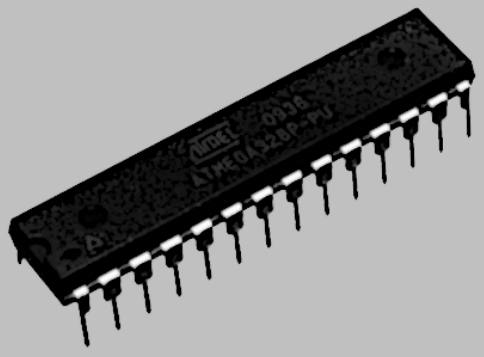

Inteligencia Artificial Embebida

Nombre: _____ Grupo: _____

Dr. Enrique García Trinidad
Tecnológico de Estudios Superiores de Huixquilucan
https://enriquegarcia.xyz
enrique.g.t@huixquilucan.tecnm.mx

Práctica 13: Agrupamiento con el Algoritmo K-Means

1. Introducción Teórica

A diferencia de las prácticas anteriores centradas en el aprendizaje supervisado, el **Algoritmo K-Means** es una técnica de **Aprendizaje No Supervisado**. Esto significa que el algoritmo se utiliza típicamente en entornos de producción para procesar conjuntos de datos sin etiquetas. Su objetivo es encontrar estructuras ocultas agrupando los datos en K clústeres (grupos) distintos basándose en la similitud de sus características (distancia espacial).

El algoritmo funciona inicializando K centroides aleatorios y repitiendo iterativamente dos pasos: asignar cada punto de datos al centroide más cercano, y recalcular la posición de cada centroide como la media de los puntos asignados a él. El proceso se detiene cuando los centroides convergen y ya no cambian de posición. En esta práctica generaremos datos sintéticos artificiales para visualizar cómo K-Means identifica estas agrupaciones automáticamente.

2. Desarrollo de la Práctica e Implementación

2.1. Paso 1: Importar dependencias

Se importa `make_blobs` para generar el conjunto de datos requerido para este experimento, `matplotlib.pyplot` para la visualización de datos en gráficos de dispersión, y el modelo `KMeans` para ajustar los datos basándose en el algoritmo homónimo.

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

2.2. Paso 2: Generar el conjunto de datos

Utilizamos la función `make_blobs()` para generar un conjunto de datos para la tarea de agrupamiento. Esta función devuelve tanto las características generadas (`X`) como las etiquetas originales (`y`). Es importante recalcar que el conjunto generado contiene etiquetas **únicamente para facilitar la comprensión teórica y visualización**, ya que K-Means no las utilizará para entrenar.

Los parámetros utilizados son: `n_samples` (número total de muestras, 500), `n_features` (dimensión de los datos, 2), `centers` (número de categorías o centros, 4), y `random_state` (semilla del generador de números aleatorios para garantizar la reproducibilidad).

```
X, y = make_blobs(n_samples=500, n_features=2, centers=4,
   random_state=1)

print("Dimension de X es {}".format(X.shape))
print("Dimension de y es {}".format(y.shape))
```

2.3. Paso 3: Dibujar gráficos de dispersión

Para entender los datos con los que trabajaremos, primero los graficamos en un diagrama de dispersión simple, y luego utilizamos las etiquetas ocultas `y` generadas por `make_blobs` para colorear los 4 grupos teóricos de los datos originales.

```
# Grafico sin etiquetas (como lo ve el algoritmo)
fig, ax1 = plt.subplots(1)
ax1.scatter(X[:,0], X[:,1], marker="o", s=8)
plt.show()

# Grafico coloreado usando las etiquetas teoricas generadas
color = ["red", "pink", "orange", "green"]
fig, ax1 = plt.subplots(1)

for i in range(4):
    ax1.scatter(X[y==i,0], X[y==i,1], marker="o", s=8, c=color[i])
plt.show()
```

2.4. Paso 4: Realizar el agrupamiento K-Means

Instanciamos el modelo definiendo `n_clusters=3`. Observe que aunque los datos originales tenían 4 centros reales, obligaremos al algoritmo a encontrar solo 3 grupos. Aplicamos el método `.fit(X)` pasándole únicamente la matriz de características (ignorando por completo las etiquetas `y`). Finalmente, extraemos y mostramos las etiquetas predichas almacenadas en el atributo `labels_`.

```
n_clusters = 3
cluster1 = KMeans(n_clusters=n_clusters, random_state=3).fit(X)

# Extraer las etiquetas predichas por el agrupamiento
y_pred1 = cluster1.labels_
print(y_pred1)
```

3. Conclusión

El algoritmo K-Means procesa exitosamente los datos bidimensionales sin utilizar ninguna etiqueta predefinida. Al configurarse con `n_clusters = 3`, el modelo particionó matemáticamente las 500 muestras en los 3 grupos solicitados, asignando a cada muestra una etiqueta predictiva discreta (0, 1 o 2).

4. Evaluación de Comprensión

Instrucciones: Selecciona la respuesta correcta para cada una de las siguientes preguntas.

1. ¿A qué categoría del Machine Learning pertenece el algoritmo K-Means?
    A. Aprendizaje Supervisado, utilizado principalmente para predicción de precios.
    B. Aprendizaje No Supervisado, utilizado para identificar agrupaciones en datos no etiquetados.
    C. Aprendizaje por Refuerzo, donde un agente aprende mediante recompensas.
    D. Regresión Lineal para reducción de dimensionalidad.
2. En la práctica, se utiliza la función `make_blobs`. ¿Cuál es su propósito principal en este entorno?
    A. Entrenar automáticamente el modelo de K-Means.
    B. Limpiar valores nulos de un archivo de texto.
    C. Generar un conjunto de datos sintético (artificial) para simular una tarea de agrupamiento visualizable.
    D. Exportar el modelo entrenado en un archivo PDF.
3. En la definición de `make_blobs`, se utiliza el parámetro `random_state=1`. ¿Qué función cumple este parámetro?
    A. Obliga al algoritmo a que un porcentaje aleatorio de datos se use para validación.
    B. Funciona como una semilla matemática que garantiza que la distribución aleatoria generada sea idéntica cada vez que se ejecute el código.
    C. Indica la cantidad de iteraciones permitidas para encontrar el centroide.
    D. Selecciona aleatoriamente el número de clústeres que buscará el modelo.
4. Si K-Means es un algoritmo no supervisado que procesa datos sin etiquetas, ¿por qué la práctica genera la variable de etiquetas `y`?
    A. Porque K-Means necesita la variable `y` internamente en el comando `.fit(X, y)` para poder funcionar.

B. Exclusivamente con fines académicos y de visualización, para que el estudiante pueda comparar los clústeres reales (teóricos) dibujando los puntos con colores, contra lo que descubre el algoritmo.

C. Porque sin la variable `y`, la biblioteca `matplotlib` arrojaría un error de ejecución.

D. Para medir la ganancia de entropía del modelo.

5. Una vez entrenado el modelo, ¿qué atributo del objeto `KMeans` de `scikit-learn` se utiliza en el código para acceder a las categorías o grupos finales asignados a cada muestra?

   A. `.centers_`

   B. `.predict_`

   C. `.coef_`

   D. `.labels_`