UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

Práctica Minería de Datos

CURSO:

Minería de Datos

DOCENTE:

Msc. Ulises Roman Concha

GRUPO 02

INTEGRANTES:

Carlos Alexander Paredes Contreras	19200057
César Gabriel Urquizo Espinoza	19200048
Victor Alfonso Ochoa Flores	20200127
Anderson Carlos Minetto Mori	20200076
Jiménez Taipe César Stephano	19200169
Joseph Enrique Guadalupe Poma	19200229

LIMA, PERÚ

2023

ÍNDICE

1. Pregunta 2: Modelos de Agrupamiento (DBSCAN, k-means)

Pregunta 2.1

Compare y contraste cómo DBSCAN y k-means manejan la presencia de ruido en los datos y su capacidad para identificar grupos de diferentes formas y tamaños.

a. Manejo de ruido en los datos:

- DBSCAN: Tiene la capacidad de identificar puntos atípicos (ruido) como parte de su proceso, clasificándolos como puntos de ruido si no pertenecen a ningún grupo.
- k-means: Es sensible al ruido y puede asignar puntos a un clúster incluso si no pertenecen claramente a ninguno, ya que busca minimizar la distancia al centroide más cercano.

b. Capacidad para identificar grupos de diferentes formas y tamaños:

- DBSCAN: Es capaz de identificar grupos de formas y tamaños arbitrarios. Define los clústeres basándose en la densidad de los puntos y puede manejar clústeres de formas irregulares.
- k-means: Tiende a encontrar clústeres de formas geométricas (esféricos si se asume una varianza igual) y puede ser menos efectivo para identificar grupos de formas no convencionales.

Pregunta 2.2

Caso de Estudio: Trabaja para una compañía de servicios de entrega y tiene datos de ubicaciones de entregas pasadas. Diseñe un escenario donde DBSCAN podría identificar áreas geográficas con alta densidad de entregas y explique cómo utilizaría los resultados para mejorar las rutas de entrega.

DBSCAN podría ser útil en la identificación de áreas geográficas con alta densidad de entregas en la siguiente situación:

a. Escenario: Tienes datos de ubicaciones de entregas pasadas.

b. Uso de DBSCAN: DBSCAN podría identificar conglomerados (clusters) geográficos basados en la densidad de entregas. Por ejemplo, podría identificar áreas urbanas densamente pobladas donde se realizan más entregas.

c. Mejora en las rutas de entrega: Utilizar los resultados de DBSCAN permitiría optimizar las rutas de entrega al:

   - Establecer centros logísticos cerca de estas áreas densas para una distribución más eficiente.
   - Planificar rutas de entrega más efectivas al concentrar o distribuir los recursos logísticos según la densidad de entregas en esas áreas identificadas.

Pregunta 2.3

Escriba una parte del código en Python para implementar k-means en el escenario propuesto en la pregunta 2.2. Justifique la elección del número de clústeres.

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
    //Supongamos que tenemos un DataFrame 'delivery_data' con las ubicaciones de entregas pasadas
    //Elegimos el número de clústeres
    // Se puede utilizar el método del codo para encontrar un número óptimo de clústeres
inertia = []
for k in range(1, 11):
        kmeans = KMeans(n_clusters=k, random_state=42)
        kmeans.fit(delivery_data)
        inertia.append(kmeans.inertia_)
plt.plot(range(1, 11), inertia)
```

```
plt.xlabel('Number of clusters')

plt.ylabel('Inertia')

plt.title('Elbow Method')

plt.show()

        // Basado en el codo en el gráfico, elige el número óptimo de clústeres

optimal_clusters = 3  # Por ejemplo, elegimos 3 clústeres

        // Aplicar k-means con el número óptimo de clústeres

kmeans = KMeans(n_clusters=optimal_clusters, random_state=42)

kmeans.fit(delivery_data)

        //Visualizar los clústeres resultantes si es posible (dependiendo de las dimensiones de los datos)

        // Por ejemplo, si los datos son 2D, se puede visualizar utilizando un scatter plot

plt.scatter(delivery_data[:, 0], delivery_data[:, 1], c=kmeans.labels_, cmap='viridis')

plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], marker='x', c='red', s=200)

plt.xlabel('X-coordinate')

plt.ylabel('Y-coordinate')

plt.title('Clustering of Delivery Locations')

plt.show()
```

Pregunta 2.4

¿Cómo podría adaptar DBSCAN para manejar conjuntos de datos con atributos de diferentes escalas? Proporcione una estrategia y explique su razonamiento.

Para manejar conjuntos de datos con atributos de diferentes escalas en DBSCAN, es recomendable realizar la estandarización o normalización de los datos antes de aplicar el algoritmo. Esto garantiza que las características con escalas diferentes no dominen el cálculo de la distancia.

Utilizar técnicas como la estandarización (restar la media y dividir por la desviación estándar) o la normalización (escalar los datos al rango [0, 1]) antes de aplicar DBSCAN. Esto asegura que todas las características contribuyan por igual a la medida de distancia y evita que una característica con mayor escala domine sobre otras.