

Apuntes Semana 14 - 29/05/2025

Diana Sanabria Calvo 2021436548

Abstract—El aprendizaje no supervisado es una rama fundamental de la inteligencia artificial que se enfoca en identificar patrones, estructuras o relaciones ocultas dentro de conjuntos de datos sin utilizar etiquetas previamente definidas. A través de técnicas como la reducción de dimensionalidad (PCA), el agrupamiento (clustering) con métodos como K-Means, DBSCAN y clustering jerárquico, y el uso de redes neuronales autoencoder, se busca extraer conocimiento de forma autónoma. En esta clase se exploraron estos métodos a nivel teórico y práctico, complementando con visualizaciones, métricas de evaluación (índice de silhouette, método del codo), y una discusión crítica sobre el papel del modelo open-source DeepSeek 3.1, que destaca por su cercanía al rendimiento de modelos privados, reforzando la importancia de la IA libre para la comunidad global.

I. INTRODUCCIÓN

El aprendizaje no supervisado (unsupervised learning) representa un enfoque clave en la inteligencia artificial, permitiendo el descubrimiento automático de patrones sin supervisión humana. A diferencia del aprendizaje supervisado, aquí no se conocen las etiquetas de salida, lo que lo convierte en una herramienta poderosa para explorar datos, segmentar usuarios, identificar anomalías o comprimir información.

Durante la clase del 29 de mayo de 2025, se abordaron en profundidad las técnicas principales de aprendizaje no supervisado: reducción de dimensionalidad (PCA), clustering (K-Means, DBSCAN, jerárquico), distancias y similitudes (Euclídea y coseno), y autoencoders. Además, se discutieron métricas visuales y numéricas para evaluar el rendimiento de los modelos, y se presentó una noticia relevante sobre el avance del modelo DeepSeek 3.1.

I. NOTICIAS DE LA SEMANA

• Lanzamiento de DeepSeek 3.1:

Como parte de la sección habitual de “jueves de noticias”, se mencionó el lanzamiento del modelo DeepSeek 3.1, una evolución del modelo chino DeepSeek R1. Este modelo, open-source, ha demostrado estar muy cercano en rendimiento a modelos privados como GPT-4, lo cual representa un hito en el acceso abierto a tecnología de IA de punta. El modelo DeepSeek 3.1 posiciona a la comunidad como protagonista del desarrollo tecnológico al reducir la dependencia de soluciones privadas y centralizadas, promoviendo la equidad y la innovación abierta.

II. REPASO DE CONCEPTOS

1) *Aprendizaje No Supervisado*: El aprendizaje no supervisado (unsupervised learning) es una rama fundamental de

la inteligencia artificial y el aprendizaje automático que se enfoca en descubrir patrones subyacentes, estructuras ocultas y relaciones dentro de un conjunto de datos sin la necesidad de etiquetas explícitas o respuestas correctas. Es decir, a diferencia del aprendizaje supervisado —donde el modelo se entrena con ejemplos compuestos por pares entrada-salida (por ejemplo, imágenes de gatos con la etiqueta "gato")—, en el aprendizaje no supervisado el algoritmo trabaja únicamente con las entradas, sin conocer de antemano qué debería "aprender".

El objetivo principal es entender la organización interna de los datos para posteriormente extraer información útil, reducir su complejidad o facilitar su análisis. Este tipo de aprendizaje es especialmente valioso en escenarios donde no se dispone de etiquetas debido al alto costo de etiquetado, la falta de conocimiento previo o simplemente porque los datos provienen de fenómenos desconocidos.

III. APLICACIONES MÁS COMUNES DEL APRENDIZAJE NO SUPERVISADO

1) *Detección de grupos (clustering)*: Agrupa instancias similares del conjunto de datos en clústeres o conglomerados. Esta técnica se utiliza ampliamente en marketing para segmentar clientes, en biología para clasificar especies o tipos celulares, y en astronomía para identificar tipos de galaxias o estrellas.

2) *Reducción de dimensionalidad*: Se emplea para representar los datos en un espacio de menor dimensión, conservando la información más relevante. Esto facilita la visualización, el almacenamiento eficiente y puede servir como preprocesamiento para otros algoritmos. El ejemplo más conocido es PCA (Análisis de Componentes Principales).

3) *Compresión de datos*: Utilizando modelos como los autoencoders, el aprendizaje no supervisado permite codificar datos en una forma más compacta y luego reconstruirlos, lo que resulta útil para ahorrar espacio o para eliminar ruido y redundancia.

4) *Visualización de estructuras complejas*: Mediante técnicas como t-SNE, UMAP o PCA, se pueden representar datos de alta dimensión en dos o tres dimensiones para identificar relaciones, anomalías o agrupamientos que no serían evidentes en su forma original.

5) *Detección de anomalías (anomaly detection)*: Permite identificar patrones que se desvían significativamente del comportamiento normal, lo cual es útil en aplicaciones como la detección de fraudes, fallos en sistemas industriales, o vigilancia de seguridad.

IV. REDUCCIÓN DE DIMENSIONALIDAD: ANÁLISIS DE COMPONENTES PRINCIPALES (PCA)

A. ¿Por qué reducir la dimensionalidad?

En problemas de ciencia de datos y aprendizaje automático, es común encontrarse con conjuntos de datos que contienen una gran cantidad de variables o características (features). Esto puede dificultar tanto el análisis como el rendimiento de los modelos. La reducción de dimensionalidad tiene como objetivo proyectar los datos en un espacio de menor dimensión conservando la información más relevante.

Las principales motivaciones para reducir la dimensionalidad incluyen:

Visualización: Los datos con muchas dimensiones no pueden visualizarse fácilmente. Reduciendo a dos o tres dimensiones se puede representar gráficamente la estructura de los datos.

Reducción de ruido: Al eliminar características redundantes o irrelevantes, se mejora la calidad de la representación.

Eficiencia computacional: Menos dimensiones implican menor carga de cálculo, lo cual acelera el entrenamiento de modelos.

Evitar la maldición de la dimensionalidad: En espacios de alta dimensión, los datos tienden a volverse escasos y las distancias pierden significado, dificultando tareas como clustering o clasificación.

V. PRINCIPAL COMPONENT ANALYSIS (PCA)

El Análisis de Componentes Principales (PCA, por sus siglas en inglés) es una técnica estadística utilizada para transformar un conjunto de datos en una nueva base de ejes ortogonales (componentes principales), de manera que:

El primer componente captura la máxima varianza posible de los datos. El segundo componente captura la máxima varianza restante, siendo ortogonal al primero y así sucesivamente.

VI. PASOS DEL PCA

A. Centrar los datos

Para cada variable, se resta su media para que el conjunto de datos quede centrado en el origen.

B. Calcular la matriz de covarianza

Esta matriz captura cómo varían conjuntamente las diferentes características.

$$\Sigma = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T$$

C. Obtener eigenvectores y eigenvalues

Se resuelve el siguiente problema de autovalores:

$$\Sigma v = \lambda v$$

D. Obtener eigenvectores y eigenvalues

Se eligen los k *eigenvectors* con mayor *eigenvalue*, ya que son los que más varianza explican. Los datos originales se proyectan sobre estos nuevos ejes:

$$Z = X \cdot V_k$$

Donde Z es la representación reducida de los datos, X representa los datos centrados (con media cero), y V_k es la matriz que contiene los k *eigenvectores* seleccionados (componentes principales) asociados a los eigenvalores más grandes.

Donde Z es la representación reducida, X son los datos centrados y V_k es la matriz de los k *eigenvectores* seleccionados.

VII. CLUSTERING

El objetivo es dividir las instancias en grupos (clusters) de forma que:

- Los datos dentro del mismo grupo sean similares.
- Los datos de distintos grupos sean diferentes.

Algoritmo k-means

Pasos:

- 1) Inicializar k centroides aleatoriamente.
- 2) Asignar cada punto al centroide más cercano.
- 3) Recalcular los centroides.
- 4) Repetir hasta convergencia.

Función objetivo de K-Means

Función objetivo:

$$\min \sum_{i=1}^k \sum_{x_j \in C_i} \|x_j - \mu_i\|^2$$

Donde C_i representa el conjunto de puntos asignados al i -ésimo centroide, x_j es un punto de datos, y μ_i es el centroide del grupo i .

Donde C_i representa el conjunto de puntos asignados al i -ésimo centroide, x_j es un punto de datos, y μ_i es el centroide del grupo i .

Clustering jerárquico

El clustering jerárquico es una técnica de aprendizaje no supervisado que busca construir una jerarquía de grupos o clústeres mediante un enfoque estructurado en forma de árbol, comúnmente representado por un dendrograma. Este dendrograma visualiza la relación y similitud entre las instancias del conjunto de datos, permitiendo observar la fusión o separación de grupos a diferentes niveles.

A diferencia de algoritmos como K-Means que requieren la especificación previa del número de clústeres, el clustering jerárquico permite analizar la estructura de los datos en distintos niveles de granularidad, facilitando así la elección

posterior del número de grupos óptimo al "cortar" el dendrograma en diferentes alturas.

VIII. AUTOENCODERS

Los **autoencoders** son un tipo de red neuronal utilizada comúnmente para la reducción de dimensionalidad, compresión de datos, detección de anomalías y preentrenamiento de redes profundas. Su objetivo principal es aprender una representación (también conocida como *codificación*) de los datos de entrada, y luego reconstruirlos de la manera más precisa posible.

A diferencia de otros métodos lineales como PCA, los autoencoders pueden aprender representaciones no lineales, lo cual los vuelve muy útiles para datos complejos.

A. Arquitectura

Arquitectura de un Autoencoder

La arquitectura típica de un autoencoder se compone de tres partes fundamentales:

- **Encoder (codificador):** transforma la entrada original x en una representación de menor dimensión z . Esto se logra aplicando una función f_θ parametrizada por pesos y sesgos:

$$z = f_\theta(x)$$

- **Latent Space o Bottleneck:** es el espacio de menor dimensión donde se ubica la representación comprimida de los datos. Actúa como un cuello de botella que fuerza al modelo a aprender las características más importantes.
- **Decoder (decodificador):** reconstruye la entrada original a partir del vector codificado z usando una función g_ϕ :

$$\hat{x} = g_\phi(z)$$

Esta arquitectura en forma de "sandwich" fuerza a la red a aprender una compresión útil de los datos que conserve la información más relevante.

B. Función de pérdida

El objetivo del entrenamiento es minimizar la diferencia entre la entrada original x y su reconstrucción \hat{x} . Para ello se utiliza una función de pérdida basada en el error cuadrático medio (MSE):

$$L(x, \hat{x}) = \|x - \hat{x}\|^2$$

Donde:

- x es el vector de entrada original.
- \hat{x} es el vector reconstruido por la red.
- $L(x, \hat{x})$ mide cuánta información se perdió durante la codificación y decodificación.

Una pérdida baja indica que la red es capaz de comprimir y reconstruir los datos eficientemente.

C. Ventajas de los Autoencoders

- Permiten una reducción no lineal de la dimensionalidad.
- Se pueden entrenar de forma no supervisada.
- Se adaptan a estructuras complejas de datos.
- Son escalables a grandes volúmenes de datos.

D. Aplicaciones típicas

- Compresión de imágenes y señales.
- Detección de anomalías (outliers).
- Preentrenamiento de redes profundas.
- Generación de datos (en autoencoders variacionales).

IX. MÉTRICAS Y DISTANCIAS

En el contexto del aprendizaje no supervisado, y en particular del clustering, las métricas de evaluación y las medidas de distancia son fundamentales para entender la calidad de los agrupamientos y la similitud entre instancias. A continuación se detallan las métricas más comunes.

A. Métricas de evaluación de clustering

Estas métricas permiten evaluar cuán adecuado es el resultado del algoritmo de clustering, sin necesidad de etiquetas reales (ya que estamos en un contexto no supervisado). Algunas de las más utilizadas son:

- **Silhouette Score:** mide qué tan similar es una muestra a su propio clúster (cohesión) comparado con otros clústeres (separación). Su valor oscila entre -1 y 1. Valores cercanos a 1 indican una mejor agrupación.
- **Davies-Bouldin Index:** evalúa la calidad del clustering basándose en la relación entre la dispersión dentro del clúster y la distancia entre clústeres. Un valor más bajo indica una mejor partición.
- **Calinski-Harabasz Index (también conocido como Variance Ratio Criterion):** compara la varianza entre los clústeres con la varianza interna. Cuanto mayor sea el índice, mejor es la definición del agrupamiento.

B. Distancias comunes

Las medidas de distancia permiten cuantificar la similitud o disimilitud entre dos puntos de datos. Son fundamentales para algoritmos de clustering como *k-means* o *DBSCAN*.

- **Distancia Euclidiana:** es la más común. Calcula la raíz cuadrada de la suma de los cuadrados de las diferencias entre las coordenadas correspondientes:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Se interpreta como la distancia "en línea recta" entre dos puntos en un espacio euclidiano.

- **Distancia Manhattan (o Cityblock):** suma las diferencias absolutas de cada dimensión:

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

Se asemeja a la distancia caminada en una cuadrícula urbana.

- **Distancia del Coseno:** evalúa el ángulo entre dos vectores, útil cuando importa más la dirección que la magnitud:

$$\cos(\theta) = \frac{x \cdot y}{\|x\| \cdot \|y\|}$$

Cuanto más cercano a 1 esté el coseno, más parecidos son los vectores.

- **Distancia de Jaccard:** utilizada para conjuntos, mide la disimilitud como el complemento del índice de Jaccard:

$$d_J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

Ideal para datos binarios o categóricos representados como conjuntos.

Estas distancias influyen significativamente en el comportamiento de los algoritmos de clustering. Por ejemplo, *K-means* asume el uso de distancia euclidiana, lo cual puede ser inadecuado en presencia de variables categóricas o escalas dispares sin normalizar.

X. CONCLUSIONES

El aprendizaje no supervisado representa una de las ramas más fascinantes y útiles dentro del campo de la inteligencia artificial, particularmente en escenarios donde no se dispone de etiquetas o supervisión explícita para los datos. A través de este enfoque, se busca descubrir patrones subyacentes, agrupaciones naturales y estructuras ocultas en los datos, lo cual resulta fundamental en procesos exploratorios, reducción de complejidad y en el diseño de sistemas inteligentes más adaptativos.

A lo largo de este trabajo se han abordado las principales técnicas y herramientas empleadas en aprendizaje no supervisado, entre ellas:

Reducción de dimensionalidad (como PCA y autoencoders), que permiten representar los datos en espacios de menor dimensión conservando la máxima cantidad de información posible. Esta técnica resulta esencial para combatir la maldición de la dimensionalidad, mejorar la visualización de los datos y reducir el costo computacional.

Técnicas de clustering, como K-Means y clustering jerárquico, que permiten organizar datos no etiquetados en grupos significativos según su similitud. Se discutió su funcionamiento, ventajas, desventajas y criterios de convergencia, así como su sensibilidad a la inicialización y a la presencia de outliers.

Autoencoders, una clase de redes neuronales profundas que permiten codificar y decodificar información preservando las características más importantes. Se profundizó en su arquitectura “en forma de sándwich”, su función de pérdida y su capacidad para ser utilizados tanto para reducción de dimensionalidad como para detección de anomalías.

Métricas y funciones de distancia, como la euclidiana, Manhattan, coseno y Jaccard, las cuales permiten cuantificar la similitud entre puntos. Estas distancias son la base de la

mayoría de los algoritmos de agrupamiento y detección de patrones.

Métricas de evaluación de agrupamientos, como el *Silhouette Score*, el índice de *Davies-Bouldin* y *Calinski-Harabasz*, que permiten medir qué tan coherente o útil es una agrupación generada sin supervisión.

Además, se contextualizó esta temática con el avance reciente de modelos open source como DeepSeek 3.1, cuya relevancia en el ámbito actual de la IA no solo radica en su desempeño comparable a modelos privados, sino en su contribución al ecosistema abierto. La disponibilidad de modelos tan potentes permite a investigadores y empresas de menor capacidad en I+D competir, innovar y construir soluciones con mayor rapidez y accesibilidad. Este hito refuerza la importancia de democratizar el acceso a las herramientas de IA, alineándose con los principios fundamentales del aprendizaje no supervisado: exploración, descubrimiento y adaptabilidad sin dependencia de guías externas.