LABORATORIO 05

Detalle de lo trabajado:

Punto1

El código del punto 1, implementa una simulación de algoritmos de *clustering* utilizando KMeans y MiniBatchKMeans de la biblioteca scikit-learn, con el objetivo de analizar cómo agrupan datos sintéticos generados artificialmente. El número de clusters se determina aleatoriamente entre 1 y 20, y luego se crean datos usando una función personalizada (generate_custom_dataset) que garantiza que los centroides estén bien separados. Los datos se distribuyen alrededor de estos centroides con algo de dispersión (desviación estándar), y se dividen en un conjunto de entrenamiento y uno de prueba.

Para visualizar los resultados, el código incluye funciones como plot_data (para mostrar los puntos), plot_centroids (para marcar los centros de los clusters), plot_clusters (para mostrar la asignación de cada punto a un cluster), y plot_decision_boundaries (para visualizar las fronteras que separan los clusters). Estas herramientas gráficas ayudan a observar el comportamiento de los algoritmos al agrupar los datos y a identificar diferencias visuales entre KMeans y MiniBatchKMeans en cuanto a precisión y velocidad.

Por último, la función plot_clusterer_comparison permite realizar una comparación directa entre ambos algoritmos usando los mismos datos. Se grafican los resultados obtenidos por cada uno, junto con los centroides detectados y las fronteras de decisión. Esto permite evaluar visualmente cuál de los algoritmos se ajusta mejor a los datos, facilitando el análisis y comprensión del funcionamiento del clustering no supervisado.

Punto2

Para el aprendizaje semi-supervisado y activo se utilizó el siguiente dataset:

Nombre del Dataset:

- CIFAR-10

Link:

- https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html?utm_source=chatgpt.com

Filas: 20

Columnas: sin estructura

Descripción:

- El 20 Newsgroups es un dataset de texto ampliamente utilizado en tareas de procesamiento de lenguaje natural. Contiene cerca de 20.000 mensajes de foros organizados en 20 categorías temáticas (como política, ciencia, informática, religión, deportes, etc.). Cada documento representa un mensaje de discusión, permitiendo realizar clasificación de texto, clustering y experimentos de aprendizaje supervisado, semi-supervisado o activo.

Punto3

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Este método agrupa datos basándose en la densidad, identificando clusters como regiones con alta concentración de puntos y marcando como anomalías aquellos que quedan aislados. En el código, primero se cargaron las imágenes del dataset Olivetti Faces y se redujo su dimensionalidad usando PCA para visualizar los datos en 2D. Luego, se aplicó DBSCAN con parámetros predefinidos (eps=0.5 y min_samples=5), que determinan la distancia máxima entre puntos y el número mínimo de muestras para formar un cluster, respectivamente. La función plot_dbscan graficó los resultados, diferenciando entre puntos centrales de clusters (marcados con asteriscos), puntos periféricos (puntos pequeños) y anomalías (cruces rojas). Adicionalmente, se incluyó una visualización de las caras representativas de cada cluster, organizadas en una cuadrícula adaptable dinámicamente para evitar errores de dimensiones. Este enfoque permite identificar grupos naturales de rostros y detectar posibles outliers.

Gaussian Mixtures (Mezclas Gaussianas)

Este modelo asume que los datos provienen de una combinación de distribuciones Gaussianas, cada una con sus propios parámetros (media, covarianza y peso). En el código, se utilizó PCA para reducir las dimensiones de las imágenes y luego se ajustó un modelo de Gaussian Mixture con 10 componentes (n_components=10). La función plot_gaussian_mixture visualizó las regiones de densidad de probabilidad, con contornos que representan los niveles de densidad y líneas discontinuas rojas que delimitan los clusters. También se implementó detección de anomalías marcando las muestras con densidades inferiores al percentil 4. Finalmente, se mostraron las caras identificadas como anomalías, lo que ayuda a entender qué imágenes el modelo considera atípicas. Para mejorar la selección del número de clusters, se sugirió usar BayesianGaussianMixture, que asigna pesos cercanos a cero a los componentes innecesarios, evitando así sobreajuste.