Clasificación de géneros en Spotify, desmitificada con algoritmos de clustering

1st Paola Andrea Grajeda
email paola.grajeda@gmail.com

2nd Adrian Marcelo Paredes
email adrianparedes82@gmail.com

***Abstract*—El objetivo del estudio es determinar si la categorización por género de Spotify se corresponde a una clusterización natural o si resulta arbitraria. Se trabajará con los algoritmos de KMeans, Clustering Jerárquico Aglomerativo y Clustering Espectral. Para los dos últimos, se utilizará una optimización Grid Search y Bayesiana para encontrar los hiperparámetros que maximizarán las medidas de validación interna o externa. El resultado obtenido confirmará las sospechas de que cinco géneros no es la clasificación más natural, sino que la agrupación orgánica basada en los features de las canciones tiende a formar tres grupos. Por último, se analizará una pista del maestro Piazzola con clustering espectral.**

I. Introducción

Entre una de las clasificaciones que utiliza Spotify para recomendar canciones, se encuentra la de género. Se consultó el servicio https://developer.spotify.com/console/get-available-genre-seeds/ de Spotify y se obtuvo un listado de 126 géneros de los cuales sólo se utilizarán 5 (`ambient`, `classical`, `drum-and-bass`, `jazz` y `world-music`) en el presente estudio.

Se utilizarán los datasets provistos por la cátedra (`metadata`, `audio_features` y datasets de timbres y pitches de 2,205 pistas) para aplicar, tras una mínima preparación, con los siguientes algoritmos de clustering:

- KMeans
- Jerárquico Aglomerativo
- Clustering Espectral: UMAP + KMeans

En cada caso se intentará optimizar los hiper-parámetros correspondientes a cada algoritmo así como también el dataset a utilizar y la técnica de normalización. Dependiendo del experimento, se utilizarán distintos métodos de optimización (inspección visual, Grid Search, Optimización Bayesiana). Cada modelo será evaluado con indicadores internos y externos, siempre buscando el agrupamiento más estable y el parecido con los géneros de Spotify. Las Matrices de Confusión y TSNE, para reducir la dimensionalidad y poder graficar los resultados, serán herramientas extremadamente útiles para tomar decisiones y comunicar.

Por ende, nos proponemos a validar si la clasificación de géneros de Spotify corresponde a una agrupación natural de las canciones, o se trata de un etiquetado arbitrario como cualquier otro que podría realizar un ser humano.

II. Preparación de los Datos

Los datos fueron preparados a partir de los siguientes datasets:

- **metadata.csv:** Dataset que contiene la metadata de cada una de las pistas de audio.
- **audio_features.csv:** Dataset que contiene la información cuantitativa de las pistas de audio [1].
- **Datasets de timbres:** Un conjunto de archivos CSV, donde cada archivo es un dataset que contiene una serie temporal con los valores de los 12 timbres clasificados por Spotify en distintas fracciones de tiempo llamadas **segmentos**.
- **Datasets de pitches:** Un conjunto de archivos CSV, donde cada archivo es un dataset que contiene una serie temporal con los valores de los 12 pitches clasificados por Spotify en distintas fracciones de tiempo llamadas **segmentos**.

A. Dataset audio_features

El dataset no tiene valores faltantes. Se cuenta con 2,206 canciones y 17 features: **acousticness**, **analysis_url**, **danceability**, **duration_ms**, **energy**, **instrumentalness**, **key**, **liveness**, **loudness**, **mode**, **speechiness**, **tempo**, **time_signature**, **track_href**, **type**, **uri** y **valence**. Se puede leer una descripción de cada uno de estos atributos en la documentación para desarrolladores de Spotify [1].

Se omitieron los siguientes features no numéricos:

- **analysis_url**
- **track_href**
- **type**
- **uri**

Durante el análisis exploratorio se detectaron correlaciones entre **energy** y **acousticness** (-0.833) y entre **loudness** y **energy** (0.866154). Se planteó remover el feature **energy**, pero al final se decidió no hacerlo. Se encontraron outliers, pero no hizo falta tratarlos para los algoritmos de clustering con los se trabajó.

Dado que hay un track del cual no tenemos sus datos de timbres y pitches en audio_analysis, se elimina del dataset. Es el tema *Honeysuckle Rose*, de *Oscar Peterson*.

En lo que resta del informe, este dataset será llamado *Features*.

B. Dataset audio_analysis

Se leyó cada uno de los 2,205 archivos de timbre y los 2,205 archivos de pitches con la información de las series de tiempo de cada una de las pistas de audio. Para cada canción, se resumió las 12 variables de timbre y las 12 de pitches tomando

la media y el desvío estándar, juntando todas las canciones y obteniendo así un único dataset llamado audio_analysis con 2,205 canciones y 48 columnas (12 medias de timbre, 12 desvíos estándares de timbres, 12 medias de pitches y 12 desvíos estándares de papers).

Entre las variables resumen también se detectaron correlaciones y outliers, pero no hizo falta tratarlos para los algoritmos de clustering con los que se trabajó.

En lo que resta del informe, este dataset será llamado *Analysis*.

C. Dataset audio_tracks

Se armó un dataset con la fusión de los dos datasets anteriores. Los datasets fueron mergeados y ordenados por id para facilitar las comparaciones.

En lo que resta del informe, este dataset será llamado *Tracks*.

D. Tendencia al Clustering

Antes de comenzar con las corridas de los algoritmos de clustering, se evaluó la tendencia al clustering de cada uno de los datasets calculando el Coeficiente de Hopkins.

TABLE I
TABLA ESTADÍSTICO (COEFICIENTE) DE HOPKINS

	Hopkins
Features	0.0806
Analysis	0.0844
Tracks	0.0978

En la Tabla I se pueden ver los valores obtenidos del Coeficiente de Hopkins para cada uno de los datasets. Dado que los estadísticos están mucho más cerca de cero que de 0.5, se puede deducir que las canciones tienen una alta tendencia al clustering.

III. EXPERIENCIA 1: KMEANS

A. Métodos

Se analizaron dos escenarios de estandarización. El primero con normalización de mínimos y máximos (entre 0 y 1) y el segundo con estandarización Z-score. Se seleccionó el número de clusters óptimo en base al promedio del coeficiente de Silhoutte y del SSE para pruebas con K entre 2 y 15 para las tres variantes del dataset: *Features*, *Analysis* y *Tracks*.

En una primera corrida el dataset de *audio_analisys* no estaba estandarizado, y eso provocaba que, por ejemplo, al ejecutar KMeans sobre *Tracks*, los resultados obtenidos fueran iguales que sobre *Features*.

También se ejecutaron corridas omitiendo las variables categóricas **key** (discreta de 12 valores), **time_signature** (discreta de 4 valores) y **mode** (dicotómica). Dado que KMeans se basa en distancias euclídeas, se pensó que agregar estas variables discretas podría agregar ruido al clustering. Sin embargo, los resultados fueron similares con y sin ellas.

Optimizando las medidas de validación interna que se pueden aplicar con KMeans, o sea encontrando un balance

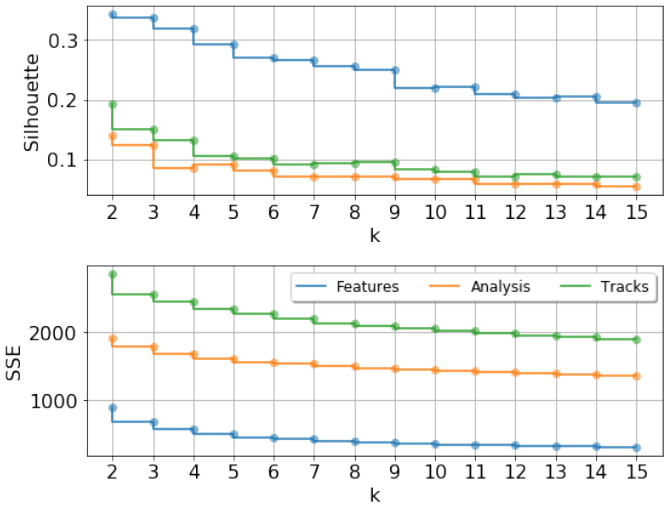

Fig. 1. Coeficiente de Silhoutte y SSE para escenario Max-Min Scale

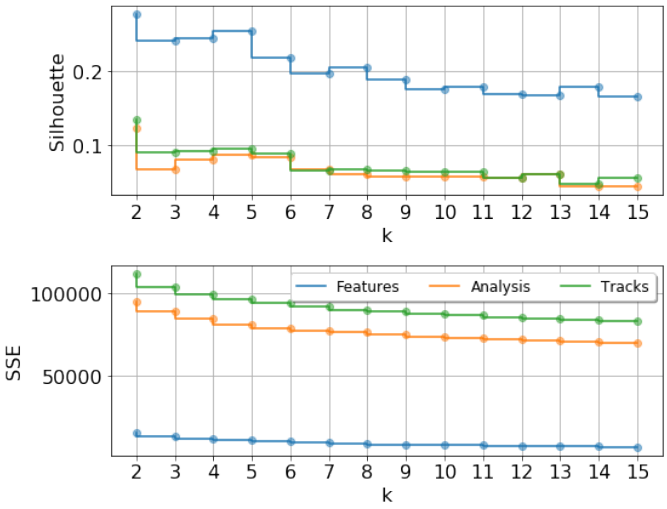

Fig. 2. Coeficiente de Silhoutte y SSE para escenario Z-score

entre maximizar Silhouette y minimizar SSE, se eligió un modelo ganador con un K, un método de estandarización y un dataset determinado. También se hicieron comparaciones entre los distintos modelos encontrados usando validaciones externas entre sí.

Con el modelo seleccionado, se realizaron validaciones externas comparando contra la agrupación de géneros provista por Spotify en el dataset *Metadata*, obteniendo la Matriz de Confusión y las medidas de van Dongen y el Rand Index Ajustado.

Por último, se aplicó TSNE como técnica de reducción de dimensionalidad para visualizar las agrupaciones propuestas por Spotify y las encontradas por nuestro modelo.

B. Resultados y discusión

Analizando la Fig. 1 del primer escenario de estandarización, se ve que los mejores valores (alto Silhoutte y

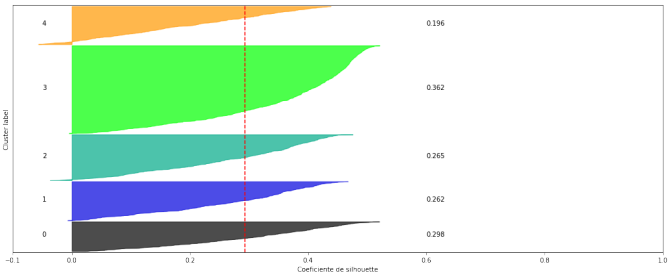

Fig. 3. Gráfico de Silhouttes para cada observación y cada cluster con K=5, usando el dataset de *Features* normalizado por mínimos y máximos

bajo SSE) se obtuvieron con el dataset de *Features*. El quiebre de SSE se obtuvo con un K entre 5 o 6. Con valores más altos de K el Silhouette decrecía demasiado y ya no se obtenía una ganancia significativa en el SSE. El Silhouette promedio obtenido en 5 fue mayor que en 6. Para K=5 el promedio de Silhoutte fue 0.2929 y el SSE 517.84

En el segundo escenario de estandarización de la Fig. 2 se mantuvo que el dataset más adecuado para aplicar KMeans fue el de *Features*. El K óptimo de vuelta fue encontrado en 5. El promedio de Silhoutte fue 0.2529 y el SSE 11,181.

El resto de los análisis se hicieron tomando el primer escenario, el de normalización de mínimos y máximos, porque el promedio de Silhoutte obtenido con este método de estandarización fue el más alto.

Se crearon las matrices de confusión y el posterior cálculo de la medida normalizada de van Dongen y Rand Index Ajustado, para comparar los resultados obtenidos con los distintos datasets con un K=5.

TABLE II
TABLA MEDIDAS DE VALIDACIÓN EXTERNA

	van Dongen	**Rand Index**
Features vs Analysis	0.6907	0.179234
Features vs Tracks	0.9944	-0.001036
Analysis vs Tracks	0.9977	0.000411

Las medidas de validación externa que se muestran en Tabla II, indican que los pares de clusterings más parecidos fueron *Features* vs *Analysis*.

Se realizó una validación interna del modelo de KMeans más óptimo conseguido (K=5 con el dataset de *Features*). En la Fig. 3 se puede apreciar un gráfico con los Silhouttes de cada una de las canciones. Las canciones se muestran agrupadas por clusters y cada cluster etiquetado con su Silhoutte promedio.

Se calculó la matriz de confusión para este mismo modelo y los índices de validación externa para probar si los clusters sobre *Features* coincidieron con el género de los datos originales del dataset *Metadata*.

Como se puede apreciar en la Tabla III, cuesta hacer coincidir los clusters encontrados por el modelo con la categorización real del dataset *Metadata*. En naranja se sugiere algún etiquetado posible. El cluster 0 podría ser etiquetado como `ambient` o como `classical`. El cluster 1 podría ser etiquetado como `drum-and-bass` con seguridad (aunque con un error considerable). Sin embargo, el cluster 3 también podría ser etiquetado como `drum-and-bass`. El cluster 2 se podría etiquetar como `jazz` y el cluster 4 como `world-music`.

TABLE III
MATRIZ DE CONFUSIÓN

	0	**1**	**2**	**3**	**4**
ambient	297	118	33	9	3
classical	320	0	78	0	7
drum-and-bass	0	229	0	221	1
jazz	116	47	115	18	130
world-music	76	26	60	95	206

Los índices de Rand Index Ajustado y van Dongen dieron 0.2271 y 0.6810 respectivamente. Estos números refuerzan el leve grado de similitud recién mencionado entre los géneros y las clusters. El Rand Index Ajustado debería haber estado más cerca de uno y van Dongen más cerca de cero para considerar que se obtuvo un buen clustering [2].

Se utilizó la técnica de reducción de dimensionalidad TSNE para graficar los géneros del dataset *Features*.

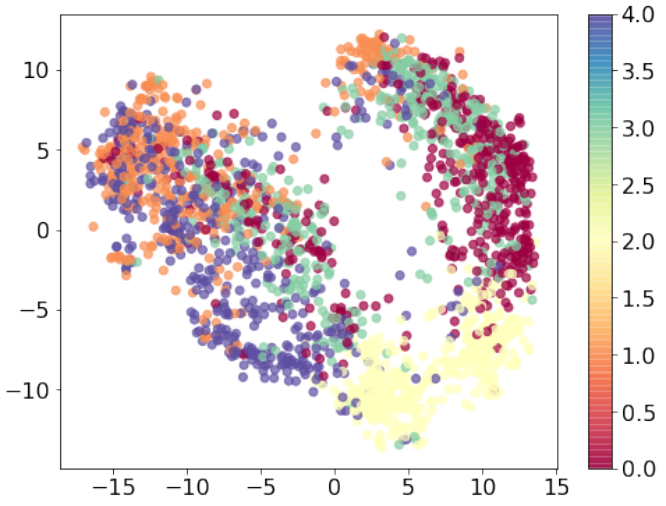

Fig. 4. Resultado TSNE para los géneros aplicado sobre el dataset *Features*

El la Fig. 4 se puede apreciar que hay un género bien definido en amarillo. Ese género es `drum-and-bass`. Luego se puede apreciar que los géneros `ambient` (azul) y `classical` (naranja) se muestran cercanos en la proyección, al igual que `world-music` (rojo) y `jazz` (verde). Claramente `drum-and-bass` es el género más separado y el resto tiende a mezclarse, indicando que en la realidad las canciones presentan más bien un continuo y que los puntos de corte que da Spotify en las clasificaciones en algún punto pueden ser arbitrarias.

A continuación se ejecutó TSNE sobre el clustering devuelto por el modelo que venimos analizando. Los resultados se pueden observar en la Fig. 5.

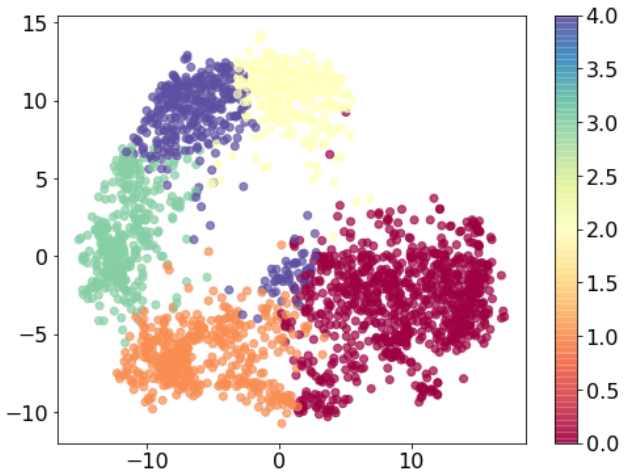

Fig. 5. Resultado TSNE para los clusters de KMeans

Ya se vio en la Tabla III que es muy difícil etiquetar los clusters de este modelo con la clasificación de géneros de Spotify. Sin embargo, en la proyección se puede apreciar que las canciones quedaron bastante bien separadas en sus cinco clusters. El cluster violeta bien podría ser `world-music`, que como vemos se mezcla un poco con el rojo (que podría ser `ambient` o `classical`) y el naranja (que podría ser `drum-and-bass`). El cluster verde también podría ser `drum-and-bass`. Y el amarillo en la punta superior podría ser `jazz`.

Como conclusión, se puede decir que el modelo KMeans que se encontró optimizando Silhuette y SSE, no funciona tan mal separando las canciones, aunque las medidas de Silhouette son bastante bajas. Sin embargo, la agrupación encontrada difícilmente coincide con la agrupación que es brindada por Spotify. Aunque sí coincide el número de grupos.

IV. Experiencia 2: Jerárquico Aglomerativo

A. Métodos

El segundo algoritmo de clustering elegido fue el de Jerárquico Aglomerativo.

Se ejecutó una optimización Grid Search con el siguiente espacio de búsqueda:

- **Datasets:** *Features*, *Analysis*, *Tracks*
- **Scalers:** `MinMaxScaler`, `StandardScaler`
- **Methods:** `single`, `complete`, `average`, `ward`
- **Matriz de Distancia:** `braycurtis`, `canberra`, `chebyshev`, `cityblock`, `correlation`, `cosine`, `dice`, `euclidean`, `jaccard`, `kulsinski`, `mahalanobis`, `matching`, `minkowski`, `rogerstanimoto`, `russellrao`, `seuclidean`, `sokalmichener`, `sokalsneath`, `sqeuclidean`

Y se buscó maximizar el Coeficiente de Correlación Cofenético, el indicador que mide la correlación entre la matriz de distancia utilizada (el cuarto de los hiper-parámetros mencionados) y las distancias extraídas del árbol generado [2].

Al igual que en el paso anterior, se evaluó el mejor modelo con los hiper-parámetros optimizados (en este caso el de Coeficiente de Correlación Cofenético más elevado) utilizando como validaciones externas van Dongen y el Rand Index Ajustado para comparar contra los géneros. Por último, se ejecutó TSNE para reducir la dimensionalidad del clustering y mostrar los resultados en un gráfico de dos dimensiones.

B. Resultados

Se ejecutaron 366 corridas de clustering jerárquicos aglomerativos. Si bien la combinación de todos los hiper-parámetros con Grid Search tendría que haber dado más cantidad de ejecuciones, algunas combinaciones que daban error o no tenían sentido desde el punto de vista teórico se saltearon, como por ejemplo ejecutar el método de Ward en matrices de distancias no euclídeas.

En la Tabla IV se muestra el top 10 de los modelos con más alto Coeficiente de Correlación Cofenético. En todos ellos se usó el dataset de *Analysis*.

TABLE IV
Top 10 de los modelos con más alto Coeficiente de Correlación Cofenético

Method	Metric	Scaler	Cophenet	vanDongen	adjRand
average	canberra	minMax	0.767256	0.998858	-0.000184
average	braycurtis	minMax	0.761583	0.998299	-0.000224
single	canberra	minMax	0.760386	0.997709	-0.000069
single	braycurtis	minMax	0.755941	0.997710	-0.000037
average	euclidean	minMax	0.747303	0.879210	0.046009
average	minkowski	minMax	0.747303	0.879210	0.046009
average	cosine	minMax	0.744705	0.731664	0.115078
single	chebyshev	standard	0.741269	0.998855	-0.000079
average	seuclidean	standard	0.740272	0.997710	0.000019

El modelo ganador fue aquel que se construyó usando el dataset *Analysis*, el método `average`, el `MinMaxScaler` y la matriz de distancia construida con la distancia de `canberra`. El Coeficiente de Correlación Cofenético obtenido fue 0.7672.

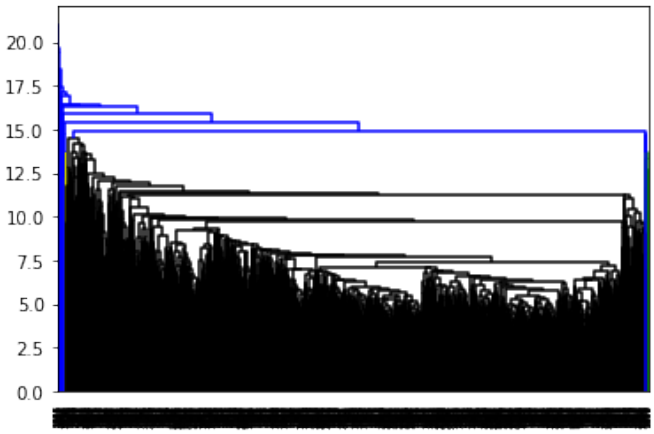

Fig. 6. Árbol de clustering jerárquico average con distancia Canberra

Si bien el Coeficiente de Correlación Cofenético fue alto, al graficar el árbol se verificó que el clustering era en verdad muy malo. En la Fig. 6 se puede ver cómo casi todas las canciones cayeron en un solo cluster, y algunas muy pocas en un segundo cluster muy pequeño.

En la matriz de confusión de la Tabla V se puede apreciar que todos los géneros de Spotify cayeron en el único cluster gigante que se ve en negro en la Fig. 6.

TABLE V
MATRIZ DE CONFUSIÓN

	0	**1**
ambient	459	1
classical	405	0
drum-and-bass	451	0
jazz	426	0
world-music	463	0

El índice de van Dongen de este modelo dio 0.9988 y el Rand Index Ajustado -0.0001, lo que indica, como bien se puede apreciar a simple vista en la Matriz de Confusión, que los resultados fueron muy malos.

Se probó graficar los árboles de los demás modelos del Top 10 de la Tabla IV y los resultados fueron similares: un cluster enorme donde caían casi todas las canciones y unos pocos clusters muy pequeños.

En la Fig. 7 se muestra el árbol de un modelo que nos llamó la atención. Es un clustering generado con el método `average` y la distancia `euclidean` sobre el dataset *Features* normalizado con mínimos y máximos. El Coeficiente de Correlación Cofenético de este modelo fue de 0.7042, no tan distinto del coeficiente óptimo.

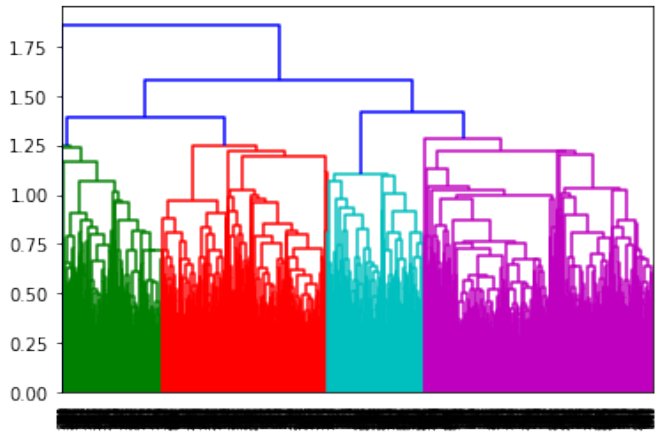

Fig. 7. Árbol de clustering jerárquico average con distancia Euclídea

Se ejecutó TSNE sobre este modelo cortando el árbol a la altura en la que genera cuatro clústeres, como sugiere la Fig. 7. El resultado obtenido puede verse en la Fig. 8, donde aparecen tres clusters bien marcados y separados (el rojo, el naranja y el verde) y el cuarto (el violeta) prácticamente no existe.

Esta agrupación está muy bien. Incluso mejor que la encontrada con KMeans, ya que la separación entre grupos es mucho

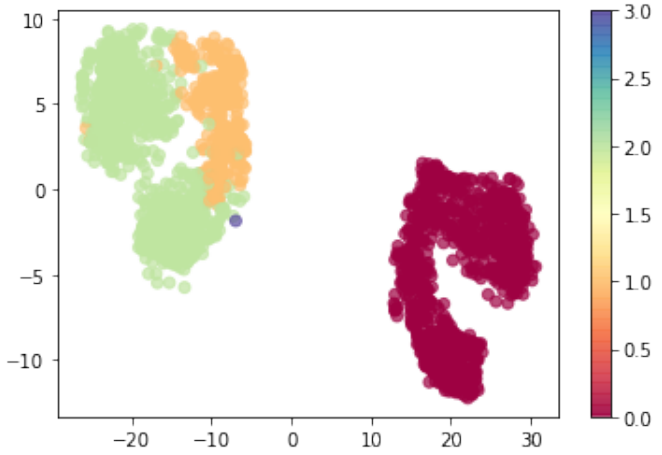

Fig. 8. Resultado TSNE para los clusters clustering jerárqico average con distancia Euclídea

más clara. Sin embargo, al aplicar las validaciones externas, se puede ver una vez más que no coincide en nada con los géneros de Spotify. El índice van Dongen fue de 0.8782 y Rand Index Ajustado de 0.0653, ambos extremadamente malos.

Como conclusión, se puede decir que, al igual que con KMeans, es posible encontrar modelos que generen buenos clusters con altos Coeficientes de Correlación Cofenético (aunque no necesariamente los más alto dan los mejores modelos). Sin embargo, la agrupación encontrada difícilmente coincide con la agrupación que es brindada por Spotify. Y esta vez ni siquiera coincide el número de grupos.

V. EXPERIENCIA 3: CLUSTERING ESPECTRAL

A. *Métodos*

Como tercer algoritmo se eligió aplicar un Cluster Espectral. Para ello se escogió una técnica de reducción de la dimensionalidad, en este caso UMAP, para luego aplicar KMeans sobre las nuevas dimensiones generadas.

Se ejecutó una Optimización Bayesiana con el siguiente espacio de búsqueda:

- **Datasets:** *Features*, *Analysis*, *Tracks*
- **Scalers:** `MinMaxScaler`, `StandardScaler`
- **Neighbors:** Hiper-parámetro de UMAP. Se muestrearon valores uniformemente variados desde 2 hasta 200 inclusive.
- **Mínima Distancia:** Hiper-parámetro de UMAP. Se muestrearon los siguientes valores: 0, 0.1, 0.25, 0.5, 0.8, 0.99.
- **Components:** Hiper-parámetro de UMAP. Es la cantidad de dimensiones. Se muestrearon valores uniformemente variados desde 2 hasta 16 inclusive.
- **Matriz de Distancia:** Hiper-Parámetro de UMAP. Se muestrearon los siguientes valores: `euclidean`, `manhattan`, `minkowski`, `canberra`, `mahalanobis`, `cosine`, `correlation`

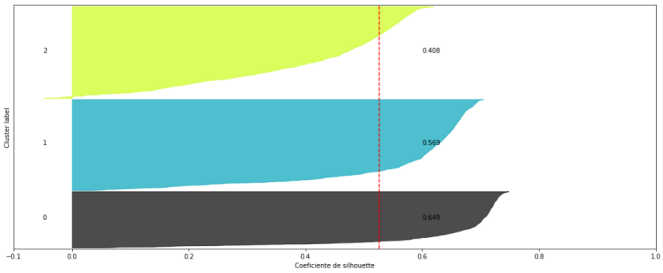

Fig. 9. Gráfico de Silhouttes para el Clustering Espectral con menor van Dongen

- **K:** Hiper-parámetro de KMeans. Es la cantidad de clusters. Se muestrearon valores uniformemente variados desde 2 hasta 15 inclusive.

Se usó el paquete `hyperopt` de `python` para la Optimización Bayesiana. Se hicieron tres corridas. La primera minimizando van Dongen con 500 iteraciones. La segunda maximizando Rand Index Ajustado con 500 iteraciones. Y una prueba más minimizando van Dongen pero con 1,000 iteraciones.

Ya sabiendo que es posible encontrar buenas agrupaciones con KMeans y Jerárquicos Aglomerativos, lo que se buscó en este experimento fue determinar si es posible encontrar un clustering que se parezca un poco más a la agrupación por géneros de Spotify.

B. Resultados

Se ejecutó la primera Optimización Bayesiana minimizando van Dongen. Los hiper-parámetros encontrados fueron:

- **Dataset:** *Tracks*
- **Scaler:** `MinMaxScaler`
- **Neighbors:** 50
- **Mínima Distancia:** 0
- **Components:** 5
- **Matriz de Distancia:** `canberra`
- **K:** 3

Con este modelo se obtuvo un Silhouette promedio de 0.5258. Bastante más alto que 0.2929, el Silhouette promedio obtenido en el primer experimento con KMeans. Lo que indica que a nivel validación interna este clustering fue bastante mejor. En la Fig. 9 se puede apreciar el alto Silhouette de los tres clusters generados (0.69, 0.549 y 0.47).

En la Matriz de Confusión de la Tabla VI se puede ver que efectivamente los clusters encontrados no difieren tanto de los géneros de Spotify. El cluster 0 se podría etiquetar como una suma de `ambient` + `classical`. El cluster 2 podría etiquetar como `drum-and-bass`. Se debe recordar que este era el género que se mostraba más separado de los demás en la proyección TSNE del dataset *Features* en la Fig. 4. Por último, el cluster 2 se podría etiquetar como una suma de `jazz` y `world-music`, quizá con un poco de `ambient` también.

TABLE VI
MATRIZ DE CONFUSIÓN

	0	**1**	**2**
ambient	318	28	114
classical	382	2	21
drum-and-bass	1	440	10
jazz	53	11	362
world-music	85	37	341

El índice van Dongen de esta matriz fue de 0.4462. Efectivamente el valor más bajo hasta ahora conseguido. El Rand Index Ajustado fue de 0.3762, el valor más alto encontrado.

A esta clusterización, se le aplicó un reducción de dimensionalidad con TSNE y se obtuvo el gráfico que se ve en la Fig. 10, donde los tres clusters se muestran bien marcados y separados. El cluster rojo representa al 0, etiquetado como `ambient + classical`; el cluster amarillo representa al 1, etiquetado como `drum-and-bass`; y el cluster violeta representa al 2, etiquetado como `jazz + world-music`.

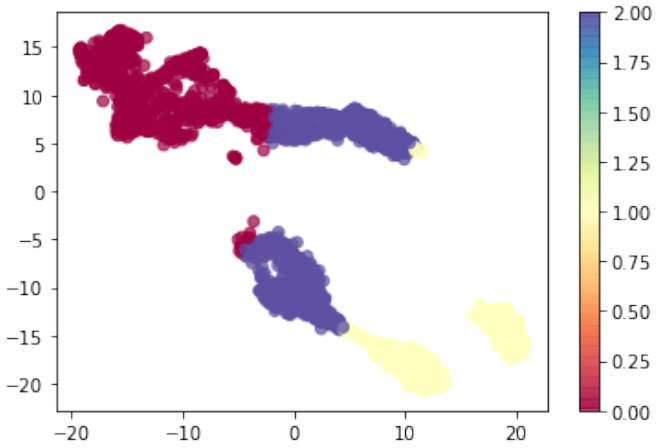

Fig. 10. Resultado TSNE para los clusters del Clustering Espectral con menos van Dongen

Maximizando el Rand Index Ajustado se obtuvo resultados muy parecidos. El mejor modelo encontrado por la Optimización Bayesiana fue uno con los siguientes hiper-parámetros:

- **Dataset:** *Tracks*
- **Scaler:** `StandardScaler`
- **Neighbors:** 9
- **Mínima Distancia:** 0
- **Components:** 4
- **Matriz de Distancia:** `manhattan`
- **K:** 3

El Silhouette promedio fue de 0.5557, van Dongen de 0.4332 y Rand Index Ajustado de 0.4018. La clusterización fue similar a la anterior: Un grupo para `ambient + classical`, otro para `jazz + world-music` y uno dedicado a `drum-and-bass`.

Con 1,000 iteraciones en la Optimización Bayesiana se logró conseguir un modelo con mejores valores aún. Un van Dongen de 0.4759, Rand Index Ajustado de 0.3883 y un Silhouette promedio de 0.4286. El Silhoutte fue un poco más bajo que los otros dos anteriores, pero este modelo tuvo algo curioso y fue que como valor de K se obtuvo 4 clusters, en lugar de 5.

Los valores de los hiper-parámetros fueron:

- **Dataset:** *Tracks*
- **Scaler:** `MixMaxScaler`
- **Neighbors:** 107
- **Mínima Distancia:** 0
- **Components:** 7
- **Matriz de Distancia:** `canberra`
- **K:** 4

En la Tabla VII se puede ver cómo en este modelo en que tenemos un cluster más se empezó a partir un poco el género `jazz`, minimizando el van Dongen, que fue lo que se le pidió a la Optimización Bayesiana, pero sacrificando un poco de Silhouette.

TABLE VII
MATRIZ DE CONFUSIÓN

	0	**1**	**2**	**3**
ambient	70	27	292	71
classical	56	1	343	5
drum-and-bass	0	439	1	11
jazz	188	5	18	215
world-music	69	29	59	306

Por último, se hizo una nueva prueba. Dispuestos a sacrificar un poco de Silhouette, se ejecutó otra Optimización Bayesiana de 500 iteraciones para minimizar van Dongen, en la que se forzó el parámetro K=5. El objetivo de esta búsqueda fue detectar si existía un modelo que agrupara en cinco clusters a las canciones de una manera más parecida a como lo hacen los géneros de Spotify.

Los valores hallados para los hiper-parámetros fueron:

- **Dataset:** *Tracks*
- **Scaler:** `StandardScaler`
- **Neighbors:** 125
- **Mínima Distancia:** 0
- **Components:** 2
- **Matriz de Distancia:** `mahalanobis`
- **K:** 5

Lo que se encontró fue un modelo con van Dongen de 0.4469, Rand Index Ajustado de 0.3611 y un Silhouette promedio de 0.3806 (se vuelve a recordar que el Silhouette del modelo de KMeans era de 0.2929). En la Matriz de Confusión de la Tabla VIII se puede ver que efectivamente este clustering puede etiquetarse muy bien con los géneros de Spotify y en la Fig. 3 se ve que cualquiera de los cinco clusters tiene mejor Silhouette que los clusters de la Fig. 3 de KMeans.

Como conclusión, se puede reafirmar lo que ya se había visto con el experimento de KMeans pero con menos fuerza:

TABLE VIII
MATRIZ DE CONFUSIÓN

	0	**1**	**2**	**3**	**4**
ambient	215	83	29	70	63
classical	32	330	6	37	0
drum-and-bass	30	1	386	3	31
jazz	36	23	17	264	86
world-music	22	32	36	132	241

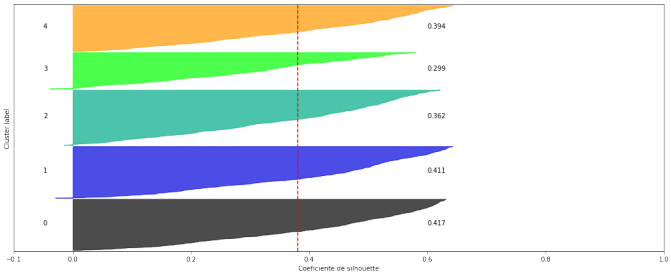

Fig. 11. Gráfico de Silhouttes para el Clustering Espectral con menor van Dongen y K=5

Los agrupamientos naturales con los que se obtuvo mayor fortaleza en los índices de validación interna fueron aquellos con los que se obtuvieron tres clusters, en los que `ambient` y `classical` se juntaron para formar un solo género, `jazz` y `world-music` para formar otro, y `drum-and-bass` fue un género aparte, aislado, bien identificado.

Cuando se asumió que podíamos conocer a priori la cantidad de géneros (como en el caso del presente trabajo, en el que sabemos que hay cinco géneros) y forzamos un K=5, se encontraron modelos en los que se logró clusterizar de una forma bastante parecida a los grupos de Spotify, pero sacrificando Silhouette, lo que significa que nos fuimos alejando del agrupamiento natural de las canciones, para acercarnos al agrupamiento un poco más arbitrario de Spotify.

VI. CLUSTERING DE SECCIONES DENTRO DE UNA PISTA

A. *Métodos*

Para esta última experiencia, se eligió una de las 2,205 canciones que contienen los datasets, y se aplicó un Clustering Espectral sobre la información segmentada de timbres que se puede encontrar en *Analysis*. Para esto se construyó una Matriz de Recurrencia a la que se le corrió un KMeans sobre sus dos primeros autovectores [3].

Antes de construir dicha Matriz de Recurrencia fue necesario partir los segmentos de forma distinta, interpolándolos para obtener una representación lineal en el tiempo, ya que la serie de tiempo que entrega Spotify no tiene una frecuencia de muestreo constante. Luego de la interpolación, los timbres fueron normalizados con una estandarización z-score y ahora sí entonces se pudo computar la matriz de distancias, utilizando la función `pdist` de `scipy` con la métrica `cosine`.

A continuación se realizaron algunas transformaciones más usando el paquete `librosa`. Se especificó la cantidad de vecinos a considerar (k=100) y se transformaron los valores de disimilaridades a afinidad. Finalmente, se suavizó la matriz

para obtener un resultado más robusto, realzando las diagonales en el sentido temporal.

Una vez obtenidos los dos primeros autovectores de la matriz y habiendo ejecutado KMeans, se graficó la Matriz de Recurrencia suavizada, coloreando los clusters obtenidos con el fin de reconocer las distintas secciones que se repiten en el audio. Para validar que el resultado fuera correcto se siguió segundo a segundo este gráfico escuchando el tema en Spotify.

La canción elegida fue *Tanguedia III*, de *Astor Piazzolla* [4].

B. Resultados

En la Fig. 12 se puede observar la Matriz de Recurrencia de la canción elegida de *Tanguedia III*. El tema tiene una duración de 4 minutos 39 segundos y puede ser escuchado en https://open.spotify.com/album/6NNvUtOmrKH7DlRUfnbx7d.

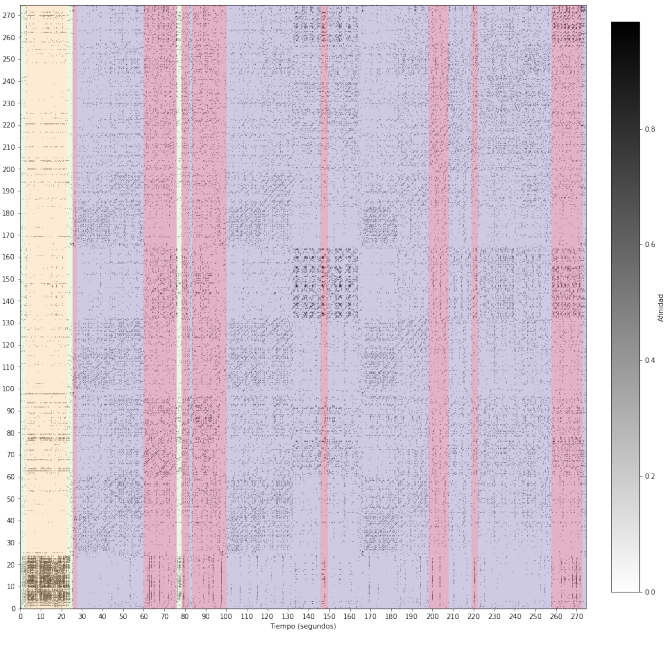

Fig. 12. Análisis Espectral de la pista *Tanguedia III*, de *Astor Piazzolla*

A continuación se describen cada una de las secciones de la pista de audio y se detalla la explicación de los puntos de afinidad más intensos y el significado de los colores de cada cluster.

- **0s - 22s:** Una introducción de voces masculinas, coros y risas. Ese barullo que se ve en el cluster amarillo claramente son esas voces y risotadas.
- **22s:** Hay un breve silencio y comienza el bandoneón. El cluster violeta arranca más o menos a los 27 segundos.
- **1m (60s):** Cambia bastante el ritmo, entra un piano y un violín y la música se acelera. Se entra en el cluster bordó.
- **1m 32s (92s):** Un silencio. El ritmo tranquilo comienza en 1m 35s.
- **1m 40s (100s):** Se vuelve al cluster violeta porque volvió el ritmo tranquilo.
- **2m 12s (132 s):** En la matriz de distancia se ve cómo empiezan a rasguear los violines bien fuerte (puntos de afinidad intensa), mientras sigue el cluster violeta del bandoneón más o menos tranquilo.
- **2m 27s (145s):** Intromisión muy esporádica del cluster bordó hasta los 2m 30s (150s). Es que el violín sube de intensidad y entra el piano por un momento. El ritmo ya no es tranquilo.
- **2m 42s (162s):** Se vuelve a tener un silencio.
- **2m 46s (166s):** Retoma el ritmo tranquilo de bandoneón característico del cluster violeta. Se puede ver que la matriz de distancia se calma, baja la afinidad.
- **3m 18s (198s):** El cluster violeta sigue su patrón en el que una vez más va subiendo de intensidad, hasta que la intensidad es tal que vuelve a entrometerse el cluster bordó, hasta los 3m 28s (208s). Aquí hay un cambio brusco en la melodía donde destaca un violín desde los 3m 21s (201s).
- **3m 28 (208s):** Aunque el violín sigue y hay unos tambores, el ritmo no es tan acelerado (es más bien pesado) y la canción vuelve a caer temporariamente en el cluster violeta hasta los 3m 38s (218s).
- **3m 38s (218s):** Una ráfaga de cluster bordó.
- **3m 42s (222s):** Se vuelve al cluster violeta. El ritmo no es tranquilo. Por eso se oscila entre el cluster violeta y el bordó, y va subiendo la intensidad.
- **4m 17s (257s):** Se acerca el final, de vuelta en el cluster bordó, el ritmo se acelera. A los 4m 20s, el bandoneón se acelera con notas nuevas. En la matriz de distancia se ven puntos salpicados de afinidad.
- **4m 40s (270s):** Final. Último estertor y última franja de silencio de cluster violeta.

VII. Conclusiones

El Estadístico de Hopkins nos indicó, al comienzo del trabajo, que efectivamente las 2,205 canciones que analizaríamos tenían una alta tendencia al clustering. Con lo cual esperábamos poder hallar grupos bien definidos aplicando los algoritmos elegidos. Y fue lo que encontramos, primero con KMeans, luego con Jerárquicos Aglomerativos y, por último, con Clustering Espectral.

En cada una de las tres experiencias se comenzó optimizando los hiper-parámetros propios de cada algoritmo, más otros parámetros fijos como el dataset a utilizar (*Features*, *Analysis*, *Tracks*) y el método de normalización a aplicar sobre los datos (normalización de máximos y mínimos o estandarización z-score). En todos los casos se trató de asumir la menor cantidad de supuestos posibles. En casos como el de Clustering Espectral se eligió trabajar sólo con un subconjunto de hiper-parámetros de UMAP, los que la documentación dice que son los más comunes, ya que sino hubiera alargado mucho los tiempos de las optimizaciones.

A medida que avanzamos por las experiencias, la cantidad de hiper-parámetros fue complejizando los métodos de optimización y selección de los mismos. Para KMeans la selección se llevó a cabo de forma manual, generando unos pocos modelos con distintos K y seleccionando por inspección visual aquel que maximizara el Silhouette promedio y minimizara

el SSE. Para el caso de los Jerárquicos, se ejecutó un Grid Search con el fin de maximizar el Coeficiente de Correlación Cofenético. Se detectó por inspección visual de los árboles generados que aquellos mejores modelos no eran útiles, ya que agrupaban prácticamente todas las canciones en un solo cluster, y se siguió probando hasta encontrar uno que dividiera los datos un poco mejor. Para la última experiencia, la del Clustering Espectral, donde ya teníamos siete hiper-parámetros para optimizar y un Grid Search hubiera sido demasiado costoso desde el punto de vista computacional, se probó usando distintas Optimizaciones Bayesianas, minimizando los indicadores de validación externa, terminando con un experimento que forzó el K=5 para ver qué podía obtenerse.

Los resultados fueron variados, pero a grandes rasgos se encontró que la cantidad de grupos naturales de canciones está más cerca de tres que de cinco. Cinco son los géneros clasificados por Spotify y cuando forzamos los algoritmos para que intenten agrupar en esa cantidad podemos aumentar las coincidencias con dichos géneros, pero no sin sacrificar la estabilidad de los clusters (Coeficiente de Correlación Cofenético para los Jerárquicos y Silhouette para KMeans).

Dependiendo del dataset y dependiendo del algoritmo, en ciertos casos conviene normalizar los datos con el método de máximos y mínimos y en otros con una estandarización z-score. Con K-Means, que usa distancias euclídeas, en general conviene más máximos y mínimos. Con los Jerárquicos Aglomerativos también. En cambio, cuando se está usando KMeans sobre la salida de una reducción de dimensionalidad generada con UMAP, hay algunos casos en los que conviene más usar la estandarización z-score.

Con respecto a las métricas para construir las Matrices de Distancia se probaron varias opciones y aquellas que salieron recurrentemente en los mejores modelos fueron `canberra` y `euclidean`. Sin embargo, el mejor modelo se obtuvo con Clustering Espectral y la distancia de `manhattan`.

En líneas generales el mejor algoritmo que funcionó fue el de Clustering Espectral, reduciendo dimensionalidad con UMAP y luego aplicando KMeans encima. El peor fue el Aglomerativo. Fue muy difícil lograr un modelo decente con un Clustering Jerárquico Aglomerativo.

En lo que respecta a los grupos naturales, se concluye que el género `drum-and-bass` es el único que se logra clasificar correctamente, prácticamente sin ningún error. En la Matriz de Confusión de la Tabla VI se puede ver que el clustering sólo se equivoca en 11 canciones y acierta en 440 con este género. En cambio, la clasificación con los demás grupos es más difusa. Aunque claramente hay un cluster que termina agrupando `ambient` y `classical` por un lado y `jazz` y `world-music` por el otro, en mayor o menor medida.

Con respecto al último experimento, el del Análisis Espectral del tema de Piazzolla, no hay mucho para concluir, salvo que es maravilloso cómo esta técnica permite identificar cada uno de los patrones de una canción de forma tan precisa, y de que siempre es un gusto tener una excusa para volver a escuchar al Maestro.

VIII. Referencias

References

[1] Spotify: Audio Features Object: https://developer.spotify.com/documentation/web-api/reference/object-model/#audio-features-object

[2] Tan, Steinbach & Kumar “Introduction to Data Mining“. Capítulo 7: “Cluster Analysis: Basic Concepts and Algorithms“: https://www-users.cs.umn.edu/~kumar001/dmbook/ch7_clustering.pdf

[3] McFee, B., & Ellis, D. (2014). Analyzing Song Structure with Spectral Clustering. In ISMIR (pp. 405-410).

[4] Astor Piazzola, “Tanguedia III“, Album: “Tango: Zero Hour“. Para escuchar en Spotify: https://open.spotify.com/album/6NNvUtOmrKH7DlRUfnbx7d.