Letra, música y clustering

Camila Sanz, Favio Di Ciocco, Daniel Cerini, Lucio García

Resumen

Base de datos

Dataset de música:
Artistas
Popularidad
Canciones
Letras
Lenguaje
Género musical

Objetivos

Clusterizar los artistas en base a las letras de sus canciones.

Analizar los clusters obtenidos.

Métodos

Pre-procesamiento

Normalización TF-IDF

LSA (Reducción de dim)

Clusterización Jerárquica (Agglomerative Clustering)

Dataset y pre-procesado

Dataset: ~2,500 artistas; ~190,000 canciones y ~80 géneros musicales

Fuente: kaggle → vagalume (La música lo es todo)

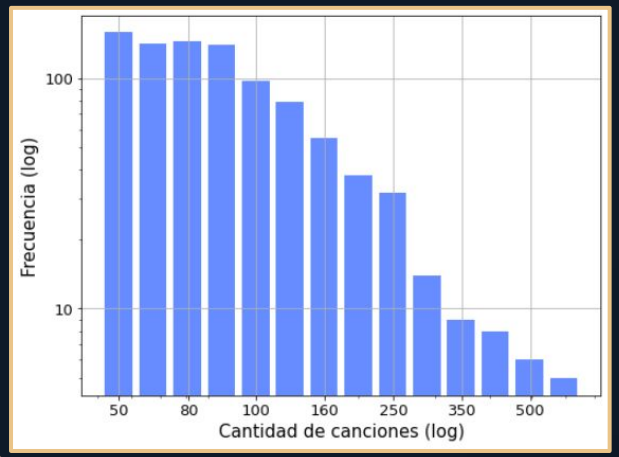
Filtros: Covers (~55,000)
Lenguaje (Spacy): Inglés ('en') + confianza > 0.9
Canciones < 30 palabras
Artistas < 50 canciones

Pre-procesamiento de letras (skit-learn NLTK): Pos-tagging de palabras y lemmas (términos de raíz). Descartamos: puntuación, mayúsculas, dígitos y sw.

Dataset y pre-procesado

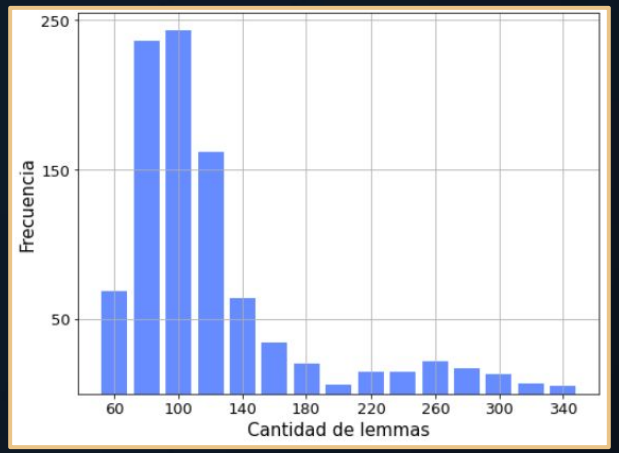
Agrupamos los datos por artistas (~930) juntando los lemmas de sus canciones

Canciones por Artista



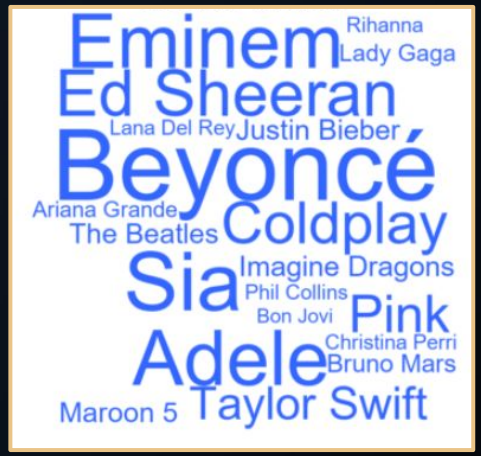
μ=115; Mediana = 90; σ= 80

Lemmas por Artista



μ=120; Mediana = 100; σ= 56

Artistas más populares



Métodos: TF-IDF

Matriz inicial: **X = Artistas x Palabras** → Lemmas de sus canciones

Normalización TF-IDF (Term Frequency - Inverse Frequency):

$$w_d = f_{w,d} \times log\left(\frac{|D|}{f_{w,D}}\right)$$

D = Colección de documentos; |D| = N° de documentos

w_d = Palabra incluída en el documento $d \in D$

$f_{w,d}$ = Frecuencia de w en d (celda original de X)

$f_{w,D}$ = N° de documentos en los que aparece w

Filtramos las palabras que aparecían en menos de 10% (poco comunes) de los documentos y en más de 90% (muy comunes)

Métodos: LSA

Matriz inicial: **X' = Artistas x Palabras'** → Habiendo aplicado TF-IDF

LSA (Latent semantic analysis):

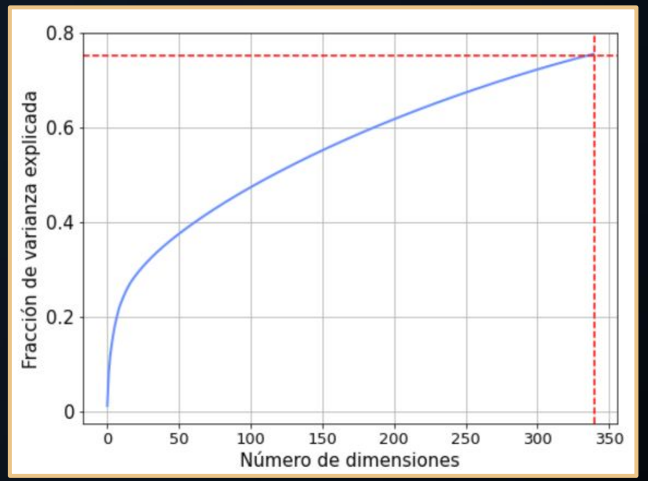
$$X' = U^{*}\Sigma^{*}V^{t}$$

U, V ortogonales y Σ "diagonal"
Descartamos elementos de Σ
(75% de la varianza acumulada)

Matriz final: **X'' = U*Σ'**

(928 x 340)

Varianza acumulada

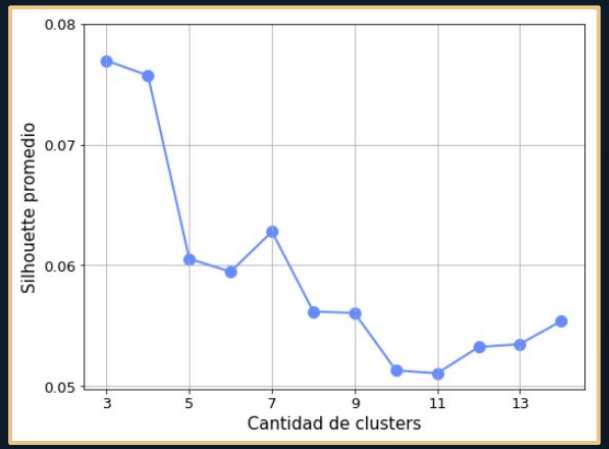


Clustering

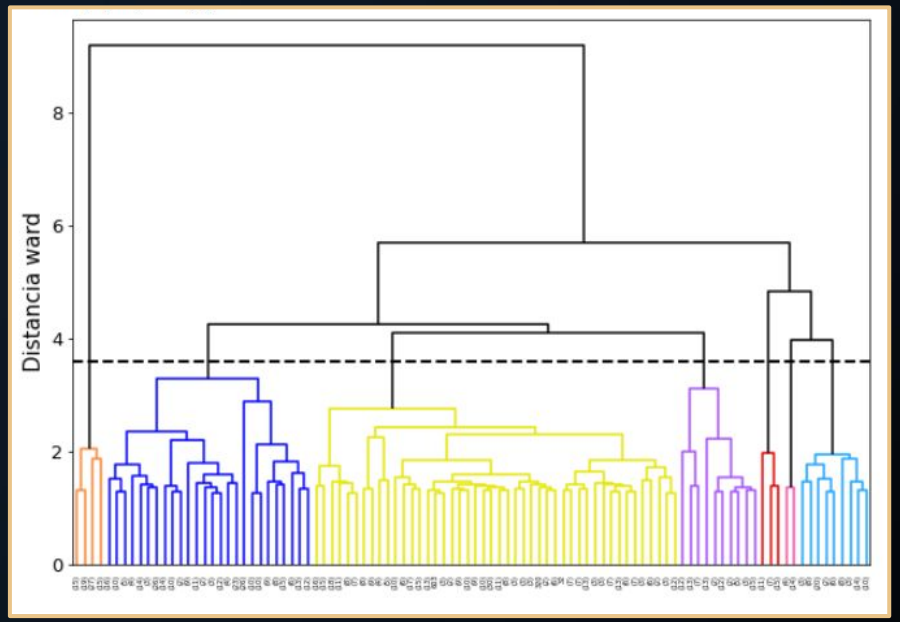
Clustering jerárquico: Agglomerative Clustering (skit-learn)

Affinity: Distancia Euclídea. Linkage: Ward.

Coef. de Silhouette promedio



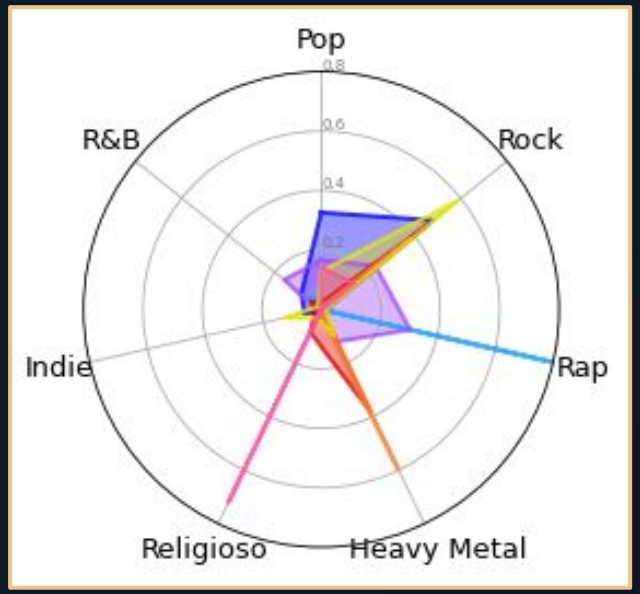
Dendrograma por artista



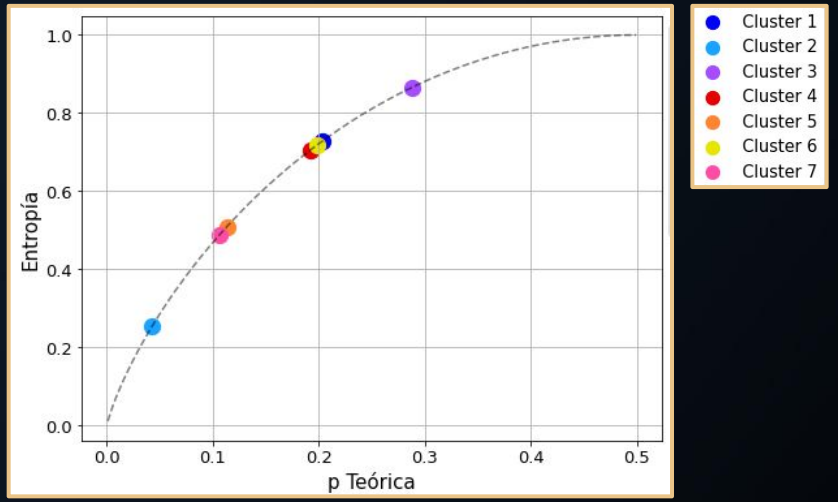
Géneros en clusters

Representación one-hot encoding con géneros + unificación (Hip-Hop+Rap; Rock + Rockabilly + …) + filtro de géneros con pocos artistas.

Proyección en los géneros



Entropía según el género



$$S = -\sum_i p_i \times log(p_i)$$

Gephi 0.9.3

Clustering (grafo)

Nodos: Artistas

Tamaño de nodos: Grado (pesado)

Peso de enlaces: distancia^{-1} (euclídea)

Palabras en clusters

Conclusiones

- El coeficiente de Silhouette promedio no mostró grandes diferencias respecto a la cantidad de clusters.
- Encontramos un sentido en los clusters al analizar los géneros y palabras más frecuentes.
- Algunos clusters (ej. 1 y 3) mostraron una mayor superposición y entropía que el resto.

Gracias!