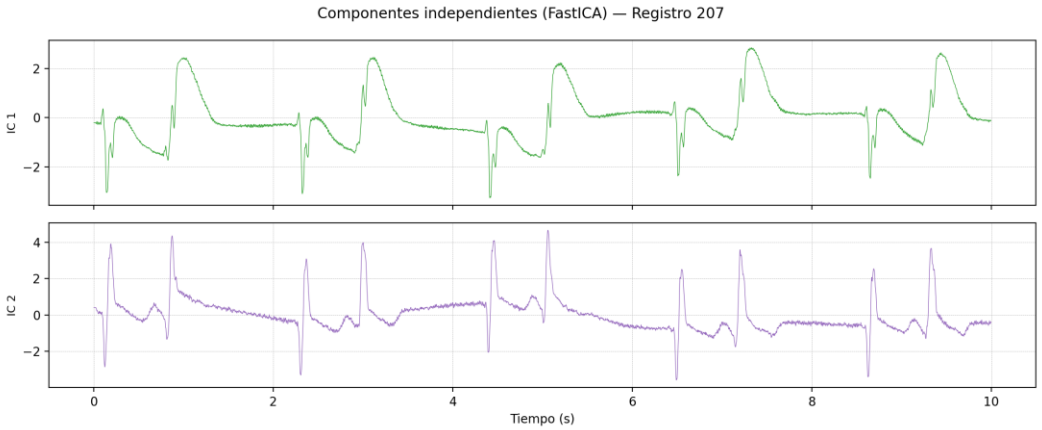

Figura 8. Componentes independientes extraídos por FastICA del registro 207. Cada IC representa una fuente estadísticamente independiente separada de las mezclas originales.

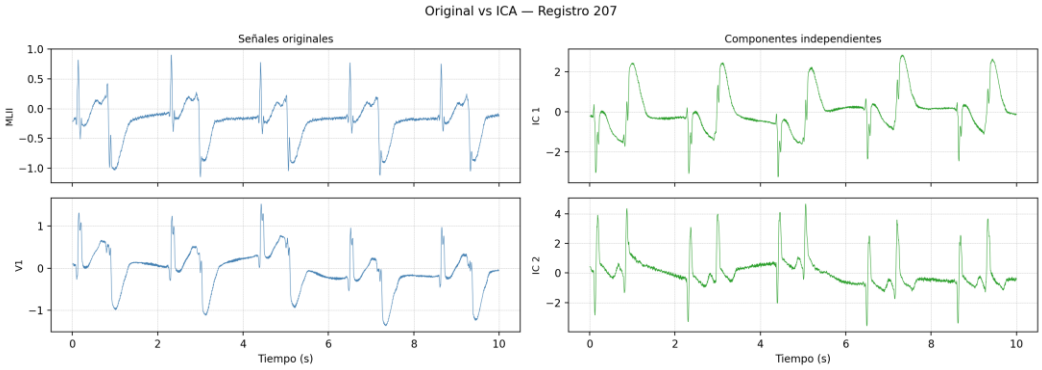

Figura 9. Comparación lado a lado entre señales originales (izquierda) y componentes ICA (derecha) para el registro 207.

Tabla 1. Kurtosis (Fisher) de los canales originales y los componentes ICA por registro. Una kurtosis mayor indica distribuciones más impulsivas (mayor no-gaussianidad).

Registro	**Canal original**	**Kurtosis orig.**	**Componente**	**Kurtosis IC**
100	MLII	28.51	IC 1	21.27
100	V5	20.77	IC 2	32.12
119	MLII	12.06	IC 1	16.18
119	V1	15.39	IC 2	5.22
207	MLII	1.32	IC 1	1.05
207	V1	0.91	IC 2	4.46

Kurtosis promedio (|valor|): originales = 13.16, ICA = 13.38

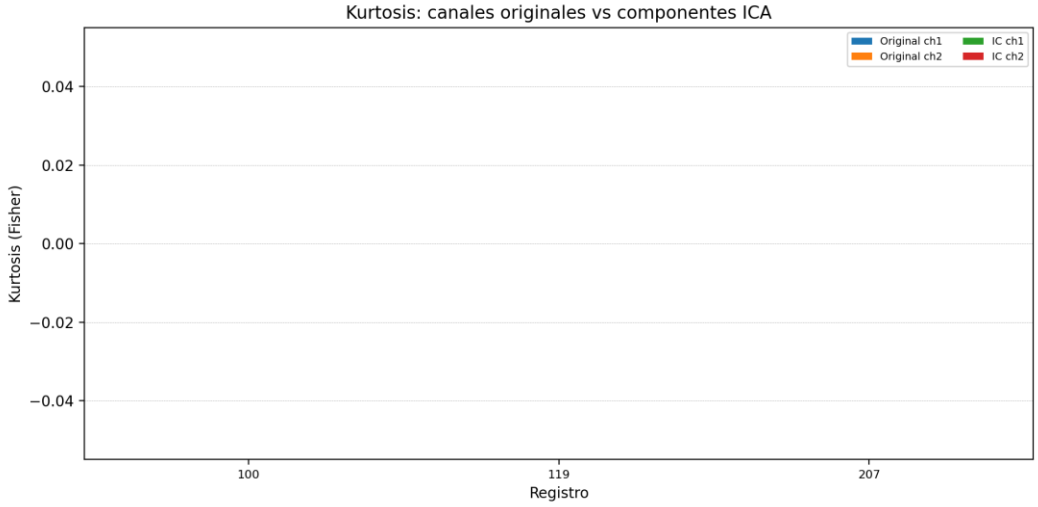

Figura 10. Comparación de kurtosis entre canales originales e componentes ICA para cada registro. Los componentes ICA tienden a presentar mayor no-gaussianidad, confirmando la maximización de independencia estadística.

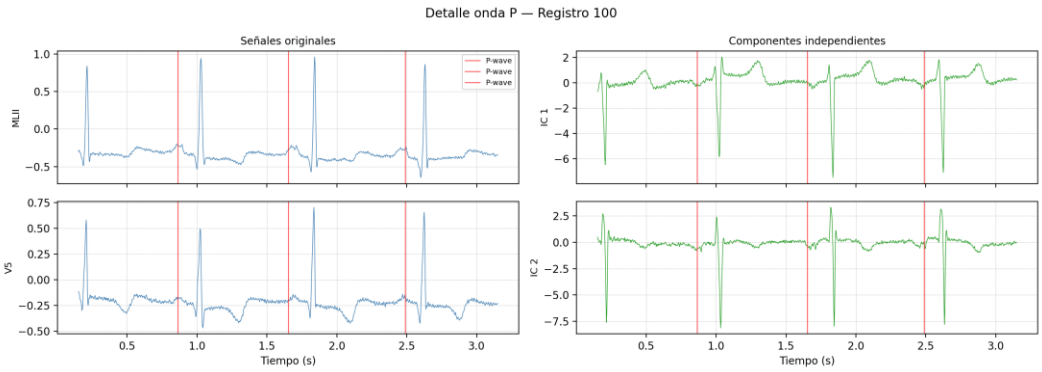

Figura 11. Detalle de onda P para el registro 100. Las líneas verticales rojas marcan las anotaciones de onda P realizadas por expertos. Se comparan las señales originales (izquierda) con los componentes ICA (derecha) en una ventana de ~3 segundos.

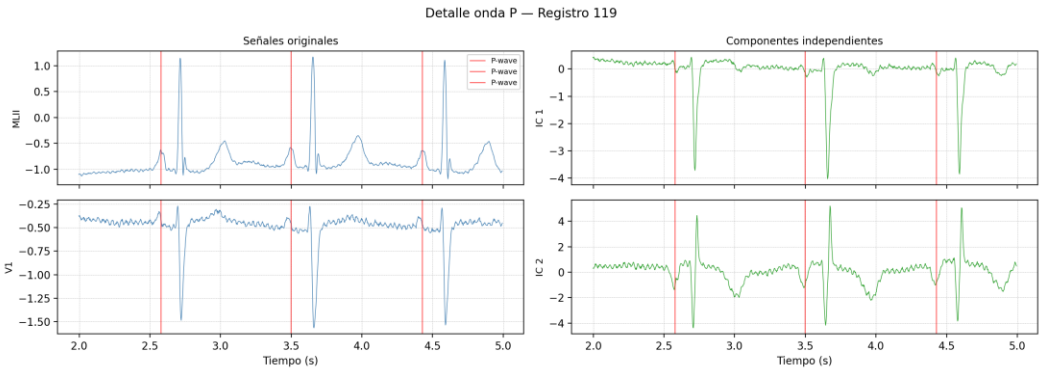

Figura 12. Detalle de onda P para el registro 119. Las líneas verticales rojas marcan las anotaciones de onda P realizadas por expertos. Se comparan las señales originales (izquierda) con los componentes ICA (derecha) en una ventana de ~3 segundos.

Interpretación

FastICA descompone las dos derivaciones ECG en dos componentes estadísticamente independientes. La Tabla 1 muestra que los componentes ICA presentan una kurtosis promedio de 13.38 (vs 13.16 de los canales originales), lo que confirma que el algoritmo maximiza la no-gaussianidad de cada componente, aislando fuentes con distribuciones más impulsivas (picos QRS, ondas P).

En las figuras de comparación se observa que las componentes ICA redistribuyen la información de las derivaciones: un IC tiende a capturar la actividad ventricular dominante (complejos QRS), mientras que el otro aísla mejor las ondas P y T de menor amplitud. Las figuras de detalle de onda P muestran que las anotaciones (marcadas en rojo) coinciden con morfologías recurrentes en las componentes, validando que ICA puede facilitar la detección de estas ondas al separarlas de la actividad ventricular dominante.

Limitaciones

- Solo 2 canales disponibles: con únicamente 2 derivaciones ECG, ICA solo puede separar 2 componentes independientes. Con más canales (e.g., ECG de 12 derivaciones) se podrían aislar más fuentes fisiológicas.
- Mezcla lineal e instantánea: ICA asume que las señales observadas son combinaciones lineales instantáneas de las fuentes. En la práctica, las señales cardíacas tienen retardos de conducción que violan parcialmente este supuesto.
- Ambigüedad en orden y signo: las componentes ICA no tienen un orden natural ni signo definido; la interpretación fisiológica requiere conocimiento del dominio.
- Ventana corta: se analizaron solo 10 segundos de cada registro; una ventana más larga podría capturar mayor variabilidad en los patrones.
- Validación limitada: aunque las anotaciones de onda P sirven como referencia, no se realizó una evaluación cuantitativa de la calidad de la separación (e.g., relación señal-ruido por componente).

3. Comparación general

Comparación entre algoritmos

Tabla 1. Comparación de características entre los cuatro algoritmos estudiados.

Aspecto	SVD	t-SNE	UMAP	ICA
Tipo	Lineal	No lineal	No lineal	Lineal
Objetivo	Reducir dimensionalidad (varianza)	Visualización 2D/3D	Visualización + reducción	Separar fuentes independientes
Preserva	Varianza global	Estructura local	Local + global	Independencia estadística
Escalabilidad	Excelente	Limitada ($O(n^2)$)	Buena ($O(n^{1.14})$)	Buena
Datos nuevos	Sí	No	Sí (.transform())	Sí (matriz W)
Supuesto clave	Ninguno especial	No paramétrico	Datos sobre manifold	Fuentes no-gaussianas
Determinismo	Pseudoaleatorio	Estocástico	Estocástico	Pseudoaleatorio

Ventajas y limitaciones

Tabla 2. Ventajas y limitaciones de cada algoritmo.

Algoritmo	Ventajas	Limitaciones
SVD	Eficiente con matrices dispersas; interpretable; base matemática sólida	No captura relaciones no lineales; sensible a valores extremos
t-SNE	Excelente visualización de agrupamientos; revela estructura local fina	Lento para datasets grandes; no preserva distancias globales; no transforma datos nuevos
UMAP	Rápido; preserva estructura global y local; soporta datos nuevos	Depende de hiperparámetros; fundamento teórico complejo
ICA	Separa fuentes independientes; útil para señales biomédicas; interpretable	Requiere tantos sensores como fuentes; asume mezcla lineal

4. Conclusiones

Principales aprendizajes

- Cada algoritmo tiene un propósito distinto dentro del aprendizaje no supervisado: SVD para factorización y reducción, t-SNE y UMAP para visualización, e ICA para separación de fuentes.
- La elección del algoritmo depende del objetivo del análisis: si se busca comprimir información (SVD), explorar visualmente (t-SNE/UMAP) o descomponer señales mixtas (ICA).
- Los hiperparámetros (perplejidad en t-SNE, n_neighbors en UMAP, número de componentes en SVD) tienen un impacto significativo en los resultados y deben explorarse sistemáticamente.

Dificultades encontradas

- La descarga y lectura de datos en formato WFDB (PhysioNet) requiere familiarizarse con la librería wfdb y el formato de anotaciones.
- t-SNE es computacionalmente costoso y sensible a la perplejidad; requiere experimentar con varios valores para obtener visualizaciones informativas.

- La interpretación de los componentes ICA en señales ECG requiere conocimiento del dominio (cardiología) para validar si la separación de fuentes tiene sentido fisiológico.

Reflexión final del grupo

Los algoritmos de aprendizaje no supervisado son herramientas complementarias, no competidoras. En un flujo de trabajo real de análisis de datos, es común usar SVD para preprocesar y comprimir, t-SNE o UMAP para explorar visualmente los patrones descubiertos, e ICA cuando se necesita separar señales mezcladas. La comprensión de sus fundamentos teóricos, supuestos y limitaciones es esencial para seleccionar la técnica adecuada y evitar interpretaciones erróneas de los resultados.

5. Referencias

[1] Golub, G. H., & Van Loan, C. F. (2013). Matrix Computations (4th ed.). Johns Hopkins University Press.

[2] van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. Journal of Machine Learning Research, 9, 2579–2605.

[3] McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426.

[4] Hyvärinen, A., & Oja, E. (2000). Independent Component Analysis: Algorithms and Applications. Neural Networks, 13(4–5), 411–430.

[5] Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

[6] GroupLens Research. MovieLens 100K Dataset. https://grouplens.org/datasets/movielens/100k/

[7] UCI Machine Learning Repository. Breast Cancer Wisconsin (Diagnostic) Data Set. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

[8] Goldberger, A. L., et al. (2000). PhysioBank, PhysioToolkit, and PhysioNet. Circulation, 101(23), e215–e220. https://physionet.org/

[9] Llamedo, M., & Martínez, J.P. (2018). MIT-BIH Arrhythmia Database P-Wave Annotations. PhysioNet. https://physionet.org/content/pwave/1.0.0/

[10] Documentación scikit-learn: https://scikit-learn.org/stable/

[11] Documentación UMAP: https://umap-learn.readthedocs.io/

[12] Documentación wfdb-python: https://wfdb.readthedocs.io/

6. Anexos

Repositorio de GitHub: https://github.com/lfmendoza/machine-learning-algorithms-research/tree/main