

Clasificación jerárquica de géneros musicales

RODOLFO FIGUEROA SORIANO, Universidad Nacional Autónoma de México, México

1. INTRODUCCIÓN

El género musical encapsula mucha información acerca de una canción en una etiqueta fácil de entender. Por ejemplo, con tan solo escuchar las palabras *rock and roll*, lo más seguro es que uno inmediatamente piense en guitarras eléctricas, baterías, música fuerte y estridente, y vocales intensas. Es por esta razón que los géneros se utilizan como un punto de partida para muchos algoritmos de recomendación, ya que las canciones etiquetadas con un mismo género suelen tener características similares, y por lo tanto podrían gustarle a un usuario.

Al día, Spotify procesa más de 60 000 canciones [10]; en un mundo ideal, le asignaría un género a cada una de ellas de manera individual, sin embargo, esta es una tarea claramente imposible, simplemente debido a la cantidad de trabajo asociado. Es más, aún si contase con la mano de obra requerida, está el hecho de que dos personas pueden asignar a la misma canción géneros diferentes, debido a diferencias subjetivas de su percepción. Claramente, esto ilustra la necesidad de una solución automatizada de clasificación de géneros. En este trabajo, intentaremos hacer una primera aproximación a la solución de este problema.

2. MARCO TEÓRICO

2.1. Clasificación multi-etiqueta

Dada una instancia de datos y un conjunto de *clases*, el problema de clasificación multi-etiqueta consiste en determinar a qué clases pertenece. A diferencia de la clasificación multiclasa, no existe un límite superior o inferior para el número de clases a las que puede pertenecer la instancia: es perfectamente válido que pertenezca a todas, o a ninguna.

Podemos visualizar este problema con una situación similar a la que trataremos en este trabajo. Supongamos que tenemos una canción y queremos determinar a qué género pertenece; en el siguiente diagrama se muestran las opciones posibles:

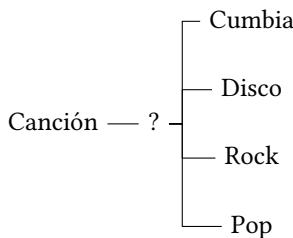


Fig. 1. Ilustración del problema de clasificación multi-etiqueta.

Una posible solución es la de añadir un *clasificador binario* para cada una de las clases. El trabajo de cada uno de estos clasificadores es sencillo: decirnos si la instancia pertenece a su clase o no; a este método se le conoce como *relevancia binaria*.

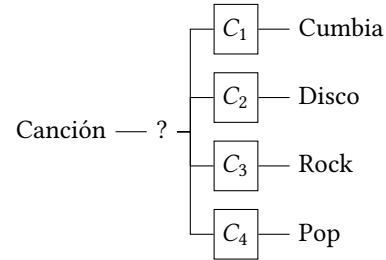


Fig. 2. Ilustración del problema de clasificación multi-etiqueta. Los rectángulos representan clasificadores binarios asignados a cada clase.

Este enfoque esencialmente transforma el problema multi-etiqueta a un conjunto de problemas de clasificación binaria, los cuales son considerablemente más sencillos de resolver, por lo cual es una solución muy utilizada. Sin embargo, esta reducción puede eliminar correlaciones útiles entre las clases, ya que cada clasificador actúa de manera independiente, sin saber las decisiones que tomaron los demás. Por ejemplo, en nuestro caso si C_3 supiese que C_1 dijo que la canción era de cumbia, podría suponer que no es de rock.

Como nota, el método de cadenas de clasificadores [7] busca resolver este problema, aunque su descripción queda fuera del alcance de este trabajo.

2.2. Clasificación jerárquica

2.2.1. *Descripción.* Si bien el método de relevancia binaria presentado en la sección anterior da buenos resultados cuando las clases son pocas y bien definidas, su desempeño se reduce conforme aumenta el número de clases. Esto se debe a que, a menos que el conjunto sea excepcionalmente ideal, entre más aumentan las clases, también lo hacen el número de ejemplos negativos para cada uno de los clasificadores; en otras palabras, el conjunto de datos se vuelve cada vez más desbalanceado. Esto causa que los clasificadores aprendan cada vez menos de los datos, ya que no tienen suficiente información como para hacer predicciones acertadas.

Continuando con nuestro ejemplo previo, supongamos que en nuestro conjunto de entrenamiento había 10 canciones por género. Esto significa que para entrenar a cada uno de los

clasificadores, habrá 10 ejemplos positivos (los correspondientes a su clase) y 30 negativos (los de todos los demás). Esto representa una proporción de 1:3, lo cual no debería de representar un problema para los clasificadores si se selecciona la arquitectura adecuada.

Ahora, supongamos que tenemos los siguientes géneros:

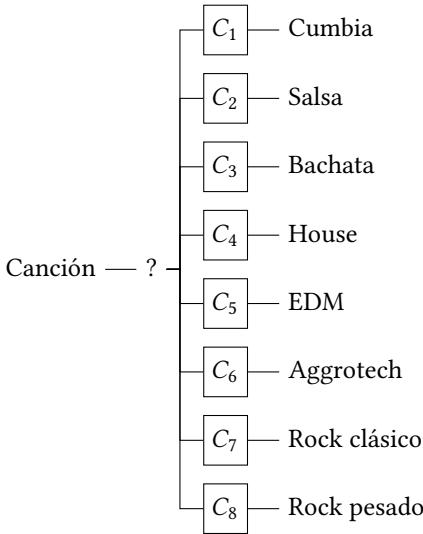


Fig. 3. Clasificación multi-etiqueta con más géneros posibles.

Si seguimos asumiendo que cada género tiene 10 ejemplos, los clasificadores tendrán una proporción de ejemplos positivos-negativos de 1:7, lo cual puede representar serios problemas. En la vida real existen muchos más géneros de los mostrados, y nada garantiza que el número de ejemplos para cada uno sea el mismo, con lo cual nos damos cuenta que el método de relevancia binaria no es suficiente.

Sin embargo, no todo está perdido. Observando los géneros del ejemplo, nos damos cuenta que podemos juntarlos en grupos con características muy similares:

- Ritmos latinos: Cumbia, salsa y bachata
- Música electrónica: House, EDM, aggrotech
- Rock: Rock clásico, rock pesado

Con esta estructura, una manera de clasificar una instancia consiste en primero determinar a qué grupo pertenece, y luego a cuál de los géneros de dicho grupo corresponde. Por ejemplo, si tenemos la canción *Livin' on a prayer* de Jon Bon Jovi, el proceso de decisión sería el siguiente:

1. **Posibilidades:** Rock, electrónica, ritmos latinos.
2. **Decisión:** Rock
3. **Posibilidades:** Rock clásico, rock pesado.
4. **Decisión:** Rock clásico
5. **Conclusión:** Rock clásico

Esta es la idea detrás de la clasificación jerárquica. En vez de considerar a todas las clases como parte de un mismo

conjunto, agrupamos a las que son similares (de acuerdo a algún criterio), y vamos decidiendo nivel por nivel a cuál clase (o clases, en el caso de clasificación multi-etiqueta) pertenece. Naturalmente, necesitamos añadir clasificadores para cada clase de cada nivel que creamos; en el caso de nuestro ejemplo, la nueva arquitectura sería:

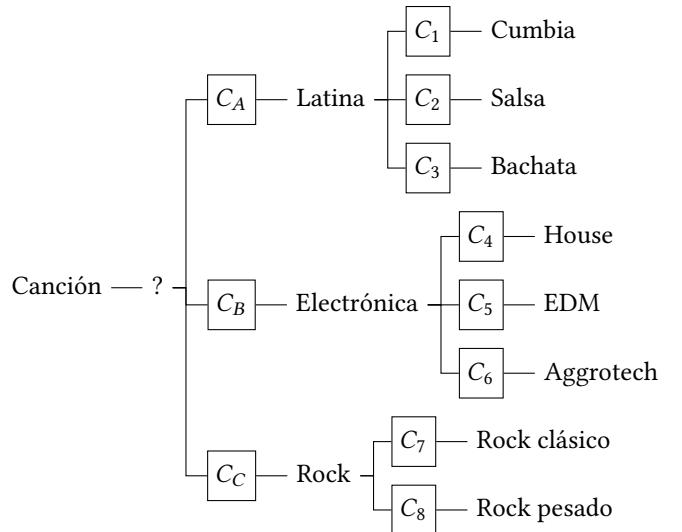


Fig. 4. Clasificación jerárquica.

Existen muchas maneras de escoger cuáles datos tomar como positivos y negativos para el entrenamiento. En nuestro caso, utilizaremos la denominada de *hermanos exclusivos* [8], en el cual los ejemplos positivos son aquellos que pertenecen a la clase y todos sus descendientes, y los negativos las de las clases hermanas y todos sus descendientes. Por ejemplo, suponiendo que queremos entrenar el clasificador C_A (música latina), sus ejemplos positivos serían los de salsa, cumbia y bachata, y los negativos todos los de electrónica y rock. Análogamente, si queremos entrenar a C_1 , sus ejemplos positivos serían los de cumbia, y los negativos los de salsa y bachata.

Esto exhibe una de las principales ventajas de la clasificación jerárquica; al “aislar” unas clases de otras, ayuda a reducir el desbalance de los datos. En nuestro ejemplo, pasamos de una proporción de 1:7 para el caso de C_1 a una de 1:3, lo cual es una reducción considerable. Otro beneficio importante es que, como los clasificadores de niveles más profundos son entrenados con ejemplos pertenecientes a clases muy similares (pero aún así distintos), se ven forzados a aprender características más finas para ser capaces de distinguirlos. Como veremos más adelante, esta “atención al detalle” de los clasificadores puede ayudarnos a explicar lo que hace a las clases diferentes.

2.2.2. Propagación de probabilidades. En nuestro desarrollo anterior, hemos tomado a los clasificadores como cajas negras

con una sola salida: si la instancia pertenece a su clase (1) o no (0). Si bien algunas arquitecturas (máquinas de vectores de soporte, k-vecinos más cercanos) efectivamente hacen esto, están en la minoría. La gran mayoría de clasificadores (regresión logística, árboles de decisión, redes neuronales, etc.) no regresan un valor absoluto, sino la *probabilidad* (p) de pertenecer a la clase. Para transformar esta probabilidad a etiquetas de clase, se compara con un cierto umbral t . Si $p > t$, entonces se determina que sí pertenece a la clase, y si $p \leq t$, entonces no lo hace.

Los clasificadores en nuestro clasificador jerárquico siguen este mismo patrón; cada uno regresa la probabilidad de pertenecer a su correspondiente clase o superclase. Sin embargo, es claro que las probabilidades entre niveles no son independientes; por ejemplo, no tiene sentido que una canción tenga una probabilidad de 0.95 de ser de cumbia si tiene una probabilidad de 0.1 de ser latina. Necesitamos una manera de “combinar” las probabilidades de manera que sean consistentes a lo largo de la jerarquía.

Antes de detallar algunas formas de hacer esto, vale la pena definir un poco de notación:

- \hat{p}_i : Probabilidad producida por el clasificador i , correspondiente a la clase i .
- p_i : Probabilidad de la clase i después de tomar en cuenta la jerarquía.

Una de las maneras más intuitivas de hacer este ajuste de probabilidad consiste en simplemente multiplicar la probabilidad de una clase padre por la de todos sus descendientes. Si la clase i es hija de j , esto se traduce a:

$$p_i = \hat{p}_i p_j$$

Para las clases en el nivel superior (i.e., que no tienen padres), su probabilidad es simplemente la calculada por su clasificador:

$$p_i = \hat{p}_i$$

Otro método se conoce como el de “camino verdadero” [12], y explota la noción que mencionamos previamente: si i es hija de j , no tiene sentido que p_i sea alta si p_j es baja. Para calcularla, el autor define la siguiente notación:

- d_i : Indica si la instancia pertenece a la clase i (1) o no (0). Esto implica que tenemos que comparar p_i con un umbral t .
- ϕ_i : Conjunto de clases hijas de la clase i tales que la instancia pertenece a ellas:

$$\phi_i = \{j | j \in \text{hija}(i), d_j = 1\}$$

Con esto, define la probabilidad global como:

$$p_i = \begin{cases} \hat{p}_i & \text{si } i \text{ no tiene descendientes} \\ \frac{\hat{p}_i + \sum_{j \in \phi_i} p_j}{1 + |\phi_i|} & \text{en otro caso} \end{cases}$$

En nuestro análisis, utilizaremos el primer método, ya que es fácil y rápido de implementar.

3. PRELIMINARES

3.1. Objetivo

Diseñar un sistema de reconocimiento de géneros musicales. Este debe de:

- dar buenos resultados.
- ser extendible a un número arbitrario de géneros.
- ser interpretable.

El último punto es especialmente importante. Como nuestro conjunto de datos es novedoso, no tiene sentido enfocarnos en desarrollar un clasificador “perfecto” (en la medida de nuestras posibilidades). Al contrario, deberíamos de entrenar clasificadores que tomen en cuenta la mayor cantidad de características posibles, y luego analizar su comportamiento para determinar qué es lo que hace a cada género único.

3.2. Alcances

Hasta donde sabemos, el etiquetado de géneros en las plataformas de *streaming* musical principales (Spotify, Apple Music, etc.) se hace de manera manual. Esto significa que un anotador humano tiene que ir canción por canción, escuchándolas y asignándoles los géneros que mejor crea la representan. Debido al volumen de canciones nuevas, esto es una tarea imposible, así que lo que suelen hacer es asignar los géneros a nivel álbum o artista, y asumir que todas las canciones lo comparten.

Si bien esta es una primera aproximación decente, claramente podemos mejorarla. Un sistema completamente automatizado podría producir mejores resultados, de manera más rápida y sin necesidad de intervención humana.

3.3. Marco estratégico

3.3.1. *Conjunto de datos*. El conjunto de datos utilizado fue recopilado por nosotros, utilizando código disponible en [2]. En el apartado 4.1 describimos de manera más detallada sus componentes, pero cabe remarcar que contiene datos de artistas, canciones y los géneros asociados a cada uno.

3.3.2. *UMAP*. UMAP (Uniform Manifold Approximation and Projection) [6] es un algoritmo de reducción de dimensionalidad no-lineal. Primero intenta aprender la estructura de la variedad en la que se encuentran los datos, y después trata de proyectarla a otra variedad de dimensión menor, preservando la información topológica esencial.

Los hiperparámetros de mayor interés son:

- Número de dimensiones de los puntos proyectados.
- Número de vecinos que el algoritmo considera para intentar entender la estructura de la variedad.

- Separación mínima que los puntos proyectados pueden tener.

3.3.3. HDBSCAN. HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) [5] combina un método de agrupación basada en densidad (DBSCAN) con agrupación jerárquica; de esta manera, puede detectar *clusters* de formas arbitrarias. Otra característica importante es que no fuerza a todos los puntos a pertenecer a un *cluster*; si determina que una instancia está demasiado lejos de cualquiera, la clasifica como ruido.

Los hiperparámetros más importantes son:

- Tamaño mínimo de los *clusters*.
- Mínimo de muestras a seleccionar para construir los *clusters*. Entre más muestras haya, más “conservador” será el modelo, y clasificará a más puntos como ruido.
- Número de *clusters* deseados. Este hiperparámetro en realidad fuerza a otro hiperparámetro a cambiar hasta obtener el resultado deseado.

3.3.4. XGBoost. XGBoost (eXtreme Gradient Boosting) [1] es un algoritmo para construir bosques aleatorios utilizando un método conocido como *impulso de gradiente*. Los árboles de decisión tienen muchas ventajas sobre otros clasificadores:

- Aceptan muchos tipos de variables (numéricas, ordinales, categóricas) sin necesidad de preprocesarlas (en la mayoría de los casos).
- Son muy fáciles de interpretar con las herramientas adecuadas.

3.3.5. Valores SHAP. Los valores SHAP (SHapley Additive exPlanations) [4] son un concepto derivado de la teoría de juegos para explicar qué rol juegan cada una de las dimensiones en la predicción de un modelo. En pocas palabras, el valor SHAP de una característica tiene las siguientes propiedades.

- La magnitud cuantifica qué tanto influye la característica en la salida del clasificador. Valores SHAP con valores absolutos más grandes tienen mayor influencia.
- Si el valor tiene signo positivo, significa que esa característica “convence” al clasificador de regresar una respuesta positiva (en nuestro caso, que sí pertenece al género).
- Si tiene signo negativo, lo convence de regresar negativo (que no pertenece).

Como veremos más adelante, para un clasificador, el valor SHAP de una característica varía de instancia a instancia.

4. DESARROLLO

4.1. Integración de la información

4.1.1. Variables totales. Nuestro conjunto de datos consiste de dos partes principales. La primera es una lista de todos los

artistas, con los atributos mostrados en la tabla 1. Esta tiene un total de 228 287 entradas, con 5554 géneros únicos.

Nombre	Tipo	Rango
ID	Alfanumérico	-
Nombre	Alfanumérico	-
Seguidores	Entero	$[0, \infty)$
Géneros	Lista	$[0, \infty)$ (Tamaño)

Tabla 1. Atributos de los artistas.

La segunda componente es una lista de 21 802 240 canciones, con las propiedades de la tabla 2.

Nombre	Tipo	Rango
ID	Alfanumérico	-
Título	Alfanumérico	-
Álbum	Alfanumérico	-
Artistas	Lista	$[1, \infty)$ (Tamaño)
# de disco	Entero	$[1, \infty)$
# de pista	Entero	$[1, \infty)$
Duración	Real	$[0, \infty)$
acousticness	Real	$[0, 1]$
danceability	Real	$[0, 1]$
energy	Real	$[0, 1]$
explicit	Entero	$\{0, 1\}$
instrumentalness	Real	$[0, 1]$
key	Entero	$[0, 11]$
liveness	Real	$[0, 1]$
loudness	Real	$[0, 1]$
mode	Entero	$\{0, 1\}$
speechiness	Real	$[0, 1]$
tempo	Real	$[0, \infty)$
time signature	Entero	$[3, 7]$
valence	Real	$[0, 1]$

Tabla 2. Atributos de las canciones.

Describimos en más detalle algunas de las columnas de esta última, basándonos en la documentación oficial de Spotify [9]:

- *Acousticness*: Qué tan probable es que la pista sea acústica.
- *Danceability*: Qué tan viable es la pista para bailar, tomando en cuenta diversos elementos musicales como tempo, estabilidad e intensidad del ritmo, y regularidad en general.
- *Energy*: Medida perceptual de la intensidad y actividad. Pistas energéticas usualmente se sienten rápidas y ruidosas. Por ejemplo, metal pesado tiene una alta energía,

- mientras que un preludio de Bach aparece más bajo en la escala. Algunas de las características perceptuales que contribuyen a este atributo son el rango dinámico, qué tan ruidosa es, el timbre, y la entropía en general.
- *Explicit*: Si la pista tiene letras explícitas (para adultos) o no.
 - *Instrumentalness*: Predice si una pista no tiene vocales. Sonidos de “ooh” y “ahh” se tratan como instrumentos. Rap o música hablada son “vocales”.
 - *Key*: La clase de tono en la que está escrita la pista, utilizando notación de clases [3].
 - *Liveness*: Detecta la presencia de una audiencia en la pista.
 - *Loudness*: El volumen promedio de la pista, en dB.
 - *Mode*: Modalidad de la pista, mayor (1) o menor (0).
 - *Speechiness*: Detecta la presencia de palabras habladas en la pista. Entre más exclusivamente hablada sea (i.e., grabaciones de radio o audiolibros), más cercana estará a 1.
 - *Tempo*: Tiempo promedio de la pista, en BPM.
 - *Time signature*: Compás de la pista, en múltiplos de notas negras.
 - *Valence*: Describe la positividad transmitida por la pista. Pistas con valencia alta suenan más positivas (i.e., alegres, eufóricas), mientras que una valencia baja suena más negativa (triste, depresiva, enojada).

4.1.2. Limpieza. Una vez removidos los artistas con información faltante, realizamos los siguientes pasos adicionales:

- Removemos todos los artistas sin géneros registrados, ya que nuestro análisis posterior depende crucialmente de dichos géneros.
- Removemos todos los artistas con menos de 1000 seguidores, para evitar introducir ruido al modelo.

Por otro lado, para las canciones, removemos todas las que tuviesen información faltante, y las que no tuviesen ningún artista en nuestra lista de artistas después de hacer la limpieza del paso previo.

5. MODELADO

Nuestro proceso de puede dividirse en tres secciones principales:

1. Detección de jerarquías
2. Clasificación
3. Interpretación

A continuación, las explicamos más a detalle.

5.1. Detección de jerarquías

5.1.1. Método. Para poder aplicar clasificación jerárquica, es necesario tener una *taxonomía* de los géneros, muy similar a la mostrada en el apartado 2.2. Si bien en teoría es posible

hacer esto de manera manual, esto no es viable debido a la gran cantidad de géneros que tenemos.

Una manera “natural” de encontrar la taxonomía sería utilizando algún algoritmo de *clustering*: cada uno de los *clusters* sería una clase padre (a las cuales llamaremos *supergéneros*), mientras que todos sus constituyentes serían sus clases hijas. Sin embargo, si intentásemos trabajar con los géneros directamente (por ejemplo, promediando los valores de todas las canciones que pertenecen a ellos y después agrupando estos puntos), lo más probable es que obtengamos resultados mediocres. Esto se debe a que al promediar todas las canciones de un género estamos reduciendo miles (o hasta millones) de puntos a uno solo, con lo cual se pierde la distribución de las características individuales.

Tomando esto en cuenta, proponemos el siguiente método para determinar la taxonomía:

1. Agrupamos las canciones **por artista** y promediamos sus dimensiones, obteniendo un punto para cada artista.
2. Agrupamos estos puntos, asignando un supergénero a cada *cluster* obtenido.
3. Cada artista tiene uno o más géneros asociados. Contamos cuántas veces aparece cada género en cada *cluster*.
4. Asignamos a cada género al *cluster* en el que aparezca más veces.

Consideramos que este método provee un balance adecuado entre sensibilidad y rendimiento, ya que, en general, las canciones de un artista no suelen estar muy separadas de su media.

5.1.2. Reducción y agrupamiento. Despues de promediar las canciones de los artistas, tenemos puntos en 10 dimensiones, cada uno correspondiente a un artista diferente. Debido a la maldición de la dimensionalidad, es muy complicado encontrar *clusters* significativos, así que nuestra primera tarea es proyectarlos a un espacio de menores dimensiones, procurando preservar las “similitudes” entre los puntos. Una vez proyectados, el siguiente paso es agrupar para encontrar los supergéneros.

Realizamos este proceso utilizando los algoritmos de UMAP y HDBSCAN, descritos en el ???. Hacemos *grid search* exhaustiva sobre el conjunto de hiperparámetros de ambos algoritmos, tomando en cuenta las siguientes métricas:

- Índice de validez de *cluster* basado en densidad (DBCV). En pocas palabras, un valor más grande indica que nuestro *cluster* representa de mejor manera la estructura de puntos original.
- Tamaño mínimo, máximo y promedio de los *clusters*. Idealmente, todos los *clusters* deberían tener tamaños con órdenes similares; después de todo, no se puede

- extraer mucha información de un cluster con 200 000 puntos y otro con 10.
- Número de puntos asignados como ruido. Si los hiperparámetros de HDBSCAN hacen que sea demasiado conservador, clasificará a demasiados puntos como ruido, y por lo tanto no tendremos suficientes datos para analizar.

5.2. Clasificación

Una vez obtenidos los *clusters*, tendremos una taxonomía de géneros/supergéneros. Para limitar el desbalance, solo consideramos las canciones pertenecientes a los n géneros más comunes en cada uno de los *clusters*, con lo cual obtenemos un total de nk clases diferentes, donde k es el número de supergéneros obtenidos.

5.2.1. División de datos. Con esta información, el primer paso es generar los conjuntos de entrenamiento y prueba para los clasificadores **tomando en cuenta la taxonomía generada**. No podemos enfatizar suficiente la importancia de esta consideración; si se hace una simple división aleatoria, se perderá la proporción de datos entre las clases, lo cual puede degradar el rendimiento de los clasificadores.

Entonces, debemos de estratificar de acuerdo a la taxonomía. Para el caso multi-etiqueta esto no es una tarea sencilla, sin embargo, podemos utilizar la librería de Python `scikit-multilearn` [11] para automatizar este proceso. Seleccionamos una proporción de entrenamiento-prueba del 80-20.

5.2.2. Entrenamiento y evaluación. Para los clasificadores, utilizamos una arquitectura de XGBoost. Para optimizar sus hiperparámetros, empleamos optimización Bayesiana con 5-fold cross-validation, y entropía cruzada como función objetivo. Escogemos los ejemplos positivos y negativos utilizando la estrategia de hermanos exclusivos, descrita en el apartado 2.2.

Para la evaluación, comparamos tres clasificadores distintos:

- Clasificador sin información multi-etiqueta.
- Clasificador jerárquico.
- Clasificador jerárquico con superclásificadores perfectos.

Consideramos el tercer clasificador por dos razones; la primera es para realizar una prueba de ablación: queremos ver cómo cambia el resultado de la clasificación cuando se remueve una de las partes de la arquitectura.

La segunda proviene de un enfoque más orientado al negocio: podríamos imaginar un modelo híbrido en el cual un anotador humano clasifica la canción en supergéneros, y luego le pasa la información a nuestro clasificador para que determine a qué géneros pertenece. La razón por la que esto

podría ser un enfoque válido es que, si bien es relativamente difícil distinguir entre géneros si estos son muy similares, distinguir entre *supergéneros* es mucho más sencillo, ya que estos suelen representar canciones con características muy diferentes. Por lo tanto, no tenemos que preocuparnos por que un anotador los confunda.

Por otro lado, utilizamos dos métricas de evaluación de clasificación:

- Precisión promedio, que es simplemente el área debajo de la curva precisión-sensibilidad. Si se tiene una lista de precisiones P y sensibilidades R para distintos valores de un umbral t (tomados en orden descendente de 1 a 0), esta puede calcularse con la regla del paralelogramo:

$$AP = \sum_n (R_n - R_{n-1})P_n$$

- Error de cobertura. Para entender cómo funciona esta medida, hay que recordar que después de hacer todos los ajustes necesarios, la salida de nuestro clasificador dada una instancia será la probabilidad de pertenecer a cada uno de los géneros. Todas estas probabilidades están en un rango de $[0, 1]$, así que podemos utilizarlas para crear un *ranking* de géneros, de modo que el más probable esté en el primer lugar, y el menos probable en el último.

El error de cobertura mide entonces qué tantos elementos de este *ranking* tenemos que tomar para garantizar que cubrimos todas las etiquetas reales. Por ejemplo, supongamos que el clasificador regresó las probabilidades de la tabla 3.

Género	Probabilidad (%)
Cumbia	90
Salsa	80
Rap	65
Bachata	60
Metal	30
Rock	20
Pop	10

Tabla 3. Ejemplo de resultados de clasificación. Los géneros verdaderos se muestran en negritas.

En este caso, para obtener todos los géneros verdaderos, tenemos que reportar los primeros 4 elementos del *ranking*. Un clasificador perfecto habría asignado a los tres primeros lugares los géneros correctos, con lo cual sólo tendría que reportar 3

El error de cobertura expresa de cierta manera qué tan confiados estamos de las predicciones de nuestros clasificadores. Viéndolo desde un punto de vista de negocio,

Clasificación jerárquica de géneros musicales

si tenemos un error mínimo significa que todos los géneros que le reportamos al usuario son correctos, y no le estamos dando información incorrecta o inservible.

5.3. Interpretación

Después de obtener los mejores clasificadores, y asumiendo que reporten buenos resultados, podemos utilizar los valores SHAP descritos en el apartado 3.3.5 para intentar explicar el método de decisión que siguen.

6. RESULTADOS

Dividimos este apartado de la misma manera que el anterior.

6.1. Detección de jerarquías

Antes de proyectar, estandarizamos los puntos. Debido a limitaciones de recursos computacionales, tuvimos que proyectar los puntos a 2 dimensiones, utilizando los siguientes hiperparámetros:

- Número de vecinos: 60
- Separación mínima: 0

En la figura 5 se muestran los resultados de esta proyección.

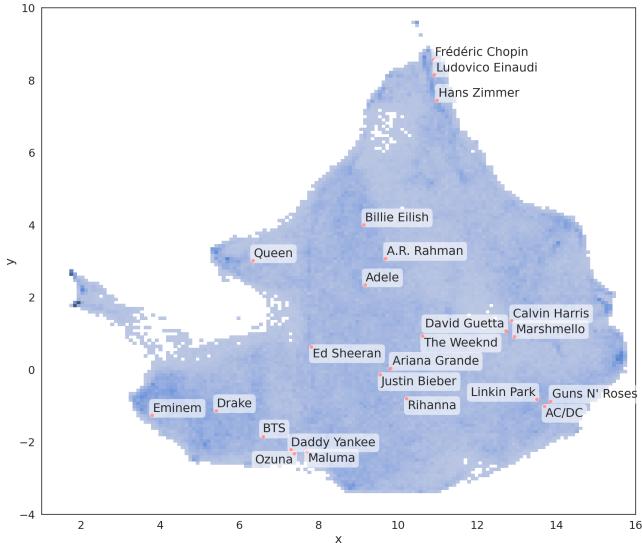


Fig. 5. Artistas proyectados, se muestran algunos artistas populares.

Desde este punto, podemos apreciar algunos patrones en la distribución de los puntos; por ejemplo, cerca de la esquina superior derecha se encuentran los artistas clásicos (Chopin, Zimmer, Einaudi), mientras que en la esquina inferior derecha están algunos artistas de rock y metal (AC/DC, Metallica, Guns n' Roses). Esto nos dice que este *embedding* en pocas dimensiones sigue preservando las complejas relaciones entre los géneros.

Una vez proyectados, agrupamos utilizando HDBSCAN, con los hiperparámetros:

- Tamaño mínimo: 10 000
- Mínimo de muestras: 1
- Número de *clusters*: 7

Los resultados pueden apreciarse en la figura 6. Algunas estadísticas de este agrupamiento pueden verse en la tabla 4.

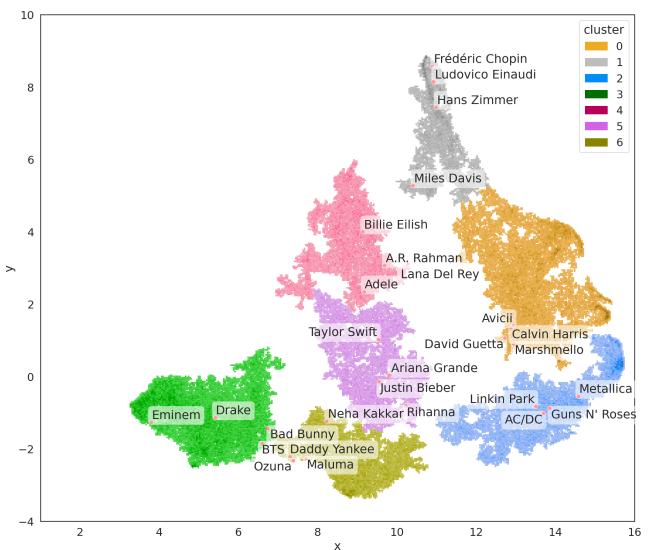


Fig. 6. Artistas coloreados por *cluster*. No se muestran los asignados como ruido.

Puntos asignados a un <i>cluster</i>	154155
Puntos calificados como ruido	68930
Tamaño máximo	31020
Tamaño mínimo	11763
Tamaño promedio	22022

Tabla 4. Estadísticas del agrupamiento resultante.

Observando la figura, podemos ver que nuestra intuición original parece ser correcta, cada uno de los *clusters* tiene artistas que comparten géneros; por ejemplo, el *cluster* verde tiene artistas de rap y reggaetón, el gris a los artistas clásicos, y el azul a los de rock y metal.

Asignando los géneros de acuerdo a la manera descrita en el apartado 5.1.1, podemos obtener los géneros más comunes en cada uno para darnos una idea de qué supergénero representan. Gráficas de estas distribuciones pueden verse en el apéndice A. Con esta información, podemos asignar un nombre a cada supergénero; estos se muestran en la tabla 5

Número	Nombre	Géneros comunes
0	Electrónica	house, EDM, electrónica
1	Instrumental	clásica, jazz, ambiental
2	Rock	rock, metal
3	Rap	rap, hip hop
4	Indie	folk, indie
5	Pop	pop, dance pop
6	Latina	corrido, cumbia

Tabla 5. Nombres asignados a cada supergénero. Se muestran algunos de sus géneros constituyentes.

6.2. Clasificación

Evaluando los tres clasificadores sobre el conjunto de prueba, obtenemos los resultados de la tabla 6.

Modelos	Precisión promedio	Error de cobertura
Vanilla	0.6060	3.396
Jerárquico	0.6229	3.305
Jerárquico c/hint	0.7501	2.035

Tabla 6. Resultados de los distintos clasificadores. El error de cobertura mínimo es 1.504.

Podemos ver que los clasificadores jerárquicos tienen mejores resultados que el vainilla. Adicionalmente, si calculamos la precisión promedio para cada uno de los géneros (véase material adicional), podemos analizar más a detalle el desempeño para cada caso.

Observando las gráficas generadas, notamos que el clasificador jerárquico sin hints tiene, en algunas ocasiones, un peor desempeño que el vainilla. Lo más seguro es que en estos casos, tanto el clasificador como el superclasificador tuvieron un error grande, el cual creció aún más al propagarse para obtener la probabilidad final. El desempeño del clasificador con hints (que, por diseño, no depende del superclasificador) soporta nuestra hipótesis; esto ilustra la importancia que tienen los superclasificadores para la clasificación final.

6.3. Análisis

Para entender cómo funcionan los valores SHAP, lo más sencillo es ver cómo cambian a lo largo del proceso de clasificación de una canción. En la figura 7 podemos ver esto para el clasificador de **rap**, y canción *Fight the power*, del grupo *Public Enemy* (que está etiquetada como hip hop). Esta gráfica ilustra cuánto cambia el valor SHAP total de la instancia cuando el clasificador “ve” cada una de sus características. Por ejemplo, al ver que era explícita, su valor SHAP incrementó por 3.52, mientras que al ver que tenía un tempo de 105.979 BPM su valor disminuyó por 0.01. Esto nos dice que, para

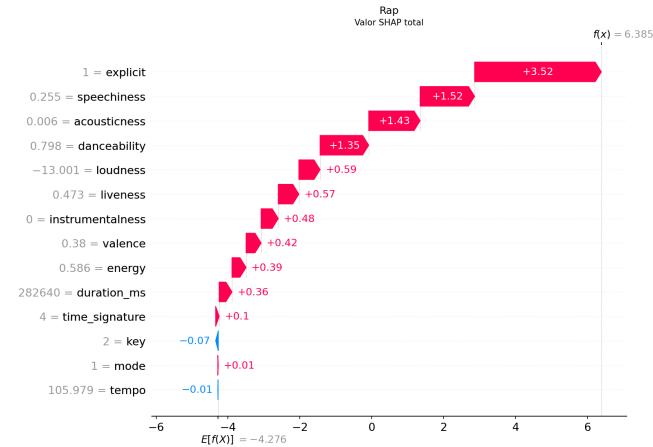


Fig. 7. Cambio de valores SHAP asociado a cada característica del clasificador de rap, para la canción *Fight the Power*.

esta instancia, el clasificador asigna mucha más importancia a la dimensión de “explícito” que a la de “tempo” (ya que la magnitud del valor SHAP de la primera es mayor que la segunda).

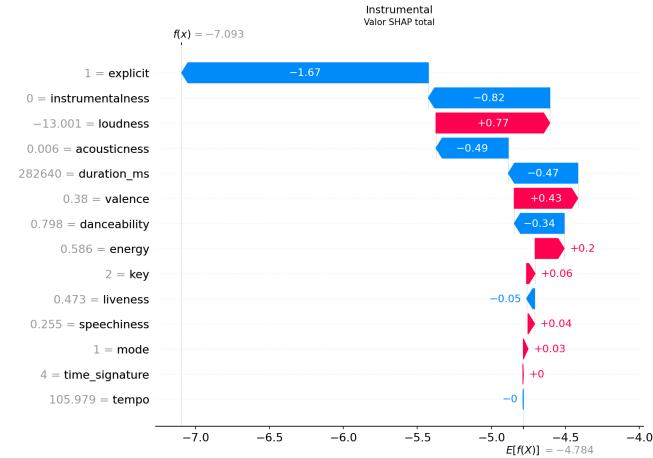


Fig. 8. Cambio de valores SHAP asociado a cada característica del clasificador instrumental, para la canción *Fight the Power*.

Si repetimos el análisis anterior para la misma canción, pero esta vez para el clasificador **instrumental**, obtenemos resultados muy diferentes (figura 8), ya que en este caso la respuesta del clasificador es negativa. Esto ilustra el hecho que mencionamos previamente: los valores SHAP varían entre clasificador, característica e instancia; en algunos casos una cierta característica puede convencer a un clasificador que la instancia sí pertenece a su clase, y en otros que no lo hace.

Clasificación jerárquica de géneros musicales

Podemos entonces graficar la distribución de los valores SHAP de cada característica para todas las instancias del conjunto de prueba de cada uno de los clasificadores, para darnos una idea del efecto promedio que tienen. Para el clasificador de **rap**, esta gráfica puede verse en la figura 9

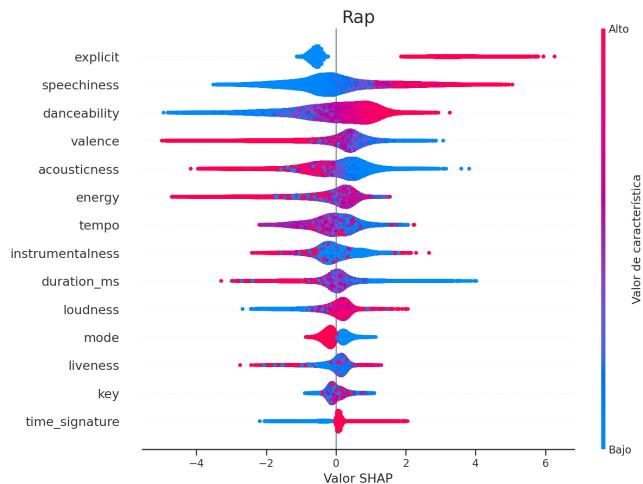


Fig. 9. Distribución de valores SHAP para el clasificador de rap. Las características están ordenadas de mayor a menor influencia.

La manera de leer esta gráfica es la siguiente:

- Cada punto representa una instancia diferente.
- El color del punto representa el valor de la característica.
- La posición horizontal representa el valor SHAP asociado.
- Cada nivel vertical es una característica diferente; se añade un ligero ruido vertical a cada nivel para evitar que los puntos se empalmen.

Podemos derivar muchas conclusiones interesantes de esta gráfica. Para empezar, podemos ver que la característica en la que más se fija el clasificador para determinar si una canción es de rap es si es explícita o no; el grupo de puntos azules del lado negativo indica que si no es explícita, en la mayoría de los casos el clasificador se “convence” ligeramente de que no es de rap, ya que los puntos están muy cerca del cero. Por otro lado, el grupo rojo a la derecha nos dice que el hecho de sí ser explícita convence al clasificador mucho más fuertemente de que sí es de rap.

Pasando ahora a la *speechiness*, en este caso tenemos una característica continua, así que los puntos pueden tomar muchos colores diferentes. Sin embargo, podemos notar que la distribución se organiza en orden creciente de manera natural: los puntos con un valor bajo se encuentran del lado negativo, y entre más nos desplazamos hacia la derecha mayores son sus valores asociados. En otras palabras, un valor

de *speechiness* bajo convence al clasificador de que la canción no es de rap, mientras que uno alto le dice que sí lo es.

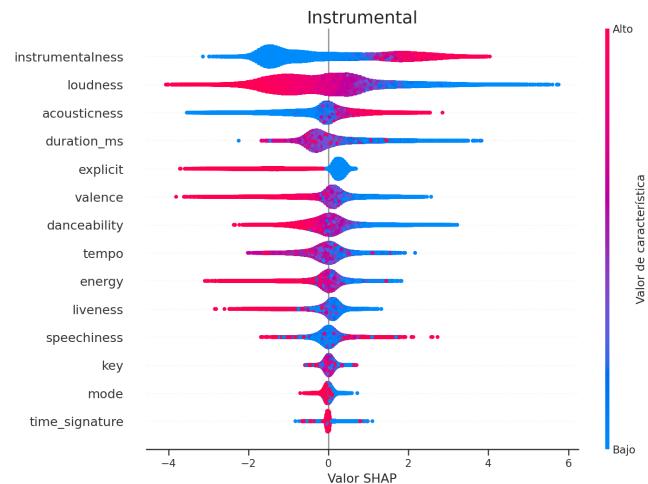


Fig. 10. Distribución de valores SHAP para el clasificador instrumental. Las características están ordenadas de mayor a menor influencia.

Podemos repetir este proceso para cualquier otro clasificador, así que seleccionamos el **instrumental** (figura 10). En este caso, podemos notar que la característica a la que asigna más importancia es *instrumentalness*, lo cual suena redundante. Sin embargo, hay que recordar que nosotros le asignamos el nombre **instrumental** después de hacer un análisis heurístico de sus géneros constituyentes, mientras que *instrumentalness* es una característica que describe qué tan instrumental es una canción. En otras palabras, el clasificador aprendió por su cuenta a asociar ambos.

Otra manera de visualizar es imaginando a las canciones como puntos movedizos en el espacio que van “desplazándose” de característica a característica. Cada vez que pasan por una, se desvían hacia la izquierda si su valor SHAP total disminuye (i.e., si el clasificador se convence del negativo) o a la derecha si aumenta (positivo). Si graficamos cómo cambia la trayectoria de muchos puntos en cada paso, es posible que notemos algún patrón.

Hacemos esto entonces para el clasificador de **rap**, la gráfica resultante se muestra en la figura 11. Inmediatamente, podemos notar algunos patrones muy interesantes: por ejemplo, si observamos la esquina superior izquierda, podemos ver que algunos puntos que estaban muy a la izquierda (es decir, que el clasificador ya se había convencido de que eran negativos) experimentaron un cambio brusco de trayectoria al pasar por la característica de *explicit*, lo cual confirma el hecho de que el clasificador le asigna mucha importancia a esta.

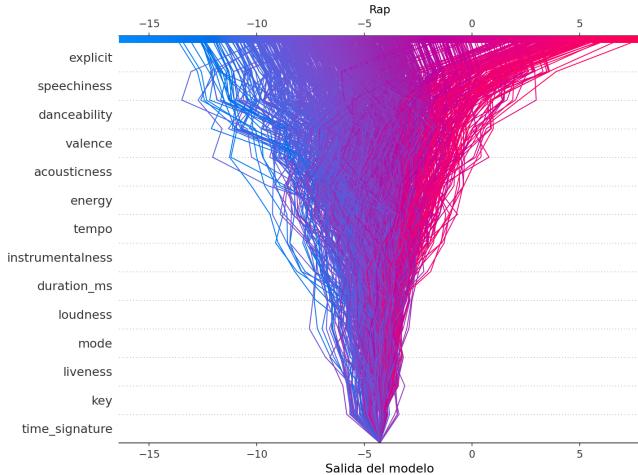


Fig. 11. Trayectorias de 1000 puntos para el clasificador de rap. Las características están ordenadas de mayor a menor influencia.

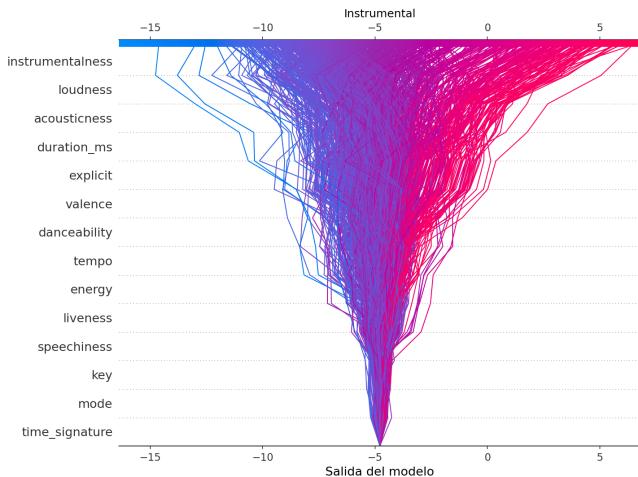


Fig. 12. Trayectorias de 1000 puntos para el clasificador de rap. Las características están ordenadas de mayor a menor influencia.

Repetiendo esta gráfica para el clasificador **instrumental** (figura 12), podemos observar algunas líneas muy separadas del grupo principal; estas pueden ser posibles *outliers*.

Gráficas para cada superclásificador pueden verse en el apéndice B. En el material adicional se incluyen las de los clasificadores individuales.

7. CONCLUSIONES

A pesar de haber hecho una reducción muy drástica de dimensiones (de 10 a 2), la proyección y el *clustering* resultantes siguen preservando mucha información de la distribución de los géneros, lo cual nos permitió obtener una jerarquía que explicaba de manera correcta las relaciones entre cada

uno de los géneros. Al utilizar esta jerarquía para la tarea de clasificación, pudimos notar un incremento en el desempeño de los clasificadores en comparación al caso sin información jerárquica, aunque el resultado fue menor al esperado debido a errores cometidos por los superclásificadores. Esto ilustra la importancia que debe dársele a estos, ya que sus errores se propagan por toda la jerarquía.

A pesar de esto, al realizar el análisis del proceso de decisión de algunos clasificadores, pudimos observar que aprendían de manera excelente distintas características de cada uno de los géneros y supergéneros, lo cual nos ayudó a explicar algunas de sus propiedades más distintivas.

8. TRABAJO FUTURO

Consideramos que este proceso aún tiene mucho por explorar. Primero que nada, la taxonomía derivada depende en gran medida de los hiperparámetros utilizados para la proyección y el agrupamiento. Si bien probamos varias combinaciones de estos, nada garantiza que nuestros resultados sean los mejores posibles, por lo cual sería interesante ver qué otras jerarquías pueden derivarse, lo cual se traduciría en clasificadores completamente diferentes.

Por otro lado, vale la pena recordar que el propósito de este trabajo no era hacer un clasificador perfecto, sino explorar el conjunto de datos de manera indirecta con ayuda de los clasificadores y valores SHAP. Por tanto, el preprocesamiento y selección de características realizado fue mínimo, ya que queríamos proveer a los clasificadores con la mayor cantidad de información posible, y dejar que aprendiesen a discriminaria. Podría repetirse el proceso detallado en este trabajo seleccionando las características que se consideren más importantes (las gráficas de valores SHAP obtenidas podrían servir como un buen punto de partida), para diseñar clasificadores con un desempeño superior. Asimismo, esto podría revelar aún más relaciones entre las características y los géneros.

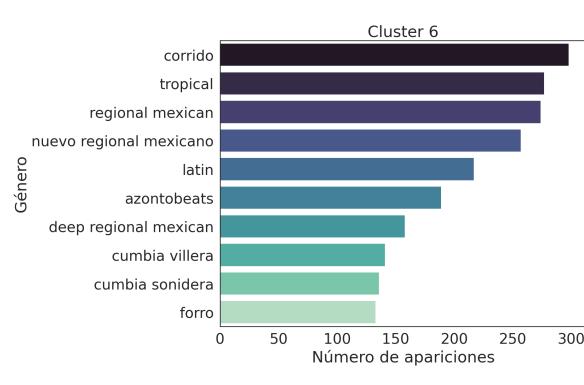
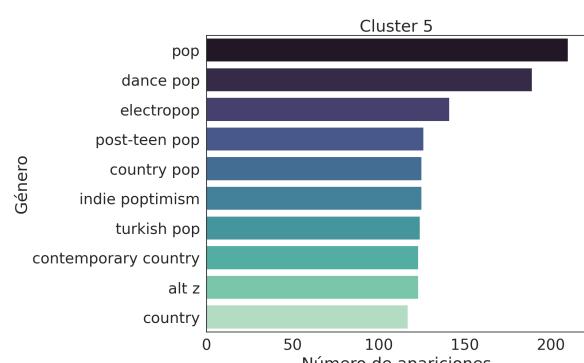
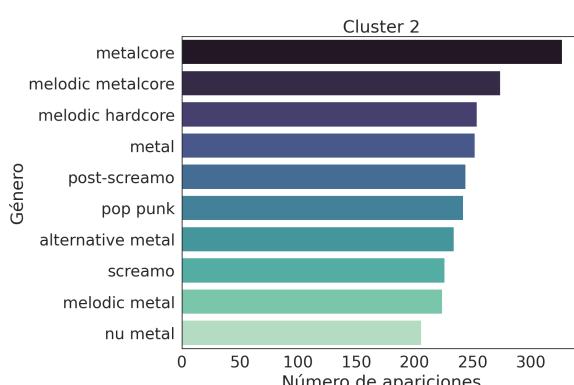
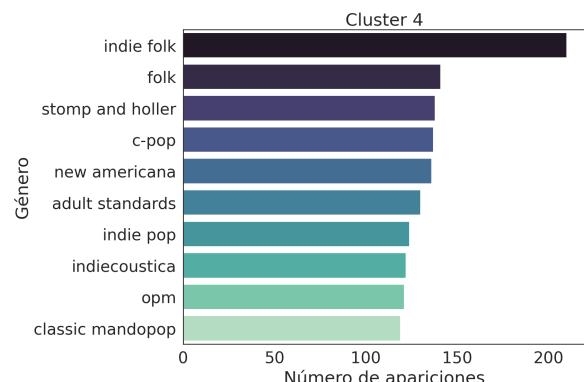
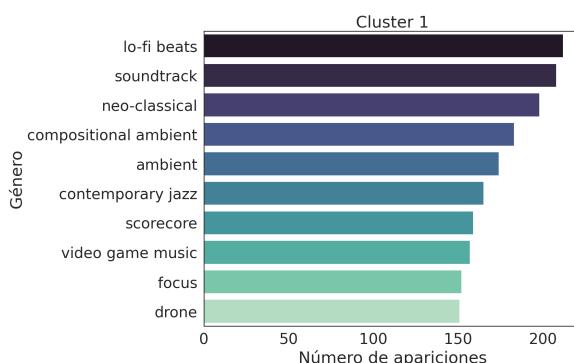
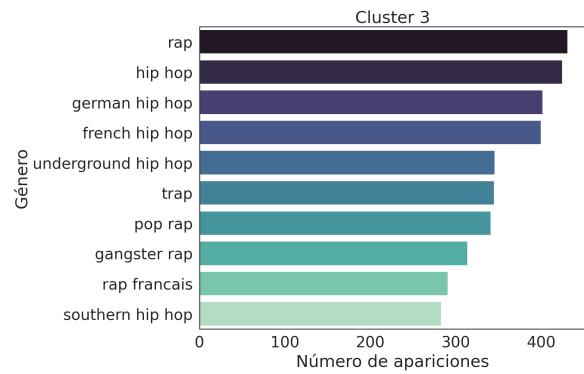
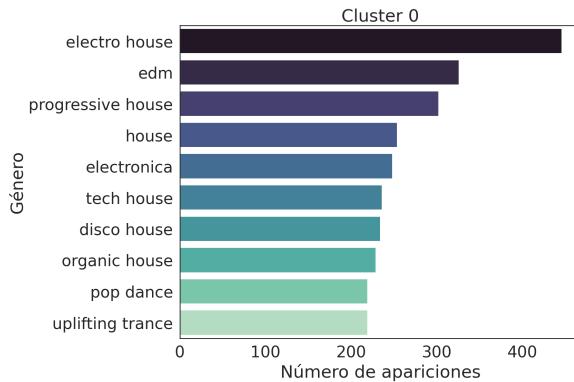
REFERENCIAS

- [1] CHEN, T., AND GUESTRIN, C. XGBoost: A scalable tree boosting system. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2016), vol. 13-17-Aug-16, ACM, pp. 785–794.
- [2] FIGUEROA, R. spotify-scrapers.
- [3] Ito, J. P. Lecture Notes on Pitch-Class Set Theory; Topic 1: Set Classes. 1–10.
- [4] LUNDBERG, S. M., AND LEE, S. I. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 2017-Decem. Curran Associates, Inc., 2017, pp. 4766–4775.
- [5] McINNES, L., HEALY, J., AND ASTELS, S. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software* 2, 11 (2017), 205.
- [6] McINNES, L., HEALY, J., AND MELVILLE, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction.
- [7] READ, J., PFAHRINGER, B., HOLMES, G., AND FRANK, E. Classifier chains

Clasificación jerárquica de géneros musicales

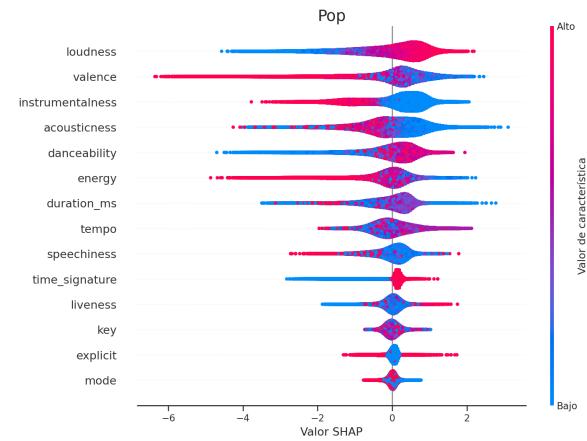
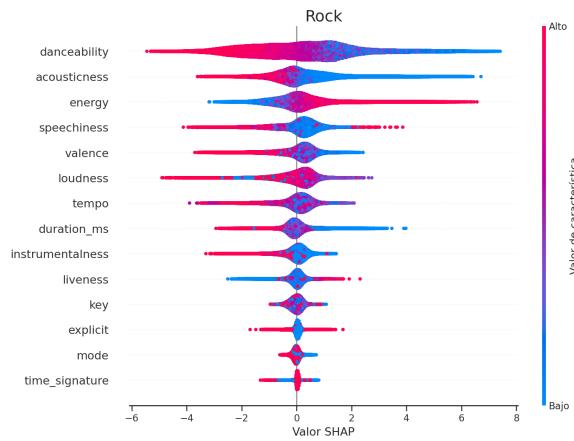
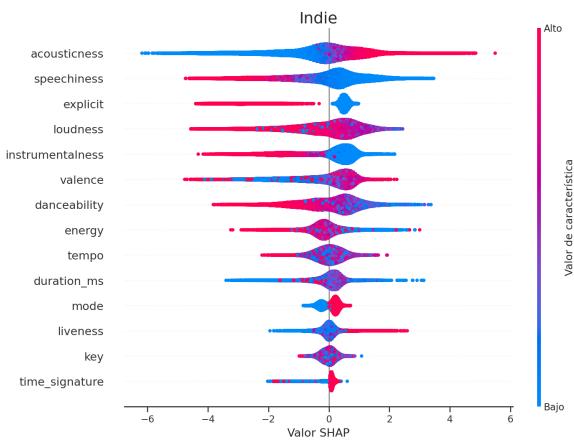
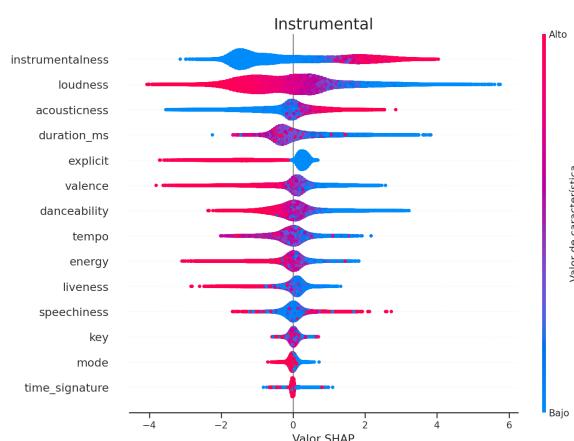
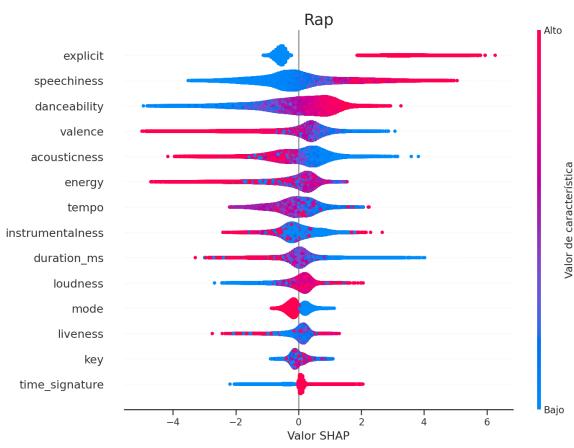
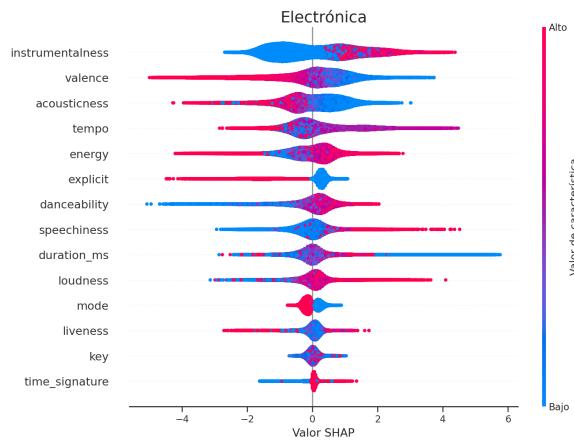
- for multi-label classification. *Machine Learning* 85, 3 (2011), 333–359.
- [8] SILLA, C. N., AND FREITAS, A. A. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery* 22, 1-2 (apr 2011), 31–72.
- [9] SPOTIFY. Spotify Audio Features, 2021.
- [10] SPOTIFY. Spotify Stream On - YouTube, 2021.
- [11] SZYMÁNSKI, P., AND KAJDANOWICZ, T. Scikit-multilearn: A scikit-based Python environment for performing multi-label classification. *Journal of Machine Learning Research* 20 (2019).
- [12] VALENTINI, G. True path rule hierarchical ensembles for genome-wide gene function prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 8, 3 (2011), 832–847.

A. DISTRIBUCIÓN DE GÉNEROS POR CLUSTER

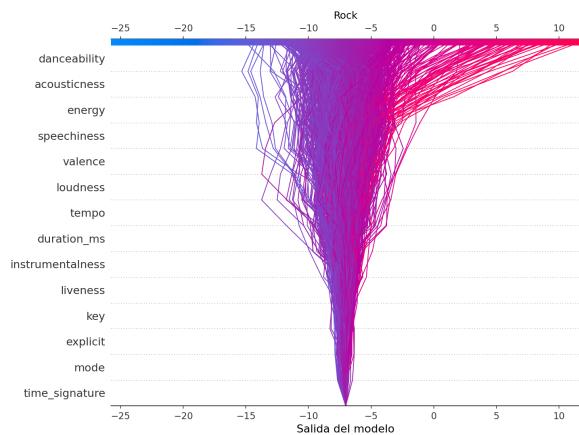
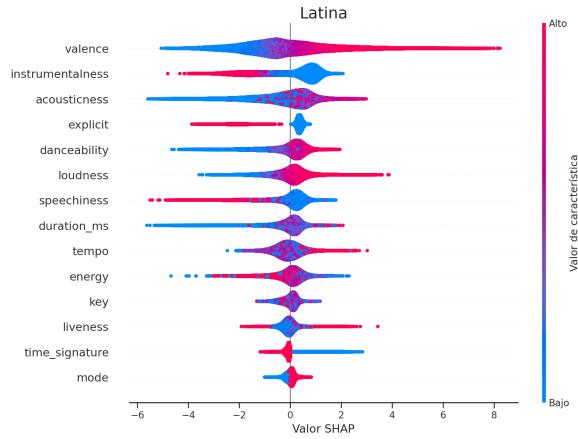


B. VALORES SHAP

B.1. Distribución



Clasificación jerárquica de géneros musicales



B.2. Trayectorias

