

Curso de IA

# Capítulo 6. Cuestionario

Para estudiantes

© 2023 SAMSUNG. Todos los derechos reservados.

La Oficina de Ciudadanía Corporativa de Samsung Electronics posee los derechos de autor de este documento.

Este documento es una propiedad literaria protegida por la ley de derechos de autor, por lo que está prohibida su reimpresión y reproducción sin permiso.

Para utilizar este documento fuera del plan de estudios de Samsung Innovation Campus, debe recibir el consentimiento por escrito del titular de los derechos de autor..

1. ¿Cuál de las siguientes declaraciones sobre el aprendizaje es incorrecta?
    - 1 El aprendizaje supervisado utiliza tanto datos de entrada como datos de aprendizaje dado un resultado, pero el aprendizaje no supervisado se utiliza en forma de aprendizaje utilizando sólo datos de entrada sin datos de aprendizaje.
    - 2 Se puede decir que el aprendizaje no supervisado es un método de aprendizaje que informa del problema pero no dice la respuesta.
    - 3 Como algoritmo de aprendizaje no supervisado, el análisis de agrupación jerárquica y la agrupación de K-means se pueden utilizar para la agrupación.
    - 4 La agrupación de K-means tiene la ventaja de ser fácil de determinar el número inicial de agrupaciones y fácil de interpretar los resultados.
  
  2. ¿Cuál de las siguientes declaraciones sobre la agrupación es incorrecta?
    - 1 La técnica de agrupamiento más utilizada es K-Means.
    - 2 La agrupación jerárquica se decide después de la agrupación sin determinar el número de grupos.
    - 3 El algoritmo de la agrupación jerárquica tiene K-Medoides.
    - 4 La agrupación es una técnica que asigna y analiza grupos basados en la similitud o disimilitud entre cada entidad.
  
  3. El algoritmo EM pertenece a los algoritmos codiciosos. Explique la razón.
  
  
  
  
  
  
  
  
  
  
  4. Escriba tres o más ejemplos de algoritmo de agrupación. (por ejemplo, análisis de datos)
- Análisis de agrupamiento, agrupación jerárquica, agrupación no jerárquica.

5. Explique los dos métodos para elegir el número óptimo de conglomerados cuando se utiliza k-means. Describa cómo encontrar el número óptimo de conglomerados utilizando los dos métodos.

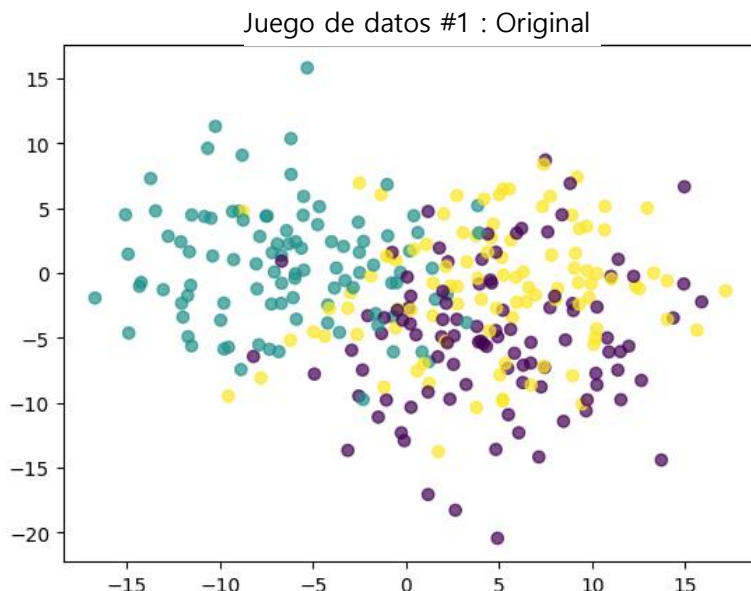
Estándar: A medida que aumentamos el número de conglomerados, la varianza dentro de cada conglomerado disminuye. Sin embargo, esta disminución se vuelve menos pronunciada a partir de un cierto valor de K. Este punto de inflexión, donde la disminución de la varianza se vuelve menos marcada.

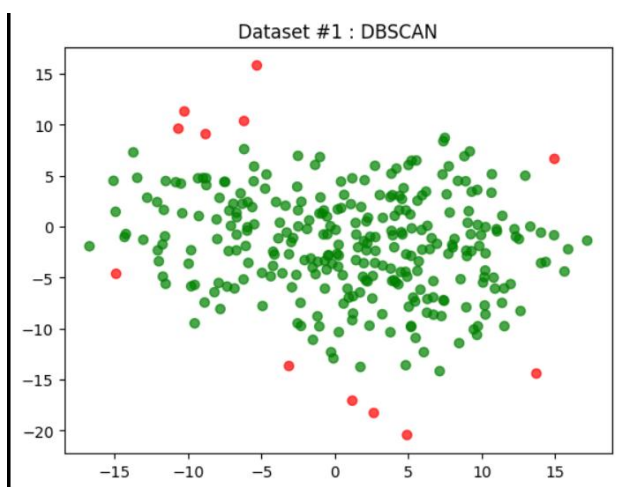
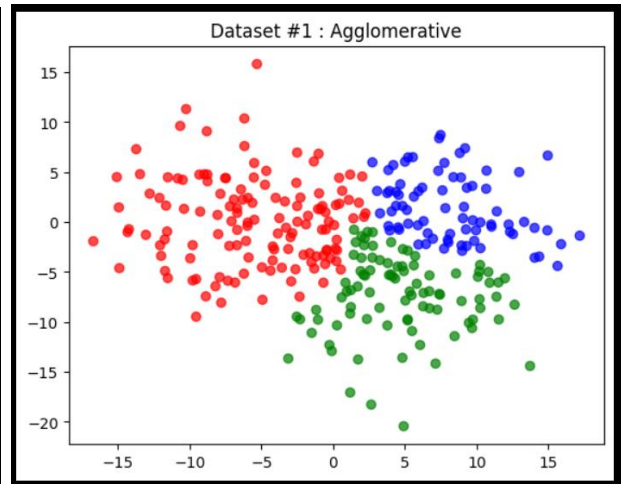
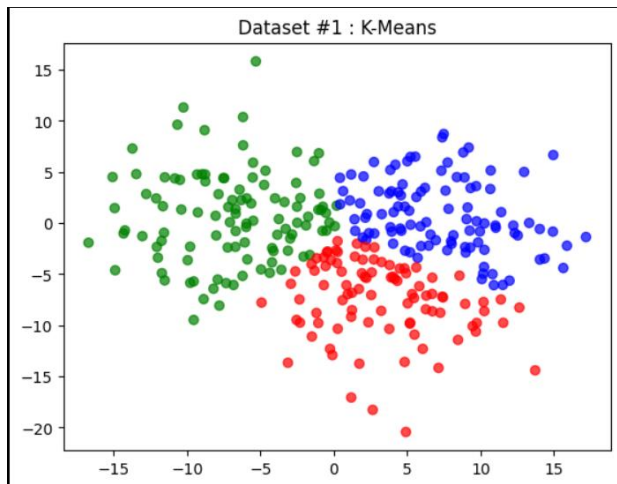
DBSCAN: Es basado en densidad que identifica clusters de diferentes formas y tamaños en conjuntos de datos con ruido. A diferencia de K-Means, no requiere especificar el número de clusters previamente.

6. Agrupe y visualice los datos de muestra del código siguiente utilizando los algoritmos KMeans, Agrupamiento Aglomerativo y DBSCAN. Sin embargo, para K-Means y los agrupamientos aglomerativos, establecer el número de grupos a 3.

```
desde sklearn.datasets importar make_blobs, make_moons
desde sklearn.cluster importar KMeans
desde sklearn.cluster importar AgglomerativeClustering
desde sklearn.cluster importar DBSCAN
importar pandas como pd
importar matplotlib.pyplot como plt

X, label = make_blobs(n_samples=300, n_features=2, centers=3, cluster_std=5, random_state=123)
plt.scatter(X[:,0], X[:,1], c=label, alpha=0.7)
plt.title('Dataset #1 : Original')
plt.show()
```



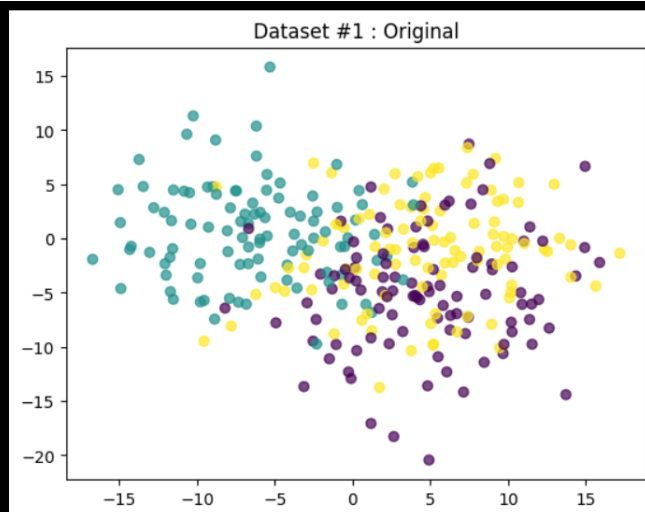


```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
from sklearn.datasets import make_blobs, make_moons
from sklearn.cluster import KMeans
from sklearn.cluster import AgglomerativeClustering
from sklearn.cluster import DBSCAN
from scipy.cluster.hierarchy import linkage, dendrogram, fcluster
```

✓ 0.0s

```
# Dataset #1.
X1, label1 = make_blobs(n_samples=300, n_features=2, centers=3, cluster_std = 5, random_state=123)
plt.scatter(X1[:,0],X1[:,1], c= label1, alpha=0.7 )
plt.title('Dataset #1 : Original')
plt.show()
```

✓ 0.0s



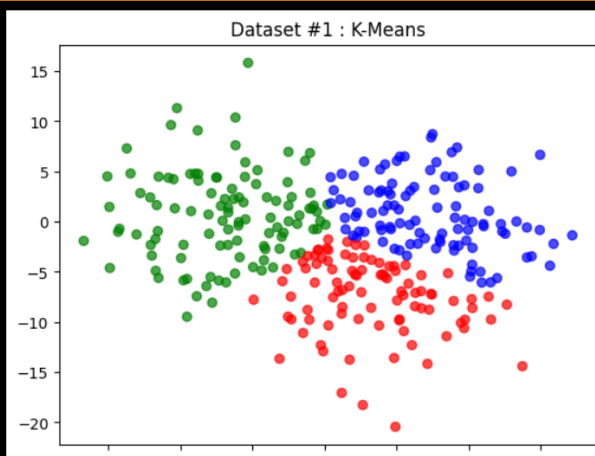
kmeans 3 clusters

kmeans 3 clusters

```
# Dataset #1 and 3 clusters.
kmeans = KMeans(n_clusters=3, random_state=123, n_init='auto') # kmeans object for 2 clusters. random_state=123 means deterministic in
kmeans.fit(X1) # Unsupervised learning => Only X1.
myColors = {0: 'red', 1: 'green', 2: 'blue'} # Define a color palette: 0~1.
plt.scatter(X1[:,0], X1[:,1], c= pd.Series(kmeans.labels_).apply(lambda x: myColors[x]), alpha=0.7 )
plt.title('Dataset #1 : K-Means')
plt.show()
```

[16] ✓ 0.1s

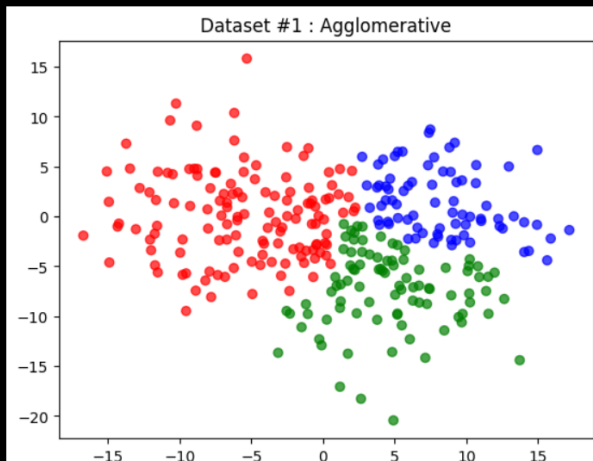
Python



## Ageupamiento Aglomerativo 3 clusters

```
# Dataset #1 and 3 clusters.
agglo = AgglomerativeClustering(n_clusters=3)
agglo.fit(X1)
myColors = {0: 'red', 1: 'green', 2: 'blue'} # Define a color palette: 0~1.
plt.scatter(X1[:,0],X1[:,1], c= pd.Series(agglo.labels_).apply(lambda x: myColors[x], alpha=0.7 )
plt.title('Dataset #1 : Agglomerative')
plt.show()
```

[37] ✓ 0.0s Python



## DBSCAN

```
# Dataset #1.
dbscan = DBSCAN(eps=3, min_samples=5)
dbscan.fit(X1)
myColors = {-1: 'red', 0: 'green', 1: 'blue'} # Define a color palette: -1~1. Red = -1 = outlier.
plt.scatter(X1[:,0],X1[:,1], c= pd.Series(dbscan.labels_).apply(lambda x: myColors[x], alpha=0.7 )
plt.title('Dataset #1 : DBSCAN')
plt.show()
```

[38] ✓ 0.0s Python

