Inteligencia Artificial
Proyecto 2: Clustering

*Universidad de Ingeniería y Tecnología UTEC - Curso: Inteligencia Artificial

1st Vasquez Auqui, Fabrizio (100%)
201810423
Computer Science
Lima, Perú
fabrizio.vasquez@utec.edu.pe

2nd Rubio Montiel, Ignacio (100%)
201910423
Computer Science
Lima, Perú
ignacio.rubio@utec.edu.pe

I. Introducción

El clustering es una técnica de aprendizaje no supervisado ampliamente utilizada en diversas áreas, como la minería de datos, la inteligencia artificial y el análisis de datos. Consiste en agrupar un conjunto de datos en subconjuntos más pequeños y coherentes, conocidos como clústeres, en función de su similitud. En este informe, se presentarán tres métodos de clustering comunes: GMM (Gaussian Mixture Model), K-Means y DBSCAN.

La proporción de imagenes de la carpeta que contiene las imágenes de las siguientes emociones: *happy*,*contempt*, *fear*,*surprise*,*sadness*,*anger*,*disgust*

Emoción/carpeta	%imagenes
surprise	25.38
happy	21.10
disgust	18.04
anger	13.76
sadness	8.56
fear	7.65
contempt	5.50

TABLE I: Proporción de imágenes de la base de datos de imágenes de emociones.

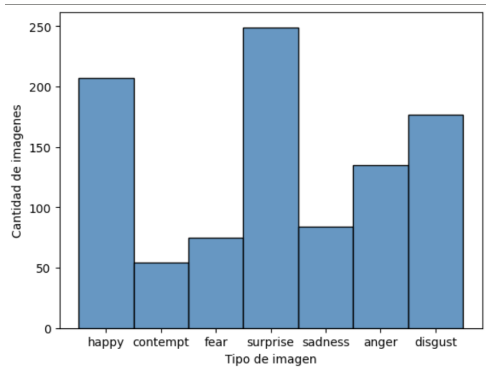

Fig. 1: Distribución de imágenes de la base de datos.

II. Métodos de Clustering

A. GMM (Gaussian Mixture Model)

El método de clustering GMM se basa en la suposición de que los datos pertenecen a una mezcla de distribuciones gaussianas. Se asume que cada clúster sigue una distribución gaussiana y que los datos se generan mediante una combinación lineal de estas distribuciones. El objetivo del algoritmo GMM es encontrar los parámetros óptimos de las distribuciones gaussianas y las ponderaciones de cada clúster, de manera que maximice la verosimilitud de los datos observados. La asignación de puntos a un clúster específico se realiza en función de las probabilidades posteriores calculadas para cada clúster.

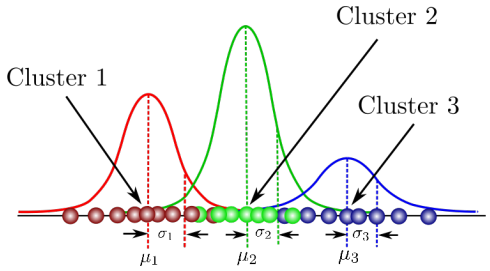

Fig. 2: Clusteres encontrados por un algoritmo GMM

B. K-Means

El algoritmo K-Means es uno de los métodos de clustering más utilizados. Este método agrupa los datos en K clústeres, donde K se define de antemano. El algoritmo K-Means asigna inicialmente K centroides aleatorios y luego itera hasta converger a una solución estable. En cada iteración, los puntos se asignan al centroide más cercano y luego se recalculan los centroides en función de los puntos asignados. Este proceso se repite hasta que los centroides ya no cambien significativamente. La elección inicial de los centroides puede afectar los resultados del algoritmo, por lo que es común ejecutar el algoritmo varias veces con diferentes inicializaciones. Para nuestra implementación, utilizamos KD tree para inicializar los

clusters. En general, la inicialización de los clusters con KD tree en el algoritmo puede mejorar la eficiencia, la calidad de la solución inicial, la estabilidad y la resistencia a los valores atípicos.

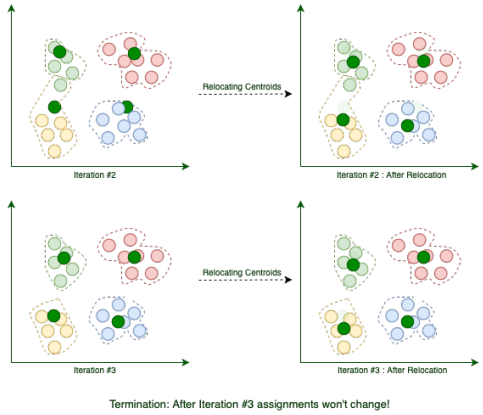

Fig. 3: Ajuste de centroide en algoritmo K-means

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

El algoritmo DBSCAN se basa en la idea de densidad. En lugar de definir clústeres en términos de centroides o distribuciones gaussianas, DBSCAN agrupa los datos en función de la densidad local. Cada punto se clasifica como núcleo, borde o ruido según el número de puntos cercanos dentro de un radio especificado. Los puntos de núcleo que están lo suficientemente cerca se agrupan en el mismo clúster, mientras que los puntos de borde se agregan a los clústeres existentes si están lo suficientemente cerca. Los puntos de ruido no pertenecen a ningún clúster. La elección del radio de vecindad y el número mínimo de puntos requeridos son consideraciones importantes al aplicar DBSCAN.

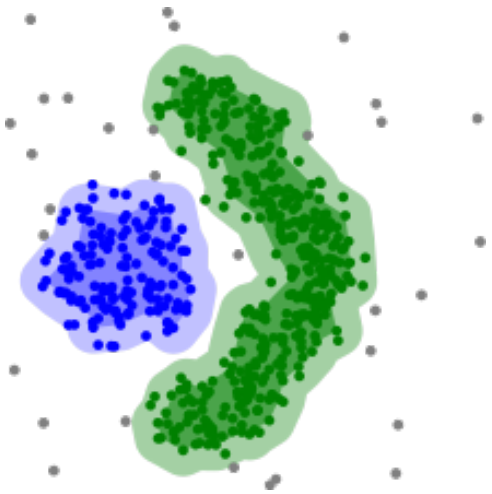

Fig. 4: Clusters encontrados utilizando algoritmo DBSCAN

III. Consideraciones en la Generación de Vectores Característicos

A. Wavelets

Utilizamos PyWavelets, una biblioteca de Python para el análisis de señales. El número de cortes o niveles de descomposición es una consideración importante. Los niveles de descomposición determinan cuántas veces se aplicará la transformada wavelet a la señal original para obtener los coeficientes de detalle y aproximación en cada nivel. Variamos la cantidad de cortes entre 2 y 4 para nuestra experimentación.

B. PCA (Principal Component Analysis)

PCA es un método estadístico que utiliza una transformación ortogonal para convertir un conjunto de observaciones de variables posiblemente correlacionadas en un conjunto de valores de variables linealmente no correlacionadas llamadas componentes principales. Esta transformación se define de tal manera que el primer componente principal tiene la mayor varianza posible (es decir, representa la mayor parte de la variabilidad en los datos), y cada componente sucesivo a su vez tiene la mayor varianza posible bajo la restricción de que sea ortogonal a los componentes precedentes.

Usaremos PCA para lidiar con la maldición de la dimensionalidad, ya que permite reducir la cantidad de variables que estás utilizando para describir tus datos sin perder demasiada información.

IV. Métodología

La metodología planteada para la obtención de resultados de cada modelo se planteo en el siguiente orden. En ese sentido, la evaluación de los modelos será en el siguiente orden: K-means, DBSCAN, GMM.

Primero haremos pruebas con cortes de la imagen para la obtención de vectores característicos mediante *pywaveletes* con el método 'haar' con cortes iguales a $2, 3, 4$ lo que significaría en una reducción de la dimensionalidad de los vectores carácteristicos en la siguiente proporción $dim_n(picure)/4^cortes$. Como hipótesis inicial planteamos que a una menor cantidad de dimensionalidad de los vectores mejor podrían evidenciarse la presencia de clusters. Cada una de las pruebas se realizaron planteando testing para un número de clusters iguales a $k = 2, 3, 4, 5, 6, 7$. Mencionar que para las pruebas aquellas filas que se encuentren con (S) con aquellos modelos usados de la librería de *sklearn* y (I) la implementanda por nosotros.

V. Resultados

A. Experimentos para cada modelo

Clusters	Silhouette	Homogeneity	Rand	Adjust Rand	Adjust Mutual Information
k=2 (S)	0.189676	0.007124	0.497489	0.003833	0.007926
k=2 (I)	0.189784	0.007814	0.497169	0.003842	0.008933
k=3 (S)	0.218303	0.031324	0.609378	0.019086	0.035232
k=3 (I)	0.218303	0.031324	0.609378	0.019086	0.035232
k=4 (S)	0.19894	0.040262	0.652787	0.031617	0.041086
k=4 (I)	0.19894	0.040262	0.652787	0.031617	0.041086
k=5 (S)	0.185943	0.043023	0.700354	0.034241	0.039048
k=5 (I)	0.18578	0.044798	0.700514	0.034946	0.040964
k=6 (S)	0.196134	0.050708	0.704772	0.020129	0.043892
k=6 (I)	0.17355	0.048049	0.718869	0.036612	0.040701
k=7 (S)	0.204282	0.056741	0.727223	0.023782	0.046178
k=7 (I)	0.181494	0.053924	0.730768	0.036609	0.043401

TABLE II: K-Means. El resultado según las siguientes métricas cuando el número de cortes es igual a 2, variando K

Clusters	Silhouette	Homogeneity	Rand	Adjust Rand	Adjust Mutual Information
k=2(S)	0.336173	0.009283	0.500447	0.005789	0.011059
k=2(I)	0.336265	0.009011	0.499965	0.005325	0.010665
k=3(S)	0.350402	0.018401	0.583189	0.014176	0.019319
k=3(I)	0.230974	0.013046	0.606794	0.016436	0.0122
k=4(S)	0.283296	0.023264	0.664593	0.02386	0.020916
k=4(I)	0.283606	0.023264	0.664522	0.023815	0.020918
k=5(S)	0.26297	0.026721	0.693701	0.01224	0.021465
k=5(I)	0.261173	0.037096	0.69126	0.031702	0.032927
k=6(S)	0.26206	0.03579	0.705197	0.016447	0.028375
k=6(I)	0.261758	0.03426	0.705007	0.015997	0.026787
k=7(S)	0.204282	0.056741	0.727223	0.023782	0.046178
k=7(I)	0.181494	0.053924	0.730768	0.036609	0.043401

TABLE III: K-Means. El resultado según las siguientes métricas cuando el número de cortes es igual a 3, variando K

Clusters	Silhouette	Homogeneity	Rand	Adjust Rand	Adjust Mutual Information
k=2(S)	0.525843	0.005902	0.471015	0.001568	0.006242
k=2(I)	0.525843	0.005902	0.471015	0.001568	0.006242
k=3(S)	0.585789	0.023171	0.510171	0.001598	0.026745
k=3(I)	0.414453	0.011625	0.595706	0.010572	0.010486
k=4(S)	0.489429	0.028894	0.634862	0.011567	0.027992
k=4(I)	0.351709	0.021777	0.659319	0.019135	0.019273
k=5(S)	0.508766	0.036751	0.640658	0.004911	0.034058
k=5(I)	0.45376	0.034906	0.691046	0.015465	0.030387
k=6(S)	0.516549	0.041645	0.649161	0.006824	0.036512
k=6(I)	0.408788	0.047512	0.706256	0.018367	0.040468
k=7(S)	0.485092	0.047876	0.705157	0.010083	0.038354
k=7(I)	0.409343	0.047286	0.723339	0.014037	0.036931

TABLE IV: K-Means. El resultado según las siguientes métricas cuando el número de cortes es igual a 4, variando K

Clusters	Silhouette	Homogeneity	Rand	Adjust Rand	Adjust Mutual Information
k=2(S)	0.188958	0.008229	0.496794	0.004107	0.009541
k=3(S)	0.218317	0.031716	0.60954	0.019613	0.035727
k=4(S)	0.182846	0.035958	0.651476	0.018748	0.035961
k=5(S)	0.197881	0.042199	0.669047	0.018313	0.039197
k=6(S)	0.185673	0.050138	0.698246	0.021592	0.043673
k=7(S)	0.204286	0.055065	0.727121	0.023441	0.044516

TABLE V: GMM. El resultado según las siguientes métricas cuando el número de cortes es igual a 2, variando K

Clusters	Silhouette	Homogeneity	Rand	Adjust Rand	Adjust Mutual Information
k=2(S)	0.327049	0.006855	0.504225	0.007945	0.007512
k=3(S)	0.194134	0.016734	0.60208	0.009423	0.016884
k=4(S)	0.27819	0.022082	0.663677	0.022493	0.019565
k=5(S)	0.260996	0.029187	0.681899	0.027439	0.024638
k=6(S)	0.232732	0.046607	0.721415	0.022114	0.043673
k=7(S)	0.239741	0.057569	0.732876	0.028063	0.04677

TABLE VI: GMM. El resultado según las siguientes métricas cuando el número de cortes es igual a 3, variando K

Min_samples	Silhouette	Homogeneity	Rand	Adjust Rand	Adjust Mutual Information
2(S)	-0.365799	0.174124	0.377923	0.021154	0.099578
2(I)	-0.365799	0.174124	0.377923	0.021154	0.099578
3(S)	-0.448739	0.030367	0.213967	0.003773	0.016884
3(I)	-0.448739	0.030367	0.213967	0.003773	0.019565

TABLE VII: DBSCAN. El resultado según las siguientes métricas cuando el número de cortes es igual a 3, variando K

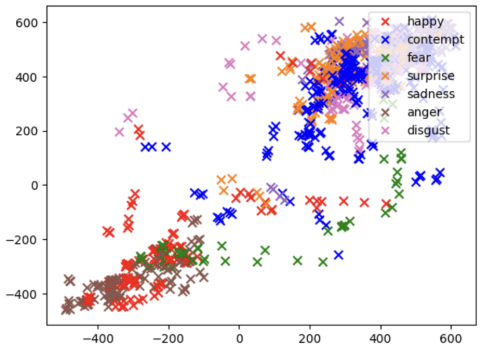

Fig. 5: GMM con 2 cortes

VI. Conclusiones

El mejor algoritmo en terminos de las metricas evaluadas es el GMM con 7 clusters. Según lo obtenido, la implementación de dbscan se asemeja bastante al de la librería *sklearn* por las métricas obtenidas. Las métricas de RandIndex ajustado nos provee de mayor fiabilidad junto con el ajustado para

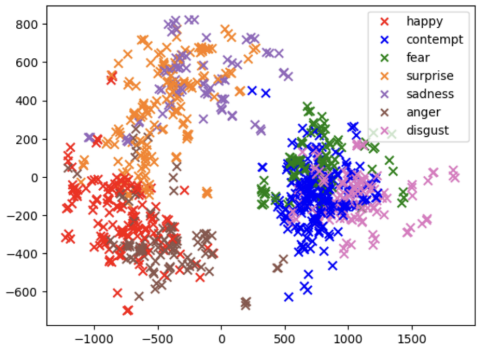

Fig. 6: GMM con 3 cortes

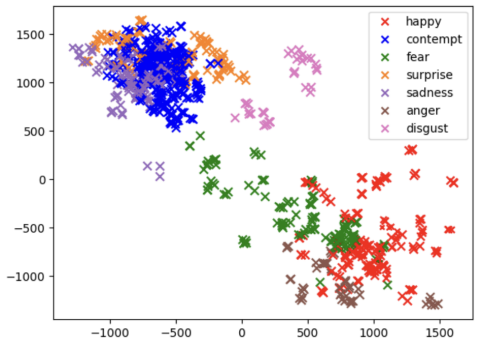

Fig. 7: GMM con 4 cortes

poder indicar lo anterior, aunque en ciertos escenarios bajo 3 clusteres el kmeans funciona relativamente mejor.

VII. Repositorio

Incluimos el enlace al entorno en donde ejecutamos la experimentación:

Proyecto 2

VIII. Referencias

Referencias Bibliográficas

[1] Ravihara, R. (2023, January 10). Gaussian Mixture Model Clearly Explained. Towards Data Science. Retrieved May 29, 2023, from https://towardsdatascience.com/gaussian-mixture-model-clearly-explained-115010f7d4cf

[2] Yıldırım, S. (2020, March 3). K-Means Clustering — Explained. Towards Data Science. Retrieved May 29, 2023, from https://towardsdatascience.com/k-means-clustering-explained-4528df86a120

[3] Yıldırım, S. (2020, April 22). DBSCAN Clustering — Explained. Towards Data Science. Retrieved May 29, 2023, from https://towardsdatascience.com/dbscan-clustering-explained-97556a2ad556