

---

## RESULTADO DE APRENDIZAJE

---

### RdA de la asignatura:

- **RdA 2:** Aplicar modelos de aprendizaje automático supervisado y no supervisado, así como su validación y optimización, en la resolución de problemas tanto reales como simulados.

### Resultados específicos:

- Comprender el funcionamiento del algoritmo k-Means y su pseudocódigo.
- Analizar criterios para la selección del valor de k.
- Explorar métodos derivados del algoritmo k-Means y sus aplicaciones prácticas.
- Implementar el algoritmo k-Means en un entorno práctico, evaluando resultados.

---

## INTRODUCCIÓN

---

**Pregunta inicial:** ¿Qué significa agrupar datos y cómo podríamos identificar patrones en un conjunto de datos sin etiquetas?

---

## DESARROLLO

---

### Actividad 1: Introducción al algoritmo k-Means

#### ¿Cómo lo haremos?

- **Video introductorio:** Visualización del video [K Means Clustering](#), que explica de manera visual y dinámica el concepto y pasos del algoritmo k-Means.
- **Clase magistral:** Presentación de los elementos teóricos principales:
  - Concepto de k-Means.
  - Pseudocódigo del algoritmo.
  - Importancia del valor de k.
- **Discusión:** Reflexión grupal sobre cómo se aplicaría k-Means en problemas reales. Los estudiantes compartirán ideas sobre escenarios prácticos.

**Verificación de aprendizaje:**

- ¿Cuáles son los pasos principales del algoritmo k-Means?
- ¿Qué significa minimizar la inercia en el contexto de k-Means?
- ¿Qué ocurre si seleccionamos un valor incorrecto de k?

**Actividad 2: Implementación práctica de k-Means****¿Cómo lo haremos?**

- **Preparación:** Los estudiantes accederán a un cuaderno de Google Colab preparado para esta actividad. El cuaderno incluye una implementación básica de k-Means con datos simulados.
- **Ejercicio guiado:** Los estudiantes ejecutarán y modificarán el siguiente código base:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

# Generar datos de ejemplo
X, y_true = make_blobs(n_samples=300, centers=4, cluster_std=0.60, random_state=42)

# Visualizar los datos
plt.scatter(X[:, 0], X[:, 1], s=50)
plt.title("Datos de ejemplo")
plt.show()

# Aplicar k-Means
kmeans = KMeans(n_clusters=4, init='k-means++', random_state=42)
kmeans.fit(X)
y_kmeans = kmeans.predict(X)

# Visualizar los clusters
plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50, cmap='viridis')
centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='red', s=200, alpha=0.75, label='Centroides')
plt.title("Clusters encontrados")
plt.legend()
plt.show()
```

- **Experimentación:** Los estudiantes cambiarán los parámetros del algoritmo, como el número de clusters (`n_clusters`) y el método de inicialización (`init`), para observar cómo afectan los resultados.

**EJERCICIO 1.** Modifique el código para:

1. Utilizar  $k = 3$  y comparar el resultado con  $k = 4$ .
2. Cambiar el método de inicialización a 'random' y observar los efectos en la con-

vergencia.

**Verificación de aprendizaje:**

- ¿Cómo cambia la agrupación al modificar  $k$ ?
- ¿Qué ventajas tiene la inicialización `k-means++` sobre `random`?
- ¿Cómo interpretarías los centroides obtenidos?

---

**CIERRE**

---

**Tarea:** Desarrollar los ejercicios planteados en el siguiente cuaderno y entregarlo por el aula virtual:

Enlace al cuaderno: [04-Agrupamiento-kMeans.ipynb](#).

Adicional, visualizar el siguiente video:

Enlace al video: [K-means \(o K-medias\) para detección de Clusters](#).

**Pregunta de investigación:**

1. ¿Qué alternativas existen al algoritmo k-Means y en qué escenarios son más útiles?

Ver estos dos videos: [Clustering with DBSCAN](#); [Identifica Clusters con DBSCAN](#)

2. ¿Cómo se podrían mejorar los resultados de k-Means en presencia de datos con formas no esféricas?
3. ¿Qué otras métricas se pueden usar para evaluar la calidad de los clusters generados?

**Para la próxima clase:** Realizar la actividad de Clase invertida sobre Aprendizaje Supervisado, disponible aquí: [02Est-AprendizajeSupervisado.pdf](#).