Agrupación de canciones aplicando K-means sobre un FMA Dataset.

Universidad de Cuenca
Facultad de Ingeniería, Escuela de Sistemas
Abad Freddy, Cabrera Edwin, Cárdenas Paola.
freddy.abadl@ucuenca.ec, edwin.cabrera@ucuenca.ec , paola.cardenas0108@ucuenca.ec

ABSTRACT
This work describes the categorization of songs by applying the K-Means tools, Hierarchical Trees and a small introduction to the DBSCAN. These tools used in Machine Learning, have their advantages and disadvantages, that is why this report seeks to make a comparison, obtaining results from the 3 tools and analyze which is the best for FMA Dataset.

Keywords: K-means, performance, trees, clustering.

RESUMEN
Este trabajo describe la categorización de canciones aplicando las herramientas K-Means, Árboles Jerárquicos y una pequena introduccion a los DBSCAN. Estas herramientas empleadas en Machine Learning, tienen sus ventajas y desventajas, es por esto que este informe busca hacer una comparativa, obteniendo resultados de las 3 herramientas y analizar cual es la mejor para FMA Dataset.

Palabras clave: K-means, rendimiento, arboles, clustering.

1. INTRODUCCIÓN

El clustering denota el agrupamiento de un conjunto de observaciones en subconjuntos (clústeres) para que observaciones en el mismo clúster sean similares en algunos sentido. Así este es un método de aprendizaje no supervisado, y una técnica común para análisis estadístico de datos utilizado en muchos campos. Es así que se tienen los métodos jerárquicos que tienen por objetivo agrupar clusters para formar uno nuevo o bien separar alguno ya existente para dar origen a otros dos, de tal forma que, si efectúa sucesivamente este proceso de aglomeración o división, se minimice alguna distancia o bien se maximice alguna medida de similitud. Además de estos, se tienen los K-means clustering los cuales son algoritmos para clasificar o agrupar objetos basados en atributos en K número de grupo. La agrupación se hace minimizando la suma de cuadrados de distancias entre los datos y el centroide de agrupamiento correspondiente. Por lo tanto, el propósito de la agrupación de K-mean consiste en clasificar los datos. Finalizando con los métodos de agrupamiento se tiene los DBSCAN, los cuales son algoritmos de data clustering o agrupamiento basado en densidad porque encuentra un número de grupos comenzando por una estimación de la distribución de densidad de los nodos correspondientes.

2. MARCO TEÓRICO

2.1 Agrupamiento de objetos por similitud mediante k-means.

K-means es un método de agrupamiento que divide un dataset en *k* grupos. Para ello se basa en calcular la distancia entre un punto y los centroides de cada grupo. En la figura 1 se puede apreciar el proceso de agrupamiento del punto de color negro. El cluster al que pertenece el datapoint

corresponde al grupo cuyo centroide (círculo con X) se encuentre más cercano al datapoint, en el caso del punto negro, se asignará al grupo de color celeste.

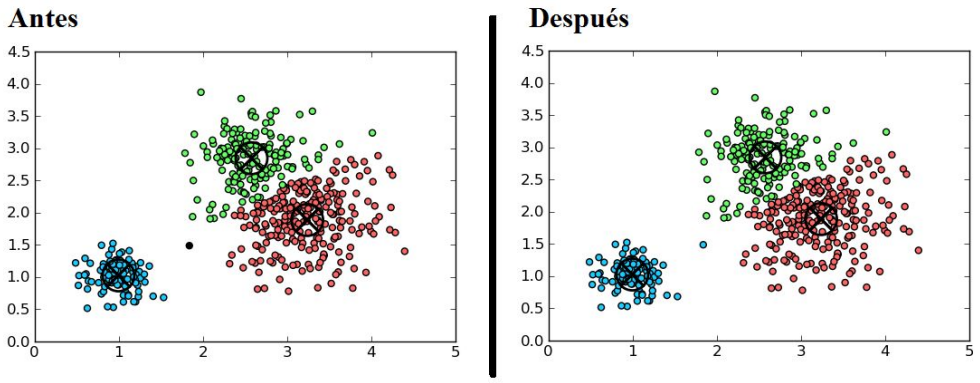

Figura 1: Ejemplo de aplicación de algoritmo de clasificación KNN.

Selección de número de grupos basado en el método elbow

El método elbow utiliza el porcentaje de varianza en función del número de grupos. Al iniciar con pocos números de grupos la varianza es grande, sin embargo; a medida que el número de grupos aumenta llega un punto donde la varianza tiende a ser constante, el número adecuado de grupos es aquel donde se produce dicho cambio.

En la figura 2 se muestra un ejemplo de la gráfica del método elbow, la distorsión comienza a minimizar de forma constante desde que k=3, por lo que se considera el el número de grupos adecuado es igual a 3.

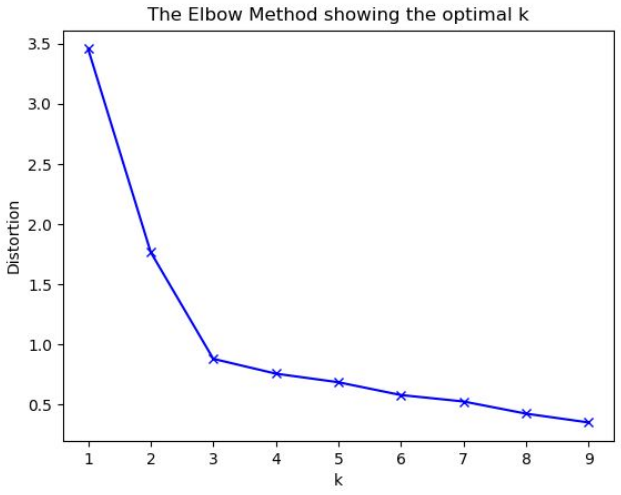

Figura 2: Ejemplo de gráfica de estimación de número de grupos aplicando método elbow.

Selección de número de grupos basado en el método siluetas

El método de siluetas facilita la selección de un número adecuado de grupos basándose en la homogeneidad de los datos de cada grupo se utiliza el método de siluetas para el cual se realiza los siguientes cálculos:

- Se calcula a(i) como la distancia promedio de un punto con respecto a los demás puntos del grupo.
- Se calcula b(i) como la menor distancia promedio de un punto respecto a los demás puntos del

grupo más cercano.

- Calcula el factor de silueta s(i) aplicando la siguiente fórmula: $s(i)=(b(i)-a(i))/\max\{b(i),a(i)\}$

El número grupos adecuado es aquel en el cual se parecen las siluetas de los grupos, en la figura 3 se puede apreciar siluetas no muy parecidas por lo cual los datos no son muy similares.

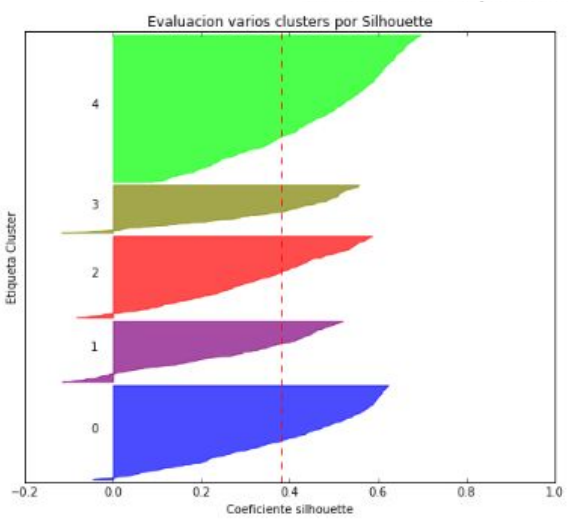

Figura 3: Ejemplo de gráfica de estimación de número de grupos aplicando método silueta.

2.2 Organización de grupos como un árbol jerárquico.

Enfoque alternativo al agrupamiento basado métodos jerárquicos tienen por objetivo agrupar clusters para formar uno nuevo o bien separar alguno ya existente para dar origen a otros dos, de tal forma que, si sucesivamente se va efectuando este proceso de aglomeración o divisi´on, se minimice alguna distancia o bien se maximice alguna medida de similitud.Los métodos jerárquicos se subdividen en:

- **Métodos aglomerativos o ascendentes:** Comienzan el análisis con tantos grupos como individuos haya. A partir de estas unidades iniciales se van formando grupos, de forma ascendente, hasta que al final del proceso todos los casos tratados están englobados en un mismo conglomerado.
- **Métodos disociativos o descendentes:** Comienzan con un conglomerado que engloba a todos los casos tratados y, a partir de este grupo inicial, a través de sucesivas divisiones, se van formando grupos cada vez más pequeños. Al final del proceso se tienen tantas agrupaciones como casos han sido tratados.

Los resultados del agrupamiento jerárquico son usualmente presentados en un dendograma (representación gráfica de datos en forma de árbol que organiza los datos en subcategorías que se van dividiendo en otros hasta llegar al detalle deseado). El agrupamiento jerárquico tiene la ventaja distintiva que todo lo que se usa es una matriz de distancia.

2.3 DBSCAN

En la Figura 4, se muestra un “clustering” mediante DBSCAN, el cual se basa en el agrupamiento espacial basado en densidad de aplicaciones con ruido, porque encuentra un número de clusters comenzando por una estimación de la distribución de densidad de los nodos correspondientes. DBSCAN puede encontrar clusters que no son linealmente separables, por ejemplo en los datos que usar k means o Gaussian Mixture EM clustering, etc. Encuentra muestras de núcleos de alta densidad y expande clusters de ellas, en donde por ejemplo los datos que contienen grupos de densidad similar.

En la Figura 4, por ejemplo: muestra que dado el dataset, el algoritmo comienza por un punto arbitrario que no haya sido visitado. La e-vecindad de este punto es visitada, y si contiene suficientes puntos, se inicia un clúster sobre el mismo. De lo contrario, el punto es etiquetado como ruido. Se debe notar que el punto en cuestión puede pertenecer a otra vecindad que lo incluya en el clúster correspondiente. Si un punto se incluye en la parte densa de un clúster, su e-vecindad también forma parte del clúster. Así, todos los puntos vecinos se añaden al clúster, al igual que la e-vecindad de estos puntos que sean lo suficientemente densas. Este proceso continúa hasta construir completamente un clúster densamente conectado. Entonces, un nuevo punto no visitado se visita y procesa con el objetivo de descubrir otro clúster o ruido.

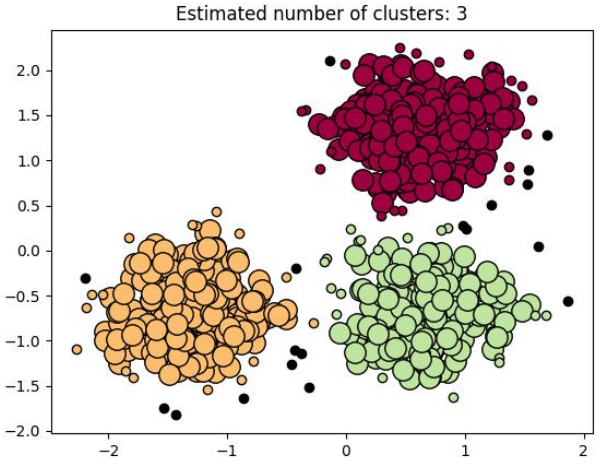

Figura 4: Implementación DBSCAN

3. MÉTODO Y MATERIALES

Previo a la aplicación del algoritmo se describen las herramientas y métodos necesarios para la ejecución del mismo y un mayor entendimiento de los procesos del algoritmo.

3.1 Materiales

Scikit-learn es una librería que provee algoritmos usados en Machine Learning para facilitar tareas de regresión, clasificación, predicción y clustering, se usan mayormente en minería de datos. El algoritmo KNN se encuentra en la función *KMeans* que cuenta con los siguientes parámetros:

- **n_clusters:** Se establece el número de clusters(grupos) a generar, por defecto se n_clusters=8.
- **init:** Establece el método de inicialización. Puede ser *k-means*(selecciona los centroides aumentando el valor de convergencia) o *random*(elige k filas aleatoriamente para generar los centroides iniciales)
- **n_init:** Establece el número de veces que el algoritmo se ejecutará con diferentes semillas de centroides, por defecto el valor de n_init es 10.
- **max_iter:** Número máximo de iteraciones del algoritmo por ejecución, por defecto max_iter=300.
- **tol:** Tolerancia relativa con respecto a la inercia para declarar convergencia, por defeto 1e-4
- **precompute_distances:** Precomputa las distancias

- **random_state:** Se establece cómo se seleccionan las semillas, en caso de establecer un número se utiliza un generador de números aleatorios basados en ese número, caso contrario; se utiliza np.random.
- **copy_x:** Al calcular distancias es más preciso numéricamente centrar primero los datos, si copy_x es true los datos originales no se modifican.
- **n_jobs:** Número de trabajos empleados para realizar el proceso de agrupamiento.
- **algorithm:** Algoritmo k means a utilizar. Utiliza el algoritmo clásico si n_jobs=full, si n_jobs=elkan se ejecuta una variación recomendada para datos densos.

La función *silhouette_score* permite calcular el coeficiente de silueta para las muestras se configura los siguientes parámetros:

- **X:** Almacena el dataset sobre el cual se aplicó el proceso de clustering.
- **labels:** Contiene una lista con las etiquetas de por cada datapoint.

4.1 Descripción del dataset.

El dataset utilizado es el archivo *dataset.csv (disponible en : https://drive.google.com/open?id=1er_xX_7xq2euO5jQeTrQKusUl7qcbZDj)* que es el corresponde a la muestra utilizada para el trabajo anterior realizado sobre Knn. El archivo cuenta con 16031 registros de pistas de audio con 518 atributos que conforman las características de audio entre las cuales tenemos:

- Centroide espectral
- Ancho de banda espectral
- Contraste espectral
- Coeficientes espectrales en las frecuencias de Mel

Además, cuenta con una columna que indica el autor de cada canción y el género musical. De este archivo se utilizó un 60% de registros para la fase de entrenamiento y un 30% para la fase de evaluación.

4.2 Aplicación e implementación del algoritmo K-means.

En la figura 5 se presenta el código implementado en el cual se realiza un bucle For varía el número de grupos hasta 16, por cada iteración se realizan las siguientes tareas:

- Creación del cluster
- Se calcula el promedio de la silueta
- Se determina el nivel de distorsión

```
for i in lista:
    clusterer = KMeans(n_clusters=i, random_state=10)  #Creación de Kmeans
    cluster_labels = clusterer.fit_predict(X_train)     #Muestra las marcas de los clusters

    silhouette_avg = silhouette_score(X_train, cluster_labels)  #Se obtiene el puntaje del cluster

    print("Para n_clusters =", i,
          "El promedio del score de la silueta es: ", silhouette_avg)
    distortions.append(clusterer.inertia_)
```

Fig.5: Código de implementación de algoritmo Kmeans.

En la Figura 5 existe una lista denominada *distortions*, esta almacena la distorsiones para graficar la función de elbow la cual se presenta en la Figura 6, por observación se puede determinar que el número de grupos adecuado es 8.

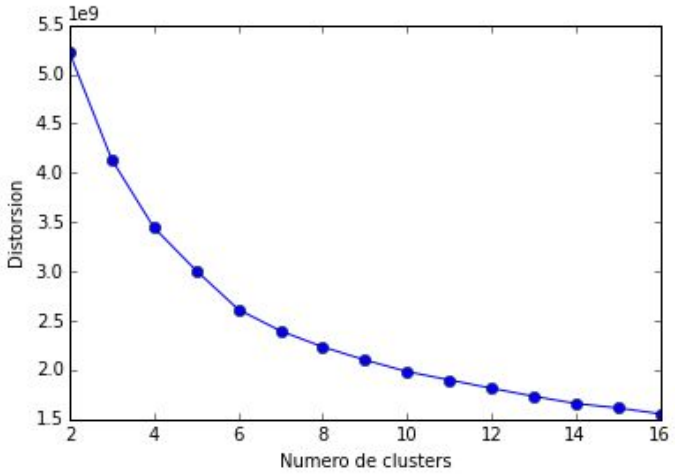

Fig.6: Código de implementación de algoritmo Kmeans.

Con el número de grupos elegido se realizó el diagrama elde siluetas para dicho modelo, en la Figura 7 se muestra el correspondiente diagrama donde se puede apreciar que los valores de las siluetas son bastante cercanos.

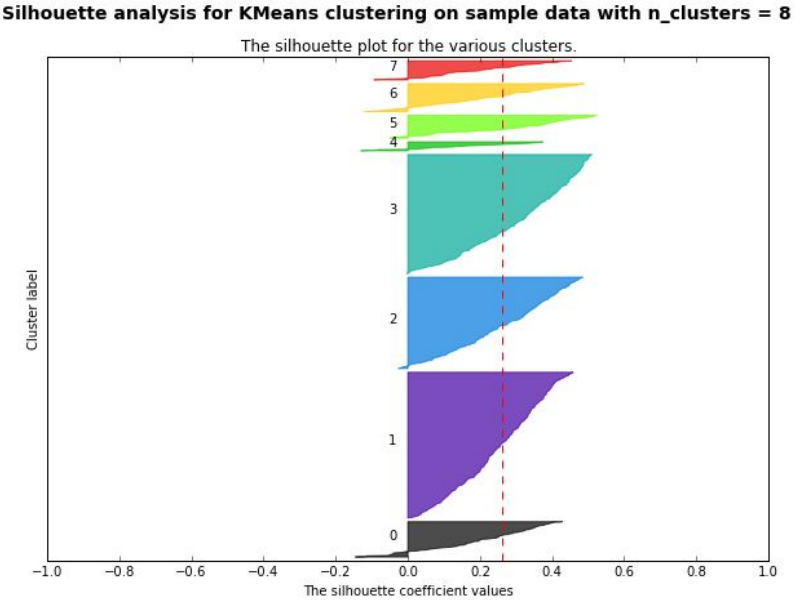

Fig.7:Diagrama de siluetas de un modelo de agrupamiento con 9 grupos..

4.3 Aplicación e implementación de Árboles Jerárquicos.

Para la implementación de árboles Jerárquicos en FMA dataset, se utiliza la librería de sklearn, scipy.cluster.hierarchy, con sus respectivas dependencias linkages y dendrogram (formas de representar árboles, con forma u invertida y enlaces de conexión). Estas dependencias permiten componer cada grupo dibujando un enlace en forma de U invertido entre un clúster no único y sus hijos. La parte superior del enlace U indica una fusión de clúster. Las dos patas del enlace U indican qué clusters se fusionaron. La longitud de las dos patas del enlace U representa la distancia entre los clusters secundarios. En la Figura 8 se presenta la implementación de los 4 pasos necesarios para implementar árboles jerárquicos los cuales son: 1) Obtención de matriz de distancia, 2) Establecer clusters aplicando distancia euclidiana, 3) representación tabular de cada cluster y 4) construcción de dendograma.

```
row_dist=pd.DataFrame(squareform(pdist(df,metric='euclidean')),columns=labels,index=labels) #Obtiene matriz de distancia

row_clusters = linkage(df.values, metric='euclidean', method='complete')    #Se establece los clusters aplicando distancia
                                                                            #euclidiana
clusters=pd.DataFrame(row_clusters,                                         #Representación tabular de los clusters
             columns=['row label 1', 'row label 2',
                      'distance', 'no. of items in clust.'],
             index=['cluster %d' % (i + 1)
                    for i in range(row_clusters.shape[0])])

row_dendr = dendrogram(row_clusters, labels=labels) #Grafica de dendograma
```

Figura 8: Implementación de Árboles Jerárquicos.

4.4 Aplicación e implementación de DBSCAN.

Para la correcta implementación de un dbscan en un dataset X, se debe implementar las sentencias presentadas en la Figura 9 pertenecientes de la librerias DBSCAN de sklearn.cluster.

```
# Compute DBSCAN
db = DBSCAN(eps=0.3, min_samples=10).fit(X)
print(X)
core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
core_samples_mask[db.core_sample_indices_] = True
labels = db.labels_

# Number of clusters in labels, ignoring noise if present.
n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)

print('Estimated number of clusters: %d' % n_clusters_)
print("Homogeneity: %0.3f" % metrics.homogeneity_score(labels_true, labels))
print("Completeness: %0.3f" % metrics.completeness_score(labels_true, labels))
print("V-measure: %0.3f" % metrics.v_measure_score(labels_true, labels))
print("Adjusted Rand Index: %0.3f"
      % metrics.adjusted_rand_score(labels_true, labels))
print("Adjusted Mutual Information: %0.3f"
      % metrics.adjusted_mutual_info_score(labels_true, labels))
print("Silhouette Coefficient: %0.3f"
      % metrics.silhouette_score(X, labels))
```

Figura 9: Código de implementación de DBSCAN

5. RESULTADOS

5.1 Resultados de proceso de agrupamiento usando algoritmo K-means.

Luego de realizar el procedimiento de agrupamiento K-means se realizó un proceso de entrenamiento con datos de prueba generó un archivo csv que contiene la identificación, el artista, género de una canción, y el grupo grupo al que pertenece (grupo generado por el proceso de agrupamiento), una porción del resultado se presenta en la Figura 10.

idCancion	Artista	Genero	NumCluster
3321	Big Digits	Hip-Hop	2
4104	Psychedelic	Rock	1
3746	Kelley Stoltz	Indetermina	1
7479	Mors Ontolo	Rock	1
14673	Raptorface	Electronic	0
19262	Receptors	Electronic	1
4745	Smokey Hor	Country	1
17370	Zack Kouns	Rock	3
713	The Functior	Rock	7
17588	Future Islanc	Electronic	1
671	Fanatic	Hip-Hop	2
21420	Dev/Null	Electronic	0
13217	The Unfinish	Rock	2
7492	BrokeMC	Hip-Hop	1
14168	Dielectric Dr	Indetermina	3

Figura 10: Archivo resultado de proceso de agrupamiento

Los géneros musicales de cada grupo se presenta en la Figura 11.

Cluster0	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6	Cluster7
Electronic	Blues	Classical	Blues	Electronic	Classical	Blues	Electronic
Experimental	Classical	Country	Classical	Experimental	Electronic	Electronic	Experimental
Hip-Hop	Country	Electronic	Country	Hip-Hop	Experimental	Experimental	Hip-Hop
Indeterminado	Electronic	Experimental	Electronic	Indeterminado	Folk	Hip-Hop	Indeterminado
International	Experimental	Folk	Experimental	Pop	Indeterminado	Indeterminado	Rock
Pop	Folk	Hip-Hop	Folk	Soul-RnB	Rock	International	
Rock	Hip-Hop	Indeterminado	Indeterminado			Jazz	
Soul-RnB	Indeterminado	Instrumental	Instrumental			Rock	
	Instrumental	International	International			Soul-RnB	
	International	Jazz	Jazz				
	Jazz	Old-Time / Historic	Old-Time / Historic				
	Old-Time / Historic	Pop	Pop				
	Pop	Rock	Rock				
	Rock	Soul-RnB	Spoken				
	Spoken	Spoken					

Figura 11: Géneros musicales de cada grupo

5.2 Resultados de proceso de agrupamiento usando Árboles Jerárquicos.

La ejecución del algoritmo de árboles jerárquicos produce una matriz que muestra lista los clusters existentes así como el número de elementos por cluster. En la Figura 12 se puede apreciar una porción de dicha matriz, así como el dendograma correspondiente a la agrupación de las canciones utilizadas.

	row label 1	row label 2	distance	no. of items in clust.
cluster 1	679	680	1.000000	2
cluster 2	684	685	1.000000	2
cluster 3	696	697	1.000000	2
cluster 4	55	56	1.046929	2
cluster 5	1786	1787	2.400637	2
cluster 6	1481	1482	2.413351	2
cluster 7	1804	1805	2.476131	2
cluster 8	851	852	2.614514	2
cluster 9	376	377	2.619228	2
cluster 10	396	397	2.708051	2

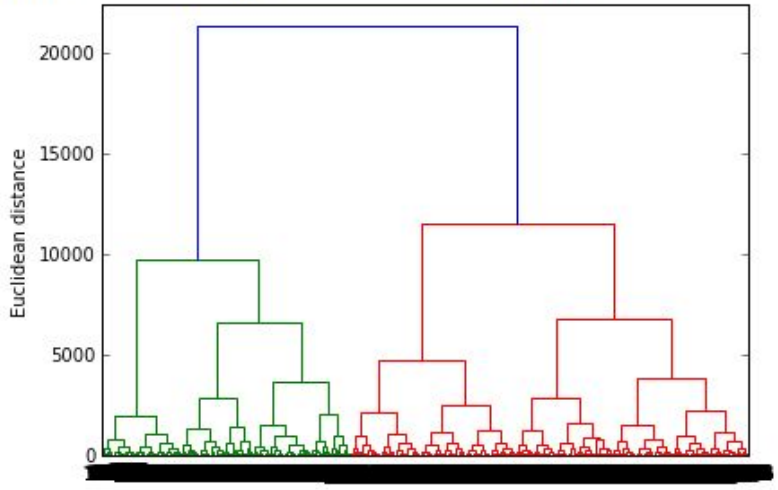

Figura 12: Matriz que describe los clusters aplicando "Árboles Jerárquicos" con su respectivo dendograma

5.3 Resultados de proceso de agrupamiento usando DBSCAN.

La Figura 13, muestra los resultados de DBSCAN, entre los cuales, homogeneidad (puntos de datos que son miembros de una sola clase) y v-measure (métrica de media armónica entre homogeneidad e integridad) tienen valores altos

```
Estimated number of clusters: 3
Homogeneity: 0.953
Completeness: 0.883
V-measure: 0.917
Adjusted Rand Index: 0.952
Adjusted Mutual Information: 0.883
Silhouette Coefficient: 0.626
```

Figura 13: Resultados del uso de DBSCAN.

6. RECOMENDACIONES

Para realizar un proceso de clustering se debe realizar primero la selección de variables que permitan identificar con claridad el tipo de cluster al que pertenece, si bien los atributos del dataset podrían facilitar la tarea debido a la caracterización de un género musical basado en sus diferentes espectros de sonido; la interpretación de cada cluster se dificulta debido a la gran cantidad de variables y a desconocimiento de temas relacionados con señales de audio.

Ante un dataset no separable linealmente es mejor utilizar un DBSCAN, ya que este puede encontrar clusters en datos no linealmente separables.

Dependiendo del caso de estudio, que se busque la clasificación de datos, se puede emplear las distintas herramientas de machine learning (Figura 14).

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium `n samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters	Large `n samples` and `n_clusters`	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters, linkage type, distance	Large `n samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large `n samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes	Distances between nearest points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large `n clusters` and `n_samples`	Large dataset, outlier removal, data reduction.	Euclidean distance between points

Figura 14: Comparativa entre las técnicas de clusterización.

7. CONCLUSIONES

El proceso de agrupamiento es una técnica aplicable cuando se requiere generar grupos de datos, sin embargo; el resultado aporta ninguna información referente a las características de cada grupo, es por ello; que la interpretación de cada cluster se dificulta. Otro caso puede donde la interpretación de cada cluster se dificulta es cuando existen demasiadas variables dependientes, es por ello; que se recomienda el uso de herramientas estadísticas como R.

La implementación del agrupamiento por DBSCAN, se debe reducir a casos en donde: se tenga un dataset no divisible linealmente, y en casos donde se tiene ruidos, ya que esta es una herramienta no determinista.

En cuanto a la técnica de árboles jerárquicos se puede concluir que es una técnica muy aplicable y con

buenos resultados, fácil de implementar y comprender, según el campo que se quiera explorar, si se buscan relaciones jerárquicas para saber que datos se derivan de otros entonces la técnica de árboles jerárquicos es recomendada, aunque por otro lado se puede argumentar que tiene un coste de recursos computacionales exigente.

8. REFERENCIAS

[1] Colaboradores UCI, *UCI Machine Learning Repository* Available online: https://goo.gl/PhtzjZ.

[2] Raschka Sebastián- 2016 - *Python Machine Learning*

[3] Gravano, Agustin - 2015 - *Aprendizaje Automatico* Available online: http://www.dc.uba.ar/materias/aa/2015/cuat2/ibl

[4] Colaboradores Scikit, *Scikit-Learn Documentation.* Available online: http://scikit-learn.org/stable/

[5] Microsoft Excel https://products.office.com/es/excel

[6] IBM SPSS Statistics 23 Available https://www.ibm.com/mx-es/marketplace/spss-statistics

[7] Editores Scipy. "*Hierarchy Dendogram"*. Available online: https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.dendrogram.html

[8] Editores Scikit. *"Hierarchy Clustering"* Availble online: http://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering

[9] Colaboradores Wikipedia. *"Dendograma"*, 2017. Available online: https://es.wikipedia.org/wiki/Dendrograma