**REPORTE DE PRÁCTICA**

**IDENTIFICACIÓN DE LA PRÁCTICA**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Práctica** | **5** | **Nombre de la práctica** | | **k-means multidimensional** |
| **Fecha** | **07/04/2025** | **Nombre del profesor** | | **Alma Nayeli Rodríguez Vázquez** |
| **Nombre del estudiante** | | | **Jesús Alberto Aréchiga Carrillo** | |

**OBJETIVO**

|  |
| --- |
| El objetivo de esta práctica consiste en implementar el algoritmo k-means multidimensional para clustering. |

**PROCEDIMIENTO**

|  |
| --- |
| Realiza la implementación siguiendo estas instrucciones. |
| Implementa el algoritmo k-means en Python. Para ello, considera los siguientes requerimientos:   1. Utiliza el set de datos “dataSet\_kmeans2D\_iris.txt” 2. Implementa el algortimo de clustering k-means utilizando la función KMeans() de sklearn. 3. Utiliza los siguientes valores para el parámetro n\_clusters:   n\_clusters = 2, n\_clusters = 3 y n\_clusters = 4   1. Dibuja los clusters y centroides con diferentes colores. 2. Reporta el valor de los centroides. Además, reporta el número de datos que tiene cada cluster en cada prueba. |

**IMPLEMENTACIÓN**

|  |
| --- |
| Agrega el código de tu implementación aquí. |
| import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  from sklearn.cluster import KMeans  data = pd.read\_csv('dataSet\_kmeans2D\_iris.csv')  data.head()  x = data.values  plt.plot(x[:,0], x[:,1], 'yo')  media = x.mean(axis=0)  sigma = x.std(axis=0, ddof=1)  x = (x-media)/sigma  model = KMeans(n\_clusters=2)  model.fit(x)  c1 = model.cluster\_centers\_[0,:]  c2 = model.cluster\_centers\_[1,:]  c1, c2  cluster1 = np.argwhere = (model.labels\_ == 0)  cluster2 = np.argwhere = (model.labels\_ == 1)  plt.plot(x[cluster1,0], x[cluster1,1], 'yo')  plt.plot(x[cluster2,0], x[cluster2,1], 'cs')  plt.plot(c1[0], c1[1], 'ro')  plt.plot(c2[0], c2[1], 'bo')  plt.show()  model = KMeans(n\_clusters=3)  model.fit(x)  c1 = model.cluster\_centers\_[0,:]  c2 = model.cluster\_centers\_[1,:]  c3 = model.cluster\_centers\_[2,:]  c1, c2, c3  cluster1 = np.argwhere = (model.labels\_ == 0)  cluster2 = np.argwhere = (model.labels\_ == 1)  cluster3 = np.argwhere = (model.labels\_ == 2)  plt.plot(x[cluster1,0], x[cluster1,1], 'yo')  plt.plot(x[cluster2,0], x[cluster2,1], 'cs')  plt.plot(x[cluster3,0], x[cluster3,1], 'gd')  plt.plot(c1[0], c1[1], 'ro')  plt.plot(c2[0], c2[1], 'bo')  plt.plot(c3[0], c3[1], 'mo')  plt.show()  model = KMeans(n\_clusters=4)  model.fit(x)  c1 = model.cluster\_centers\_[0,:]  c2 = model.cluster\_centers\_[1,:]  c3 = model.cluster\_centers\_[2,:]  c4 = model.cluster\_centers\_[3,:]  c1, c2, c3, c4  cluster1 = np.argwhere = (model.labels\_ == 0)  cluster2 = np.argwhere = (model.labels\_ == 1)  cluster3 = np.argwhere = (model.labels\_ == 2)  cluster4 = np.argwhere = (model.labels\_ == 3)  plt.plot(x[cluster1,0], x[cluster1,1], 'yo')  plt.plot(x[cluster2,0], x[cluster2,1], 'cs')  plt.plot(x[cluster3,0], x[cluster3,1], 'gd')  plt.plot(x[cluster4,0], x[cluster4,1], 'k\*')  plt.plot(c1[0], c1[1], 'ro')  plt.plot(c2[0], c2[1], 'rs')  plt.plot(c3[0], c3[1], 'rd')  plt.plot(c4[0], c4[1], 'r\*')  plt.show() |

**RESULTADOS**

Agrega la(s) imagen(es) con los resultados obtenidos en los espacios indicados.

|  |  |
| --- | --- |
| Gráfica de los datos agrupados para n\_clusters = 2 | Valores de los centroides y cantidad de datos en cada cluter para n\_clusters = 2 |
|  |  |
| Gráfica de los datos agrupados para n\_clusters = 3 | Valores de los centroides y cantidad de datos en cada cluter para n\_clusters = 3 |
|  |  |
| Gráfica de los datos agrupados para n\_clusters = 4 | Valores de los centroides y cantidad de datos en cada cluter para n\_clusters = 4 |
|  |  |

**CONCLUSIONES**

|  |
| --- |
| Escribe tus observaciones y conclusiones. |
| El algoritmo de kmeans nos sirve para agrupar un grupo muy grande de datos en diferentes grupos (o clusters).  Una limitación del método kmeans() de sklearn es que, para ciertos grupos de datos, no se separan de la manera más correcta, para eso hay que implementar el algoritmo manualmente. |