Universidad Politécnica
de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Grado en Grado en Ingeniería Informática

Trabajo Fin de Grado

Análisis de Preferencias Musicales y Recomendación Personalizada

Autor: Martín Rubio Willems
Tutor: Juan Antonio Fernandez Del Pozo Salamanca

Madrid, Enero - 2025

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado
Grado en Grado en Ingeniería Informática

Título: Análisis de Preferencias Musicales y Recomendación Personalizada

Enero - 2025

Autor: Martín Rubio Willems
Tutor: Juan Antonio Fernandez Del Pozo Salamanca
Departamento de Inteligencia Artificial
Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid

Índice general

Índice de Figuras

Listings

Resumen

Este trabajo se centra en primer lugar en el análisis descriptivo de un banco de datos real perteneciente a la plataforma de Streaming musical Spotify, el cual será desgranado y estudiado para que dicho análisis de patrones en la escucha de los usuarios ayude con la segunda parte del trabajo.

La segunda parte es el desarrollo de un prototipo de sistema de recomendación musical que combina similitudes entre usuarios y características específicas de los datos. En esta fase se hace un clustering previo para facilitarnos las agrupaciones de usuarios que serán beneficiadas de las recomendaciones lo más precisas posibles. El sistema se complementa con una interfaz gráfica funcional dinámica.

El proyecto tratará de aportar mejoras en los sistemas de recomendación y también reflexionará acerca de alguno de los dilemas éticos relacionados con la privacidad de los datos y la "burbuja de contenido".

En conclusión,trata de ofrecer un análisis, una prueba de implementación prototipo y un estudio que siente las bases para futuros aportes en este sector.

Abstract

This work focuses firstly on the descriptive analysis of a real dataset from the Spotify music streaming platform, which will be examined and studied to identify listening patterns among users. These insights will support the second part of the project.

The second part involves the development of a prototype for a music recommendation system that combines user similarities with specific data features. This phase includes a preliminary clustering step to create user groupings that enable highly accurate recommendations. The system is complemented by a functional and dynamic graphical interface.

The project aims to propose improvements to recommendation systems while reflecting on ethical dilemmas related to data privacy and the çontent bubble."

In conclusion, it seeks to provide an analysis, a prototype implementation, and a study that lays the groundwork for future contributions in this field.

Capítulo 1

Introducción y objetivos

En este proyecto, nos centraremos en el análisis de preferencias musicales utilizando un dataset específico de Spotify. Analizaremos patrones de comportamiento de los usuarios con la intención de optimizar la precisión en los gustos de los usuarios en los sistemas de recomendación musical. Utilizaremos Power BI como herramienta para analizar las variables y propondremos mejoras en los sistemas de recomendación basadas los patrones que vayamos descubriendo en el análisis. Estos sistemas fundamentales en plataformas de música digital (fundamentales en plataformas como Spotify o cualquiera que ofrezca contenido) enfrentan aún desafíos relacionados con la repetición de contenido y la falta de diversidad [RRS11a].

A pesar de los algoritmos (muy avanzados a día de hoy) de aprendizaje automático que se van utilizando, Spotify y otras plataformas han convertido la experiencia de escuchar música así como las elecciones personales de música que uno quiere disfrutar. A pesar de ello, no dejan de enfrentarse a retos que siguen presentes como la creación de burbujas de contenido —los usuarios se ven limitados a un círculo cerrado de géneros o canciones [RRS11b]. En este proyecto estudiaremos qué podemos aportar de cara a optimizar las recomendaciones y mejorar la diversidad, sin sacrificar el trabajo ya conseguido.

1.1. Motivación del proyecto

Este proyecto nace del aumento en la influencia de la música en nuestras vidas y la importancia crucial de plataformas como Spotify en la personalización de contenido. Además de los retos previamente mencionados, persiste la falta de exactitud. Debido a los datos obtenidos del dataset de Kaggle en relación a Spotify, contamos con diversas columnas que nos facilitarán entender la correlación entre diferentes variables evaluables al momento de medir una canción.

1.2. Contexto del proyecto

En años recientes, el estudio de grandes volúmenes de datos ha facilitado una mejora significativa en las recomendaciones proporcionadas a los usuarios, empleando métodos sofisticados de aprendizaje automático y big data. Los usuarios persiguen experiencias más ajustadas a sus gustos personales.

Se han investigado varias metodologías, como las redes neuronales y los sistemas de embeddings, que capturan las relaciones complejas entre usuarios y canciones. [RRS11c]. A pesar de que se han implementado exitosamente en áreas como el entretenimiento, aún existe margen para mejoras, en particular en la habilidad de los sistemas para ajustarse a variaciones en las preferencias de los usuarios [RRS11d].

El proyecto toma como referencia datos objetivos, con el fin de contribuir a mejorar la experiencia del usuario utilizando el dataset de Spotify que incluye variables como el género musical, popularidad, características acústicas, y otros metadatos. Además, el estudio examina la transformación de las preferencias musicales con el paso del tiempo, los elementos que afectan al gusto musical, y cómo se puede prevenir la uniformidad en las sugerencias.

Este trabajo está en un ámbito en el que tecnologías en auge como la inteligencia artificial tienen un rol crucial en la creación de sistemas más sólidos y adaptables.

1.3. Objetivos

El propósito principal de este Trabajo de Fin de Grado es examinar patrones en la conducta de los usuarios de plataformas de música, empleando un dataset de Spotify, con el fin de optimizar la personalización de las sugerencias musicales. El objetivo será tratar de aportar mejoras a los sistemas de recomendación actuales, para perfeccionar la experiencia del usuario al generar sugerencias más útiles y variadas.

1.3.1. Objetivos Específicos

Planteamos 6 objetivos específicos relativos a la comprensión de la base de datos y a las propuestas de mejora en las recomendaciones.

1.3.1.1. Segmentación de usuarios

Identificar conjuntos de usuarios con comportamientos parecidos mediante métodos de agrupación, para incrementar la exactitud de las sugerencias personalizadas.

1.3.1.2. Correlación y asociación entre variables

Examinar la correlación entre los géneros musicales más escuchados, las canciones más famosas y elementos como la fecha de emisión o el estado emocional de los usuarios. Más a fondo, las hipótesis y sugerencias de análisis que aspiramos cumplir serán:

-Análisis de Popularidad por Año y Género

• Objetivo: Examinar cómo ha cambiado la popularidad de las canciones a lo largo del tiempo y si ciertos géneros han ganado o perdido relevancia.

• Visualizaciones: Gráficos de líneas para mostrar la evolución de la popularidad por año y gráficos de barras para comparar la popularidad media de diferentes géneros musicales.

• Preparación de los datos: Creamos una columna de año a partir de la fecha de lanzamiento y categorizamos los géneros musicales.

-Análisis de Características de Audio

• Objetivo: Investigar cómo las características de audio (danceability, energy, valence...) influyen en la popularidad de las canciones. Dichas características componen las columnas del dataset, siendo danceability el nivel de bailabilidad de una canción, energy su nivel de energía y valence una característica asociada a la positividad que transmite la canción. Otras características serán explicadas más adelante.

• Visualizaciones: Gráficos de dispersión para ver como se relacionan entre las características de audio y la popularidad, y unos posibles gráficos de barras apiladas si se quisieran segmentar las canciones por rangos de dichas características.

• Preparación de los datos: Normalizamos las columnas de características de audio si es necesario y categorizamos los valores en rangos (por ejemplo, baja, media, alta).

-Análisis por Franjas de Años

• Objetivo: Ver cómo han evolucionado las preferencias musicales en diferentes franjas de años (por ejemplo, 2013-2015, 2016-2018, 2019-2021).

• Visualizaciones: Gráficos de líneas o áreas para mostrar cómo cambian las características clave en diferentes franjas de años.

• Preparación de los datos: Creamos una columna que agrupe los años en franjas.

-Análisis de Canciones Explicitas vs. No Explícitas

• Objetivo: Analizar si las canciones explícitas tienen niveles de popularidad diferentes y si estas tendencias han cambiado con el tiempo.

• Visualizaciones: Gráficos de barras comparativas para mostrar la popularidad promedio de las canciones explícitas frente a las no explícitas.

• Preparación de los datos: Nos aseguramos de que la columna explicit esté categorizada correctamente como "Sí/"No".

-Análisis de Preferencias según el Estado de Ánimo

• Objetivo: Examinar si las canciones con diferentes niveles de valence (positividad) y energy se asocian con ciertos géneros o estaciones del año.

• Visualizaciones: Gráficos de dispersión, gráficos de burbujas y gráficos de barras por estaciones del año.

• Preparación de los datos: Clasificamos valence y energy en categorías como “Bajo”, “Medio”, “Alto” para un análisis más claro.

1.3.1.3. Mejoras en sistemas de recomendación

Proponer mejoras en los algoritmos de recomendación basadas en el análisis de los datos de los que disponemos, con el fin de aumentar la satisfacción de los usuarios.

1.3.1.4. Desarrollo de un dashboard visual

Implementar una interfaz visual que permita visualizar las correlaciones de las variables que elijamos en cada parte del análisis, así como la evolución del gusto musical o la popularidad de géneros a lo largo del tiempo.

1.3.1.5. Recomendaciones basadas en estado de ánimo y actividad

Aportar nuevas formas de recomendación que generen playlists personalizadas en función del estado de ánimo de los usuarios.

1.3.1.6. Diversificación en las recomendaciones

Aportar mecanismos que aumenten la diversidad en las recomendaciones, evitando la "burbuja"de contenido repetitivo para enriquecer la experiencia del usuario.

1.4. Estructura del Documento

- **Capítulo 1: Introducción y Objetivos**

Hemos presentado el contexto del proyecto, así como la motivación detrás del análisis de preferencias musicales y los objetivos que buscamos alcanzar.

- **Capítulo 2: Estado del Arte**

Este capítulo presenta los progresos y metodologías vigentes en la área de los sistemas de recomendación. Enumeramos los distintos tipos de sistemas, las técnicas empleadas, y examinamos estudios actuales y usos prácticos en el sector musical. Además, se debaten las consecuencias éticas y el porvenir de estos sistemas.

- **Capítulo 3: Desarrollo**

En este capítulo, detallamos el procedimiento de preparación, limpieza y tansformación de los datos. Damos una descripción de la herramienta que utilizaremos (Power BI). Y explicamos cómo se estructurarán los datos para llevar a cabo un estudio detallado de la evolución musical a través de intervalos o franjas temporales. Además, presentamos los procedimientos utilizados para identificar, segmentar y visualizar los patrones de conducta de los usuarios.

- **Capítulo 4: Impacto del Trabajo**

Evaluamos el impacto y la magnitud de importancia que puede suponer nuestro análisis en el contexto de los sistemas de recomendación y su aplicación en la industria musical y discutimos cómo el proyecto puede contribuir a un desarrollo más ético en la tecnología.

- **Capítulo 5: Resultados y Conclusiones**

En este capítulo, presentaremos los resultados obtenidos del análisis de datos y las visualizaciones generadas. Reflexionaremos sobre dichos resultados y ofreceremos conclusiones personales sobre la eficacia de los métodos empleados. Además, discutiremos las posibles direcciones de trabajo futuro y mejoras.

- **Bibliografía**

Enumeramos todas las fuentes de información y referencias utilizadas en la elaboración de este trabajo.

Capítulo 2

Estado del Arte

Este capítulo expone las investigaciones, herramientas y métodos actuales que se han utilizado en el área de los sistemas de recomendación y análisis de preferencias musicales.

2.1. Importancia de los sistemas de recomendación

Los sistemas de recomendación son herramientas esenciales en esta era digital en la que vivimos, particularmente en plataformas que gestionan grandes volúmenes de contenido como servicios de música, películas, comercio electrónico y redes sociales. Estos sistemas ayudan a los usuarios en la búsqueda de productos, servicios o contenidos que podrían interesarles de forma personalizada, haciendo más sencilla y eficaz la experiencia del usuario al proporcionar recomendaciones que se adecuan a sus preferencias y necesidades.

2.1.1. Facilitar del Descubrimiento de Contenidos

El crecimiento exponencial del contenido digital viene acompañado de un claro desafío para quienes lo consumimos: encontrar lo que realmente nos interesa en medio de un inmenso mar de opciones. Los sistemas de recomendación ayudan a reducir esta sobrecarga de información al ofrecer recomendaciones relevantes y personalizadas. En plataformas de música, como Spotify[Spo]. Estos sistemas nos permiten descubrir nuevos géneros y artistas que se alinean con nuestras preferencias musicales previas, enriqueciendo la experiencia y siempre trtando de fomentar la exploración de cosas nuevas. [RRS11e]

2.1.2. Incremento la Fidelización y la Retención de Usuarios

La capacidad de los sistemas de recomendación para personalizar el contenido suele aumentar significativamente la fidelización del usuario con la plataforma. Al recibir recomendaciones cada vez más personalizadas, los usuarios tienden a interactuar más frecuentemente con la plataforma que las ofrece, lo que genera un ciclo de retroalimentación donde el sistema aprende más sobre sus preferencias y mejora aún más las sugerencias. [RRS11f].

2.1.3. Impacto en las Ventas y el Comercio electrónico

A nivel de comercio electrónico, los sistemas de recomendación no solo optimizan la experiencia del usuario, sino que también incrementan mucho las ventas. Plataformas como Amazon [Ama] hacen uso de estos sistemas para proponer a los usuarios productos extra o complementarios, incrementando la posibilidad de que efectúen compras adicionales. Este método ha probado ser muy eficaz para incrementar las ventas y promover la lealtad de los clientes. Este aspecto lo estudiaremos más en profundidad cuando reflexionemos sobre la ética detrás de estas técnicas de marketing [RRS11g].

2.1.4. Reducción de la Sobrecarga de Información

En una era donde la cantidad de información es tan grande que abruma y que no es digerible si tuviesemos que buscar por nosotros mismos lo que queremos, los sistemas de recomendación actúan como filtros inteligentes que nos ayudan a no perdernos en tal cantidad de opciones.

2.1.5. Impulso a la Diversidad y la Innovación

Por último, por mencionar los puntos que considero más cruciales a la hora de ser conscientes de la importancia que tienen estos sistemas en nosotros, los sistemas de recomendación tienen el potencial de impulsar la diversidad en el contenido que consumen los usuarios. Cuando están bien diseñados, pueden ofrecer una mezcla equilibrada entre contenido popular y contenido menos conocido o de nicho, permitiendo que los usuarios descubran nuevos intereses y ampliando sus horizontes. ¿Cuántas veces hemos empezado viendo un artículo y hemos acabado en otro nicho completamente distinto el cual no esperábamos visitar pero descubrimos que lo estamos disfrutando? [RRS11h]

2.2. Tipos de sistemas de recomendación

2.2.1. Sistemas Basados en Contenido

Los sistemas basados en contenido analizan las características de los ítems (en nuestro caso, canciones) y las comparan con las preferencias y el historial del usuario. Se utilizan modelos que representan tanto a los usuarios como a los ítems mediante atributos clave. Por ejemplo, como vemos en la subsección 3.1.1.1, las canciones pueden tener características como danceability, energy, y valence, que se comparan con las preferencias del usuario para hacer recomendaciones [Pan23].

Estos sistemas tienen la ventaja de ser interpretables, ya que es fácil entender por qué se ha recomendado un contenido en particular. Aunque como desventaja, son algo limitados al no ser capaces de sorprender al usuario con contenido fuera de sus gustos conocidos, lo que puede resultar monótono y aburrido[HKP11].

2.2.2. Filtrado Colaborativo

El filtrado colaborativo es uno de los métodos más utilizados y se basa en el análisis de patrones y preferencias que comparten múltiples usuarios. Este enfoque puede dividirse en dos tipos:

2.2.2.1. Filtrado Colaborativo Basado en Usuarios

Se analizan los gustos de los usuarios para encontrar otros usuarios con intereses similares. Si un usuario A tiene gustos similares a un usuario B, se recomienda al usuario A contenido que B haya disfrutado [TSK06].

2.2.2.2. Filtrado Colaborativo Basado en Ítems

Este enfoque se centra en los ítems (canciones, en este caso) y sugiere ítems similares a aquellos que el usuario ya ha disfrutado. Se utilizan técnicas como la factorización de matrices y la descomposición en valores singulares (SVD) para optimizar las recomendaciones .[Sci]

2.2.3. Modelos Híbridos

Estos sistemas combinan varios enfoques para generar recomendaciones más precisas. Por ejemplo, al unir el filtrado colaborativo con el sistema basado en contenido, se pueden superar las limitaciones de cada uno, logrando sugerencias más completas y personalizadas.

2.3. Metodologías y aplicaciones prácticas en la industria

2.3.1. Modelos de Aprendizaje Profundo y Redes Neuronales

El *deep learning* ha revolucionado por completo a los sistemas de recomendación, ya que somos capaces de capturar patrones complejos en grandes volúmenes de datos. Las redes neuronales convolucionales (CNN) y recurrentes (RNN) son ampliamente utilizadas en plataformas de música o de contenido cinematográfic.

Las CNN, por ejemplo, pueden analizar características visuales, pero en nuestro caso (Spotify) son las RNN las que son útiles en el análisis de secuencias de interacciones previas para predecir comportamientos futuros.

Están diseñadas para manejar datos secuenciales, se utilizan cuando es importante considerar el orden o la secuencia de interacciones del usuario. Un claro ejemplo sería Spotify, que emplea RNN para analizar el historial de escucha de los usuarios, considerando no solo las canciones escuchadas, sino también el orden en que se escuchan y los tiempos de reproducción.

Luego, esta información será utilizada para crear listas de reproducción más personalizadas como *Daily Mix* o *Discover Weekly*, donde el sistema puede predecir qué canciones o géneros le gustarán más al usuario en función de sus patrones de escucha anteriores.

Lo mismo ocurre con Amazon y las compras complementarias, que no deja de ser una estrategia de marketing muy potente basada en estas herramientas. [RRS11i].

2.4. Evaluación de los sistemas de recomendación

Evaluar la efectividad de los sistemas de recomendación es importante para cerciorarnos de que los usuarios están recibiendo el contenido que demandan de una forma

precisa. Existen distintas formas de evaluarlos como:

2.4.1. Métricas

Las métricas de precisión como el error absoluto medio (MAE), accuracy, recall y la métrica F1 se utilizan para evaluar la precisión (accuracy) en las recomendaciones respecto de las preferencias reales de los usuarios. [RRS11j].

- **Error absoluto medio (MAE)**

Mide el promedio de las diferencias absolutas entre las predicciones del sistema y los valores reales. En nuestro caso, el valor real es la calificación que un usuario da a una canción, mientras que la predicción es la calificación que el sistema de recomendación anticipa.

El MAE se utiliza con calificaciones numéricas (por ejemplo, de 1 a 5 estrellas) y nos da una medida aproximada de la precisión de las predicciones. Un MAE bajo indica que el sistema de recomendación predice con mayor accuracy.

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{2.1}$$

Dónde:

- N es el número total de predicciones.
- y_i es la calificación real del ítem i.
- $\hat{y}_i$ es la calificación predicha para el ítem i.

- **Accuracy**

Es una métrica para determinar evaluar si un usuario interactuará con la recomendación o no. Mide la proporción de recomendaciones relevantes (aquellas que el usuario realmente disfrutó o con las que interactuó) sobre el total de recomendaciones generadas por el sistema.

$$\text{Accuracy} = \frac{\text{Recomendaciones Relevantes}}{\text{Total de Recomendaciones}} \tag{2.2}$$

- **Recall**

Es una métrica que mide la capacidad que tiene el sistema de almacenar todas las recomendaciones que fueron relevantes para un usuario. Un alto valor de recuerdo indicaría que el sistema es capaz de captar la mayoría de artículos que le interesan al usuario.

$$\text{Recall} = \frac{\text{Recomendaciones Relevantes}}{\text{Total de Ítems Relevantes}} \tag{2.3}$$

Dónde:

- Recomendaciones Relevantes es la cantidad de recomendaciones que el usuario considera relevantes.

- Total de Ítems Relevantes es el número total de ítems relevantes que podrían haber sido recomendados.

.

Capítulo 3

Análisis Descriptivo del Dataset

3.1. Importación y preparación de los datos

3.1.1. Presentación del conjunto de datos

Para este proyecto se ha seleccionado un conjunto de datos que se encuentra disponible en la web de Kaggle [RRS11k] por su riqueza en variables y la cantidad de información que ofrece acerca de las canciones más populares desde 2013 en Spotify.

La elección de este conjunto de datos está principalmente basada en la adecuación a los objetivos planteados y que permite un análisis descriptivo predictivo y de segmentación de dicho conjunto.

3.1.1.1. Descripción del dataset

El conjunto de datos incluye 21 columnas con información detallada de las canciones en Spotify. Inicialmente, cuenta con 74.000 líneas, cada una correspondiente a una canción. A continuación, se resumen las variables más importantes:

- **track_id:** Identificador único de Spotify para cada canción.
- **artists:** Nombres de los artistas que interpretaron la canción. Si hay múltiples artistas, están separados por un punto y coma (;).
- **album_name:** Nombre del álbum donde se encuentra la pista.
- **track_name:** Nombre de la canción.
- **popularity:** Valor entre 0 y 100 que mide la popularidad de una pista. Se calcula según la cantidad de reproducciones recientes.
- **duration_ms:** Duración de la pista en milisegundos.
- **explicit:** Indica si la pista contiene contenido explícito (True/False).
- **danceability:** Mide qué tan bailable es una canción. Su valor oscila entre 0.0 (menos bailable) y 1.0 (más bailable).
- **energy:** Una medida entre 0.0 y 1.0 que indica la intensidad y actividad percibida en la canción.

- **key:** Representa la tonalidad musical de la canción, utilizando la notación estándar de clases de tono (0=C, 1=D, 2=E etc.).
- **loudness:** Nivel de volumen promedio de la pista en decibeles (dB).
- **mode:** Indica la modalidad musical: 1 para modo mayor y 0 para menor.
- **speechiness:** Detecta la presencia de palabras habladas en la pista. Los valores altos indican contenido mayormente hablado.
- **acousticness:** Mide si una pista es acústica (0.0 a 1.0).
- **instrumentalness:** Indica la probabilidad de que una canción sea instrumental.
- **liveness:** Representa la probabilidad de que la canción se haya grabado en vivo.
- **valence:** Describe la positividad que transmite una canción. Valores altos reflejan emociones positivas.
- **tempo:** Ritmo de la canción en pulsaciones por minuto (PPM).
- **time_signature:** Compás de la pista (por ejemplo, 4/4, 3/4).
- **track_genre:** Género musical asociado a la canción.

3.1.2. Importación del dataset

El dataset ha sido importado en Power BI [Mic25] utilizando la opción de *Obtener datos* para archivos en formato CSV, lo cual integra todos los datos inicialmente en crudo en el programa. Aunque el archivo de origen sea un .xls (extensión estándar para documentos creados con Microsoft Excel), puede ser fácilmente convertido a formato .csv para obtener algunas ventajas al trabajar en PowerBi como un mayor rendimiento (al ser simplemente texto delimitado, es mucho más ligero y rápido en comparación con los archivos excel).

Además, los archivos CSV no tienen que ser interpretados por Power BI con estilos, celdas combinadas, metadatos complejos o fórmulas, por lo que siempre ocupan menos espacio en disco y eso nos beneficia a la hora de ser eficientes en la transferencia de datos.

3.1.3. Limpieza y preparado de los datos

Una vez cargados los datos, abrimos **Power Query** (pestaña dedicada a transformar los datos). Es un editor de consultas de Power BI donde podemos limpiar y transformar nuestros datos.

Lo primero que hicimos, para ir facilitando la preparación de datos, fue eliminar toda la información que consideraba irrelevante, fueron las columnas **trackid, albumname, key, loudness** y **timesignature** ya que no aportan ningún valor directo al análisis. Así, pasamos de las 21 columnas que teníamos originalmente a tener 16, todas ellas relevantes para el análisis. Sin embargo, dado que es esencial tener la fecha de salida de cada una de las canciones, decidimos, con ayuda de chatGPT-4o [16] añadir una columna **release_date:** Fecha de lanzamiento de la pista, que sería clave para trabajar con franjas de años y conocer mucho mejor los patrones de comportamiento según la fecha.

En esta parte del proceso (Power Query) verificamos los tipos de datos de cada una de las columnas para que se correspondiesen con su naturaleza. Esto es, que los datos numéricos sean números, las letras sean texto, las fechas sean un dato de tipo fecha, etc. Revisar los tipos de datos es fundamental para garantizar que cada columna se interprete de manera correcta por la herramienta.

3.1.3.1. Ajustes adicionales

En esta parte del proceso también ajustamos la duracion de las canciones (columna **duration_ms**) de tal manera que apareciese en un formáto cómodo con el que trabajar, creando la columna **duration_mm_ss** a partir de la original, con la siguiente fórmula):

$$\begin{aligned}\text{Duración (mm:ss)} = \text{Pad}\big(\text{From}\big(\text{IntDiv}(\text{dur\_ms}, 60000)\big), 2, \text{"0"}\big) \\ + \text{" : "} + \\ \text{Pad}\big(\text{From}\big(\text{IntDiv}\big(\text{Mod}(\text{dur\_ms}, 60000), 1000\big)\big), 2, \text{"0"}\big)\end{aligned} \tag{3.1}$$

Donde:

- **Pad**: Equivale a `Text.PadStart`, que añade ceros iniciales al texto hasta alcanzar la longitud especificada.
- **From**: Equivale a `Text.From`, que convierte valores numéricos en texto.
- **IntDiv**: Equivale a `Number.IntegerDivide`, que realiza una división entera entre dos números.
- **Mod**: Equivale a `Number.Mod`, que calcula el residuo de la división entre dos números.
- **dur_ms**: Representa la duración de la canción en milisegundos.

De esta forma, pasamos de tener el tiempo en milisegundos a tenerlo en minutos y segundos, como estamos acostumbrados a ver.

También redondeamos todas las columnas a 2 decimales ya que no es necesaria una precisión de más numeros para el análisis.

Por último, ordenamos todas las columnas según mi criterio en cuanto a relevancia de cada una de ellas con respecto al análisis.

3.1.3.2. Detección de valores nulos o filas duplicadas

No detectamos ningún valor nulo ni campos vacíos en ninguna de las columnas, lo cual fue una ventaja porque no tuvimos que pensar en como tratarlos.

Lo que sí detectamos en cantidad fueron filas duplicadas, muchas se repetían idénticas excepto por el índice que se genera automáticamente (1ª columna). Para esto, directamente en Power Query seleccionamos *Quitar duplicados* y aseguré que cada fila fuese única y apareciese una sola vez para no alterar resultados del análisis.

3.1.3.3. Normalización y estandarización de las variables

Con el fin de que las columnas con valores continuos estuvieran en un rango homogéneo, normalizamos la mayoría de variables como **popularity**, **danceability**, **energy**, **valence**... entre otras. Esto lo realizamos dividiendo los valores de cada columna por su máximo corrrespondiente posible, para escalar los valores entre 0 y 1. Por ejemplo:

$$\text{popularity\_normalized} = \frac{\text{popularity}}{100} \tag{3.2}$$

Esta normalización de todas las variables continuas facilita comparaciones entre columnas, hace que no trabajemos con diferentes escalas y representa de una manera más fiel e uniforme los datos que queremos visualizar.

3.1.3.4. Columnas que no necesitan normalización:

- **explicit:** Es categórica (verdadero/falso).
- **track_genre:** Es categórica.

3.1.3.5. Creación de nuevas columnas

Durante la preparación del dataset, decidimos añadir nuevas columnas a partir de las que tenía, con la intención de facilitar y hacer más preciso el análisis y posteriormente sacar mejores conclusiones relacionadas con patrones de comportamiento de las distintas variables. A continuación se mencionan modificaciones que se han hecho sobre el conjuntos de los datos.

3.1.3.5.1. Categorización de géneros musicales: La columna `track_genre` contenía muchos géneros musicales específicos distintos, demasiados como para no perderse a la hora de saber el tipo de género global al que nos podemos referir. Percibimos que muchos de ellos podrían agruparse dentro de categorías más generales para simplificar el análisis. Creamos una nueva columna denominada `genre_agrupado`, en la que los géneros se agruparon en las siguientes categorías principales:

- **Pop**: Incluye variantes como *pop, synth-pop, indie pop, k-pop, latin pop, pop punk*.
- **Rock**: Agrupa géneros como *rock, alternative rock, hard rock, classic rock, indie rock, punk rock*.
- **Electrónica**: Contiene *electro, house, dance, techno, trance, edm, deep house, ambient, chillout*.
- **Hip-Hop/Rap**: Incluye *hip-hop, rap, trap*.
- **Reggaetón/Latino**: Cubre géneros como *reggaetón, latin urban, cumbia, banda, salsa*.
- **R&B/Soul**: Agrupa *r&b, soul, neo-soul, funk*.
- **Jazz/Blues**: Incluye *jazz, blues, swing, bebop*.
- **Clásica**: Contiene *classical, opera, baroque*.
- **Country/Folk**: Incluye *country, folk, americana, bluegrass*.

- **Otros**: Para géneros que no encajan en las categorías anteriores.

El proceso para realizar esta categorización en Power Query fue el siguiente:

1. **Selección de la columna a categorizar:** Elegimos la columna **track_genre**.
2. **Creación de una nueva columna categorizada:** En *Agregar columna* seleccionamos *Columna personalizada*. y la renombramos **genre_agrupado**.
3. **Aplicación de una lógica condicional:** Utilizamos una fórmula condicional que identifique palabras clave en los géneros y asigne cada género a una de las siguientes categorías: **Pop, Rock, Electrónica, Hip-Hop/Rap, Reggaetón/Latino, R&B/Soul, Jazz/Blues, Clásica, Country/Folk,** y **Otros**. De esta manera:

Listing 3.1: Clasificación de géneros musicales en Power Query

```
if Text.Contains([track_genre], "pop") then "Pop"
else if Text.Contains([track_genre], "rock") then "Rock"
else if Text.Contains([track_genre], "electro") or
    Text.Contains([track_genre], "house") then "Electronica"
else "Otros"
```

4. **Guardamos los cambios aplicados**

3.1.3.6. Extracción del año de la fecha de lanzamiento

La columna `release_date` contenía originalmente fechas completas (formato `YYYY-MM-DD`) y no estábamos siendo capaces de acceder al dato del año para analizar. Así que extrajimos el año de cada fecha para facilitarme el trabajo creando una nueva columna denominada `year`, que almacena únicamente el año de lanzamiento utilizando la función **Date.Year()**

3.1.3.7. Agrupación de años en franjas

Agrupamos los años en las siguientes franjas:

- 2013-2015
- 2016-2018
- 2019-2021
- 2022 en adelante

Dichas franjas las puse en una nueva columna llamada `year_range`.

3.1.4. Reducción del dataset

Siguiendo con la revisión del dataset, fuimos conscientes de que era demasiado grande (80.000 filas) y además contenía datos totalmente irrelevantes para un estudio como el que quería hacer. Había campos tomados como canciones que eran compilaciones de música de hasta 60, 70 u 80 minutos de duración, así como efectos de sonido o fragmentos de piezas de 15, 20 o 30 segundos de duración.

Para garantizar un análisis más limpio y sobretodo relevante, decidimos quedarnos solo con canciones desde 1 minuto y medio a 7 minutos. La columna `duration_mm_ss`

contenía la duración de las canciones en formato `mm:ss`. Para filtrar las canciones que estuvieran dentro del rango de 1:30 (90 segundos) a 7:00 (420 segundos), realizamos los siguientes pasos en **Power Query**:

- Agregamos una columna temporal que convertía el formato `mm:ss` a segundos utilizando la función `Text.Split` en Power Query, que divide el texto por el delimitador ":", extrayendo minutos y segundos, y los combina en un valor numérico en segundos con el siguiente cálculo:

  Listing 3.2: Cálculo de duración en segundos (etapa intermedia)

```
duration_seconds_temp = (minutos * 60) + segundos
```

  La fórmula completa implementada en Power Query es:

Listing 3.3: Fórmula completa en Power Query para calcular duración en segundos

```
let
    parts = Text.Split([duration_mm_ss], ":"),
    minutes = Number.From(parts{0}),
    seconds = Number.From(parts{1})
in
    (minutes * 60) + seconds
```

  Donde:

  - `Text.Split([duration_mm_ss], ":")`: Divide el valor de la columna en dos partes separadas por ":".
  - `parts{0}`: Representa los minutos.
  - `parts{1}`: Representa los segundos.
  - `Number.From()`: Convierte los valores extraídos de texto a números.
  - `(minutes * 60) + seconds`: Calcula el total de segundos.

- Aplicamos un filtro para conservar únicamente las filas donde `duration_seconds_temp` estuviera entre 90 y 420 segundos de la siguiente manera:

  1. En la columna `duration_seconds_temp` seleccioné la opción **Filtro numérico >Entre**.
  2. Especificamos los valores del rango:
     - Valor mínimo: 90 (1 minuto y 30 segundos).
     - Valor máximo: 420 (7 minutos).
  3. Aplicamos el filtro y confirmé los cambios.

  Este paso permitió reducir el dataset a aquellas canciones cuya duración era relevante para el análisis, excluyendo duraciones extremadamente cortas o excesivamente largas.

- Eliminamos la columna temporal `duration_seconds_temp`, dejando únicamente la columna original `duration_mm_ss` en su formato `mm:ss`.

De esta manera ajustamos los datos para solo tener duraciones más convencionales, sin alterar la columna `duration_mm_ss`.

3.1.5. Reducción del dataset con muestreo aleatorio

Dado que el dataset seguía conteniendo 74.000 filas tras el filtro por duración de canciones, consideramos necesario realizar una reducción adicional para manejar mejor los datos y que se procesasen de forma más fluida. Reducirlo aproximadamente a la mitad pero sin que afectase a la representatividad del conjunto y sin que estuviese esta reducción sesgada por nada. Optamos por coger un muestreo aleatorio por ese motivo.

Para ello, utilizamos la función de **Conservar filas superiores** en Power Query siguiendo estos pasos:

- Primero, añadimos una columna aleatoria con la opción **Columna Personalizada** y empleando la función `Number.RandomBetween(1, 100000)` para asignar a cada fila un valor aleatorio.

- Luego, ordenamos las filas según los valores generados en la columna aleatoria, utilizando la opción **Ordenar por** para organizar los datos en base a esta nueva columna.

- Después seleccionamos la opción **Conservar filas** en el menú desplegable de **Quitar filas**.

- En **Conservar filas**, pusimos **40.000**, lo que indica que solo quiero quedarme con las primeras 40000 filas aleatoriamente

- Finalmente, eliminamos la columna aleatoria

El resultado es un conjunto que mantiene las características para el análisis pero que es mas manejable.

3.2. Análisis descriptivo basado en la popularidad

3.2.1. Relación entre características de audio y popularidad

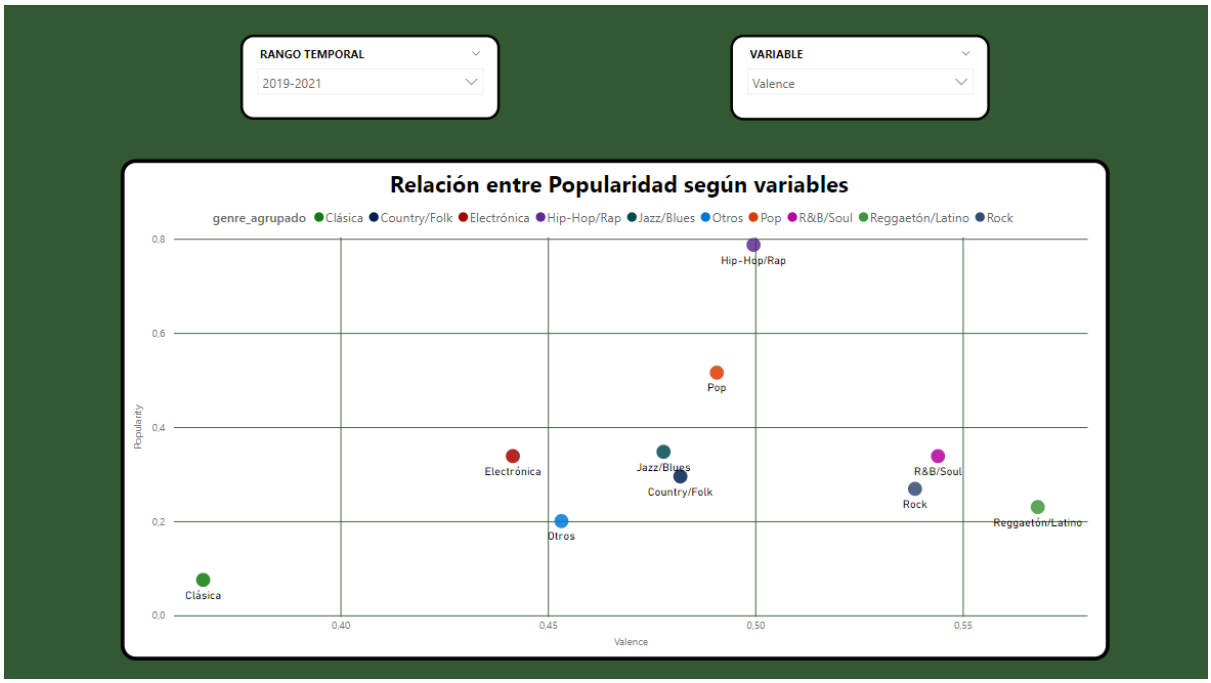

Figura 3.1: La popularidad frente al Valence de los datos agregados por generos (en la franja 2019-2021)

Para analizar cómo las características de audio (`valence`, `energy`, y `danceability`) influyen en la popularidad de las canciones decidimos crear un gráfico de dispersión dinámico en Power BI ya que se consideró que era la forma en la que mejor se apreciaba ver como varía la popularidad de la canción según cada una de estas variables.

En la figura 3.1, se observa dicho gráfico de dispersión, el cual muestra y permite analizar relación entre la popularidad promedio de distintos géneros musicales y el valence (positividad) en la franja temporal 2019-2021. Cada punto representa un género agrupado, donde el eje x indica el valence y el eje y la popularidad.

Los géneros más populares, como **Hip-Hop/Rap** y **Pop**, vemos que están más equilibrados en valence, mientras que **Reggaetón/Latino** destaca con un valence más alto y una alta popularidad también, probablemente por ser un género más festivo. Por otro lado, géneros como **Clásica** y **Otros** muestran baja popularidad y valence, lo que indica que tienen una audiencia más concreta y reducida. **Electrónica** y **Rock** tienen una posición intermedia tanto en valence como en popularidad.

- **Cálculo de las medias:**
  - Creamos una tabla de medidas calculadas donde obtuve las medias de las variables `valence`, `energy` y `danceability` para comparar las características de audio de manera homogénea utilizando la función `AVERAGE()`.

- **Creación de un parámetro dinámico:**

  - Implementamos un parámetro en Power BI que contemplase las tres medidas (`valence`, `energy`, y `danceability`) para facilitar la selección de la variable que se quiere comparar con `popularity`.

  - Este parámetro permite cambiar dinámicamente la variable que se utiliza en el eje X del gráfico de dispersión.

- **Configuración del gráfico:**

  - Seleccionamos **Gráfico de dispersión** en Power BI y asignamos las siguientes configuraciones:

    - **Eje X:** El parámetro dinámico creado previamente (que permite alternar entre `valence`, `energy`, y `danceability`).

    - **Eje Y:** `popularity`.

    - **Leyenda:** `genre_agrupado`, para diferenciar cada género por un color distinto.

  - Personalizamos levemente el diseño pensando en que se viese lo más claro posible con los colores y formatos del gráfico.

- **Filtros:**

  - Añadimos un filtro que permitiese seleccionar que variable comparar con respecto a la popularidad.

  - Añadimos un segundo filtro para segmentar los datos por `year_range`, para que la relación entre variables fuese visible en diferentes franjas de años.

Este enfoque permitió analizar las relaciones de manera dinámica, con una interfaz intuitiva que facilita el cambio de variables y rangos temporales.

3.2.2. Evaluación de la popularidad en función de la duración y los géneros musicales por época

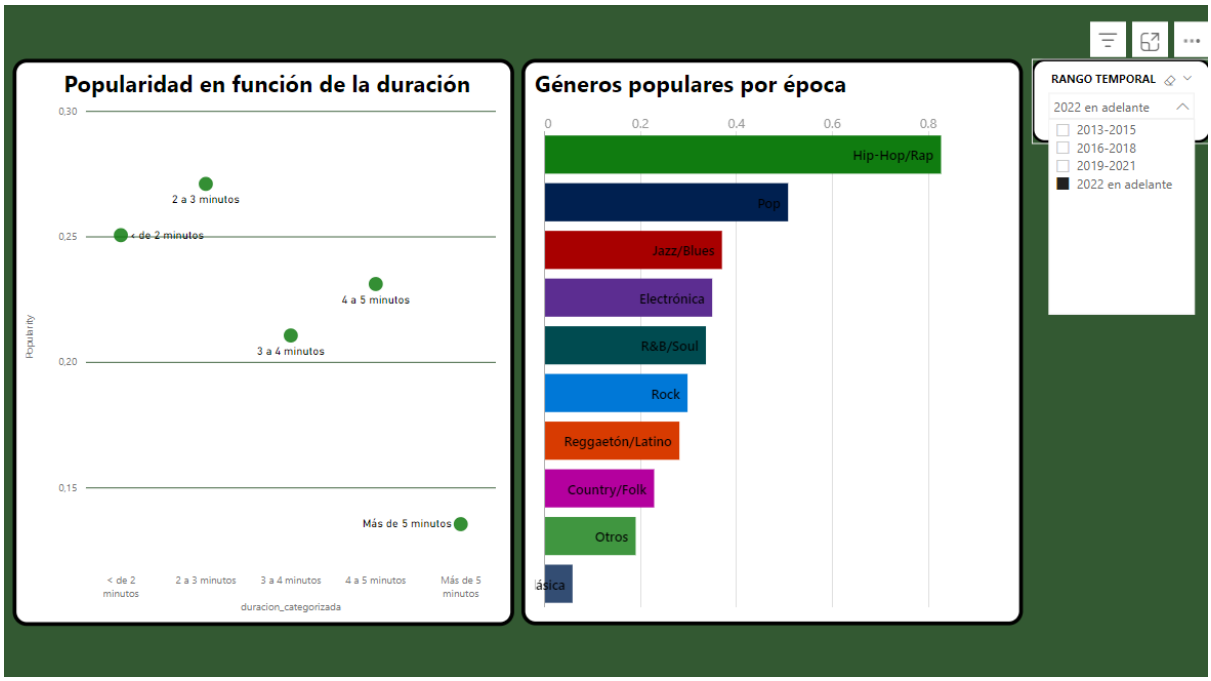

Figura 3.2: Generos populares por época y en función de su duración

En esta página del dashboard se quieren ver de manera clara dos aspectos:
El primero cómo influye la duración de las canciones en su popularidad, y el segundo qué generos han sido más populares en cada época.

En la figura 3.2, a la izquierda observamos que las canciones con una duración de **2 a 3 minutos** son las más populares, seguidas por las de **3 a 4 minutos**. Las canciones con una duración mayor a **5 minutos** tienen la menor popularidad, los usuarios prefieren claramente piezas más breves y concisas.

En el gráfico de la derecha, los géneros más populares varían según la época, pero en la franja que mostramos (que es la más actual de la que disponemos) destacan **Hip-Hop/Rap** y **Pop** como líderes en popularidad. Por otro lado, géneros como **Clásica** y **Otros** presentan menor relevancia.

3.2.2.1. Relación entre la duración de las canciones y su popularidad

Para analizar cómo influye la duración en la popularidad, creamos un gráfico de dispersión. El proceso fue el siguiente:

- Se utilizó un gráfico de dispersión, configurando:
  - El eje X con `duracion_categorizada`.
  - El eje Y con la media de `popularity_norm`.
- Configuramos también una interacción dinámica para que al seleccionar un rango de duración en el gráfico de dispersión, el de barras apiladas de la derecha

también se actualice, mostrando cómo influye esa duración en los géneros más populares.

3.2.2.2. Popularidad de los géneros musicales por época

Para observar cuáles han sido los géneros más populares en diferentes épocas, optamos por un gráfico de barras apiladas.

- Usamos `genre_agrupado` como eje categórico para identificar los géneros.
- En el eje Y, utilizamos la media de `popularity_norm`.
- Añadimos el filtro que nos muestra los datos por las franjas temporales que ya se configuró previamente.
- También hicimos la interacción dinámica entre el gráfico de dispersión y el gráfico de barras, de modo que al seleccionar un rango de duración en el gráfico de dispersión, el gráfico de barras actualizase automáticamente los géneros más populares dentro de esa duración.

La relación entre ambos gráficos permite analizar de manera conjunta cómo influyen las diferentes duraciones de canciones en los géneros más populares según la época. Se utilizaron las siguientes configuraciones adicionales: el filtro de franja temporal (`year_range`) afecta a ambos gráficos, permitiendo observar cómo cambian las tendencias a lo largo de los años.

3.2.3. Preferencias musicales según estados de ánimo

3.2.3.1. Determinar el estado de ánimo a partir de las métricas

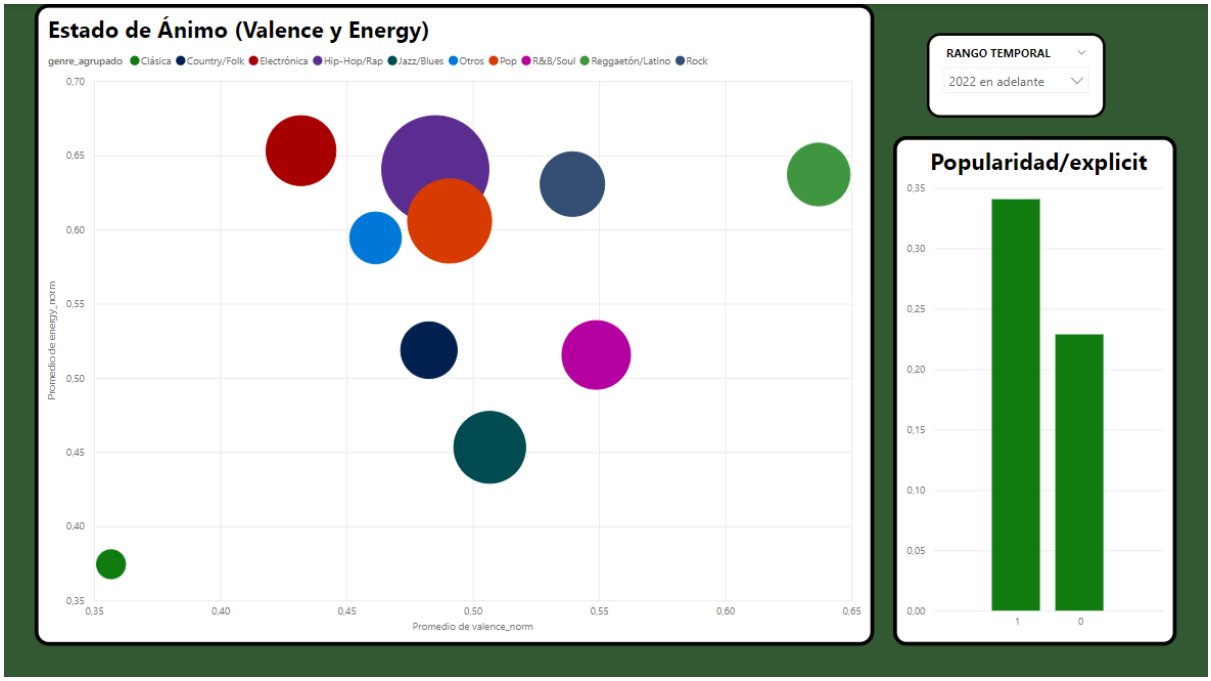

Figura 3.3: Estado de ánimo // Popularidad vs Contenido explícito

Para analizar cómo los valores de `valence` (positividad) y `energy` (nivel de energía) afectan a la popularidad de los géneros, decidimos crear un gráfico de (de dispersión). En la figura 3.3, el gráfico de la izquierda muestra la relación entre **valence** y **energy**,que como ya mencionábamos, son indicadores relacionados con el estado de ánimo . **Reggaetón/Latino** y **Electrónica** tienen mayor energía, mientras que **Clásica** y **Country/Folk** destacan por valores bajos en ambos ejes, al ser géneros más relajados.

En el gráfico de la derecha, se analiza la relación entre la **popularidad** y el atributo **explicit**. Las canciones explícitas presentan una popularidad mayor que las no explícitas.

Esta parte del análisis sirve para ver qué generos son mas escuchados según la positividad y la energía que posean. Esto nos acerca más a saber que cuando un usuario está en un estado de ánimo más feliz, eligirá normalmente géneros con una alta positividad, y al contrario si está más triste. Los resultados y conclusiones más detallados los desarrollaremos en un apartado posterior.

- Seleccionamos un gráfico de burbujas y configuramos los siguientes elementos:
  - **Eje X:** `valence`.
  - **Eje Y:** `energy`.
  - **Tamaño de la burbuja:** `popularity_norm`, para reflejar la popularidad de las canciones.
  - **Leyenda:** `genre_agrupado`, para diferenciar los géneros con colores.

3.2.4. Popularidad y explicit lyrics

En esa misma página de PowerBi, aprovechamos para poner otro gráfico de barras apiladas que mostrase en general si tenían mayor popularidad las canciones explícitas o las no explícitas. Como la columna de explicit era categórica (true/false) creamos una para poder compararla. Esta nueva se llamaríamos *explicit_numérica* y contendría un 1 para **true** y un 0 para **false**. Siendo entonces 1 explícita y 0 no explícita.

3.3. Conclusiones del análisis descriptivo

A modo de interpretación de las relaciones entre columnas del conjunto de datos que se han hecho, se relatan a continuación las conclusiones que sacamos de cada parte del análisis. La intención de obtener conclusiones de los resultados es poder aportar mejoras en cuanto a la precisión de futuros sistemas de recomendación.

3.3.1. Dashboard 1 (Figura 3.1): Relación entre la popularidad y las variables `danceability`, `valence` y `energy`

Esta primera parte del análisis tiene como objetivo evaluar la relación entre la popularidad de las canciones y tres características fundamentales (`danceability`, `valence` y `energy`) a lo largo de todas las franjas temporales que se configuraron. Si bien la hoja es interactiva para poder ver cualquier franja, comentaré la evolución entre los extremos temporales, la franja más antigua y la más reciente: 2013-2015 y 2022 en

adelante. A continuación, se presentan las conclusiones más relevantes para cada variable y período analizado.

3.3.2. Danceability y Popularidad

En esta sección analizamos cómo el atributo `danceability` influye en la popularidad de los géneros musicales, teniendo en cuenta franjas temporales concretas.

3.3.2.1. Franja 2013-2015

En este período de tiempo, observamos que los géneros con mayor `danceability` como *Reggaetón/Latino*, *Electrónica* y *R&B/Soul* tienden a ser más populares. En contraste, géneros como *Clásica* y *Country/Folk*, con valores bajos en `danceability`, muestran una popularidad significativamente menor.

El *Hip-Hop/Rap* aparece como el género que combina los más altos niveles de `danceability` con una popularidad elevada, lo que sugiere que esta variable sea lo que fomente su atractivo durante este período.

3.3.2.2. Franja 2022 en adelante

En la franja 2022 en adelante, se observa que géneros como *Pop* y *Electrónica* siguen manteniendo altos valores de `danceability` así como su popularidad. El *Reggaetón/Latino* incrementa su representación, lo cual hace que este género se consideres como muy popular y sobre todo muy bailable, posiblemente por la gran influencia que tiene la globalización de los artistas y el aumento la presencia de los mismos en plataformas de streaming.

Por otro lado, el *Hip-Hop/Rap* mantiene su popularidad y `danceability`, mientras que géneros como el *Rock* y la *Clásica* se mantienen con valores bajos en ambas variables, mostrando poca variación respecto a 2013-2015.

3.3.3. Valence y Popularidad

En esta sección analizamos cómo el atributo `valence` influye en la popularidad de los géneros musicales, teniendo en cuenta franjas temporales concretas.

3.3.3.1. Franja 2013-2015

Durante este período, es notable que los géneros con valores de `valence` más altos como el *Pop* y el *Country/Folk*, tienen una popularidad relativamente alta, sugiriendo que en general se prefieren canciones que transmiten emociones positivas. Por el contrario, la *Clásica* y el *Jazz/Blues*, con valores de `valence` más bajos, son menos populares en general.

El *Hip-Hop/Rap* tiene valor de `valence` medio-alto y una popularidad destacada, lo cual indica que es positivo y atractivo para los usuarios.

3.3.3.2. Franja 2022 en adelante

En el período más reciente que tenemos disponible en el dataset, la tendencia muestra que géneros como el *Pop* y el *Reggaetón/Latino* son los que más altos niveles de

valence y popularidad tienen, confirmando que las canciones con emociones positivas siguen siendo las favoritas.

El *Hip-Hop/Rap* mantiene un valence estable y sigue siendo altamente popular, mientras que géneros como *Rock* y *Clásica* mantienen bajos valores de valence y popularidad.

3.3.4. Energy y Popularidad

En esta sección analizamos cómo el atributo energy influye en la popularidad de los géneros musicales, teniendo en cuenta franjas temporales concretas.

3.3.4.1. Franja 2013-2015

Los géneros con alta energy, como el *Hip-Hop/Rap*, la *Electrónica* y el *Rock*, muestran una mayor popularidad durante este período, lo cual indica una clara preferencia por canciones con alta intensidad y ritmo. En contraste, géneros como *Clásica* y *Jazz/Blues*, que tienden a ser más suaves, presentan una menor popularidad.

El *Pop*, aunque tiene niveles de energy medios, tiene notoria popularidad, lo que indica que no es precisamente la energía alta lo que determina la popularidad.

3.3.4.2. Franja 2022 en adelante

En la franja más reciente que tenemos, géneros como el *Reggaetón/Latino* y la *Electrónica* tienen altos niveles de energy y una popularidad creciente, lo que nos dice que el interés va yendo hacia preferencias por canciones más intensas. El *Hip-Hop/Rap* sigue manteniendo un balance entre alta energy y popularidad, mientras que el *Rock* parece haber disminuido en relevancia.

Por otro lado, géneros con baja energy como la *Clásica* y el *Country/Folk*, mantienen una popularidad limitada, lo que sin duda confirma la idea de que las canciones más dinámicas y enérgicas son más atractivas para los oyentes ahora.

3.3.5. Resumen

El análisis de estas tres variable en ambos períodos revela patrones firmes en cuanto a la popularidad de los géneros. Géneros como el *Reggaetón/Latino*, la *Electrónica* y el *Hip-Hop/Rap* tienen un crecimiento uniforme, mientras que géneros más tradicionales como la *Clásica* y el*Country/Folk* son menos populares.

Consideramos esta observación de los resultados importante en cuanto al desarrollo de sistemas de recomendación porque permiten identificar atributos clave que influyen en las preferencias de los usuarios según el contexto de época en el que estemos y también según el género que previamente haya predominado en sus escuchas. Además, confirman la importancia de considerar variables como el valence y la energy al diseñar algoritmos de recomendación personalizados.

3.3.6. Dashboard 2 (Figura 3.2): Conclusiones sobre popularidad en función de la duración por géneros

En esta parte del análisis se pretenden interpretar dos variables que me parecen muy importante: cómo influye la duración de las canciones en su popularidad y qué géneros han sido más populares en diferentes épocas. Así podremos identificar más patrones de preferencia de los usuarios según la época.

3.3.6.1. Duración de las canciones y su relación con la popularidad

En el gráfico de la izquierda, se compara la duración de las canciones, categorizada en rangos de minutos, con su popularidad media.

- **Canciones cortas:** las canciones con una duración inferior a 2 minutos tienden a ser menos populares en comparación con otros rangos, lo que nos indica que no son tan atractivas como canciones que duran más, o que el usuario las interpreta como demasiado incompletas.
- **Canciones de 2 a 3 minutos:** este rango es el que muestra el mayor nivel de popularidad promedio, lo cual indica que los oyentes prefieren canciones que son lo suficientemente breves como para mantener su atención, pero no tan cortas como para quedarse faltos de información.
- **Canciones largas:** las canciones con más de 5 minutos de duración presentan una menor popularidad promedio. Esto nos puede decir que los oyentes actuales prefieren experiencias musicales más concisas, probablemente influenciadas por el consumo cada vez más rápido en plataformas digitales. Como hipótesis personal, si hiciesemos este análisis en 2040, cambiaría drásticamente este gráfico, tendrían seguramente más interés canciones de duración aproximada de 1 minuto y bajaría considerablemente el interés por canciones *tan* largas como 4, 5, 6 minutos. Esto viene por la influencia de la cultura del placer instantáneo, que promueve tener lo que queremos siempre disponible y por la cual nuestra capacidad de interés se ve transformada en querer estar probando nuevas experiencias constantemente, sin importarnos si son demasiado cortas.

Este análisis sugiere que para capturar el interés de la mayoría del público, hoy en día las canciones deben estar dentro de un rango de 2 a 4 minutos, alineándose con las tendencias de consumo actuales.

3.3.6.2. Popularidad de los géneros musicales según la época

El gráfico de barras de la derecha muestra los géneros musicales más populares en cada franja de años, de nuevo, comentaremos solo los dos periodos de tiempo: 2013-2015 y 2022 en adelante.

- **2013-2015:** durante este periodo, el género **Hip-Hop/Rap** lidera ampliamente en popularidad, seguido por **Pop** y **Jazz/Blues**. Esto es un reflejo claro de la influencia de la cultura en esa época.
- **2022 en adelante:** aunque el **Hip-Hop/Rap** sigue siendo dominante, el **Pop** muestra una popularidad significativamente más alta en comparación con el periodo de 2013, seguramente se deba al impacto que han conseguido tener algunos artistas en los últimos años. Los géneros como **Reggaetón/Latino** y

Electrónica también destacan, reflejando el crecimiento de estos géneros en el panorama internacional.

- **Menor popularidad:** géneros como **Clásica** y **Country/Folk** permanecen entre los menos populares en ambos periodos, lo que indica que existe una audiencia de nicho para estos estilos.

3.3.6.3. Interpretación general

Este análisis consigue ayudar a entender cómo han cambiado las preferencias musicales con el tiempo. El **Hip-Hop/Rap**, por ejemplo, ha mantenido su popularidad, pero otros como el **Reggaetón/Latino** han crecido considerablemente, lo que indica una evolución. También consideramos que la relación entre la duración de las canciones y el impacto que luego tienen en los oyentes, es algo que los productores y compositores tienen cada vez más en cuenta al crear música.

Estos resultados podrían servir de ayuda a la hora de diseñar sistemas de recomendación personalizados que consideren tanto las preferencias de género como la duración preferida por los usuarios.

3.3.7. Dashboard 3 (Figura 3.3) Conclusiones del Estado de Ánimo y Explicit Lyrics

(2013-2015 vs. 2022 en adelante) En este dashboard [**fig-3.3**], la intención fue analizar dos aspectos clave relativos al estado de ánimo del usuario: (`valence` y `energy`) según los distintos géneros musicales, así como la relación entre canciones explícitas o no explícitas y su popularidad. A continuación, se presentan las conclusiones segmentadas por rangos temporales:

Análisis del estado de ánimo y canciones explícitas

1. **Análisis del estado de ánimo: 2013-2015**
   Durante este periodo, los géneros musicales con mayor energía y positividad fueron **Hip-Hop/Rap** y **Reggaetón/Latino**, los cuales destacan como los más energéticos, con valores altos tanto de `valence` como de `energy`. Esto refleja una fuerte preferencia por canciones intensas en cuanto al ritmo y la fuerza, así como que acompañen en emociones positivas.

   Por otro lado, géneros como **Jazz/Blues** y **Country/Folk** presentaron valores más bajos en energía y positividad, posicionándose como opciones más relajadas y no necesariamente "felices". Finalmente, el género **Clásica** sobresale con los valores más bajos tanto en energía como en positividad, indicando que durante este periodo las canciones clásicas tuvieron un carácter más melancólico o al menos fueron utilizadas para acompañar sentimientos más nostálgicos o emociones no tan positivas.

2. **Análisis del estado de ánimo: 2022 en adelante**
   En este rango de años más reciente, los resultados indican que el **Hip-Hop/Rap** mantiene su posición como un género con valores más altos de energía y positividad, lo cual sugiere su éxito en contextos más festivos o emociones relacionadas con la celebración. No obstante, **Reggaetón/Latino** sobresale aún más,

consolidándose como un género dominante en los últimos años gracias a su alta energía y positividad.

El **Pop** también muestra gran positividad, aunque con niveles de energía moderados, lo que indica su adaptabilidad a diferentes intensidades. Mientras tanto, la **Electrónica** y el **R&B/Soul** se posicionan como géneros intermedios, con un equilibrio entre energía y positividad. Finalmente, géneros como la **Clásica** y el **Country/Folk** mantienen valores bajos en ambas variables, considerandose como opciones más tranquilas y emocionales.

3. **Análisis de las canciones explícitas**
   En ambos periodos, las canciones explícitas (`explicit` = 1) tienen una popularidad significativamente mayor que las no explícitas (`explicit` = 0). Este resultado es consistente a lo largo de los años, lo que sugiere que las canciones con contenido explícito suelen tener un mayor impacto en general entre los oyentes. Esto puede deberse al auge de géneros como el **Hip-Hop/Rap** y el **Reggaetón/-Latino**, que suelen incluir contenido explícito.

4. **Conclusión general**
   Este dashboard de PowerBi refleja el comportamiento de la sociedad, que mueve las escuchas en plataformas como Spotify, actualmente tan extendidas. Es evidente que nuestras escuchas dependen del estado de ánimo que experimentamos en cada momento. Este aspecto es crucial para estudiar patrones de comportamiento en los usuarios y orientar las recomendaciones según su estado de ánimo.

Capítulo 4

Implementación de un prototipo de Sistema de Recomendación

1. **Clustering del dataset:** elegimos tres algoritmos de clustering (CLARA, DBSCAN y HCLUST) para hacer agrupaciones según las características musicales que pudiesen agruparse, para así ir identificando patrones y facilitar la segmentación del dataset.
2. **Análisis de resultados:** generamos estadísticas y visualizaciones para interpretar los resultados de los distintos algoritmos de clustering y sus propiedades.

4.1. Clustering del Dataset

Para optimizar el rendimiento, optamos por trabajar con una muestra reducida de 5000 filas seleccionadas aleatoriamente del dataset original, ya que si trabajaba con todos los datos era demasiado lento el algoritmo y no era eficiente.
De esta manera pudimos conseguir un equilibrio de representatividad que no me hiciese perder eficiencia a nivel computacional.

4.1.1. Elección de algoritmos

- **CLARA (Clustering LARge Applications) :** elegimos este algoritmo como método particional, esta basado en medoids (medoides) que son puntos reales del dataset que representan el clúster. La diferencia con respecto a los centroides es que estos últimos son puntos calculados promediando características. Seleccioné 10 clústeres para organizar el dataset en agrupaciones significativas. Además, los medoides fueron posteriormente utilizados como entrada para el clustering jerárquico.
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** el algoritmo de densidad me parmitiría identificar no solo clústeres definidos por densidad, sino también outliers, que recogen rarezas o canciones que no pueden ser agrupadas.
- **Clustering jerárquico:** basado en los medoides obtenidos de CLARA, este método permitiría generar estructura jerárquica que será útil para analizar las relaciones entre clústeres.

4.1.2. Implementación

La implementación se llevó a cabo en Python utilizando `pandas`, `scikit-learn`, `matplotlib` y `scikit-learn-extra`. Los pasos principales fueron:

1. **Carga y preprocesamiento del dataset:** seleccionamos las características o variables musicales más relevantes(numéricas todas ellas):
   - `popularity_norm`, `danceability_norm`, `energy_norm`, `valence_norm`, `acousticness_n`
     `tempo_norm`, `liveness_norm`, `speechiness_norm`.
2. **Clustering:** aplicamos los algoritmos seleccionados en el orden CLARA → DBSCAN → Jerárquico.
3. **Visualización y análisis:** generamos gráficas y promedios por clúster para interpretar los resultados.

4.1.3. Resultados y Conclusiones del Clustering

En esta sección, discutiremos los resultados obtenidos del clustering, veremos cómo han sido las agrupaciones resultado de cada una de las técnicas y comentaremos qué sacamos en claro de cada una de ellas.

4.1.3.1. Resultados del Clustering CLARA

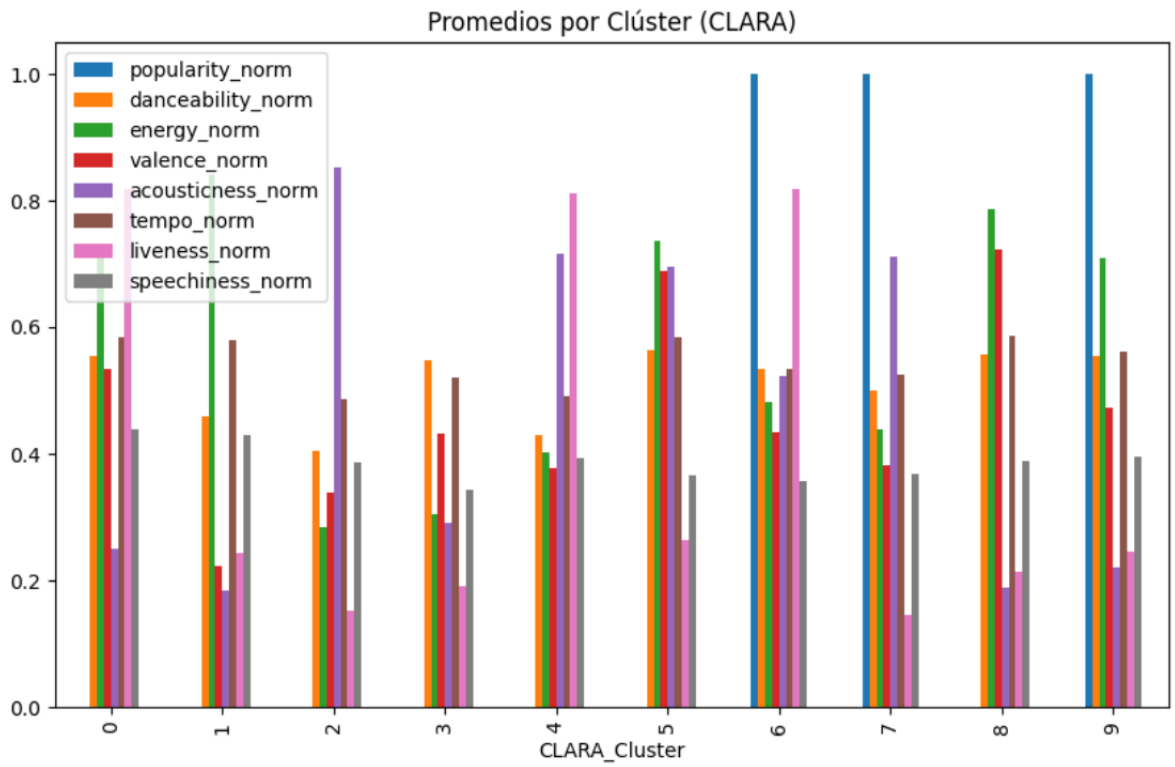

Figura 4.1: Clustering Clara

El algoritmo CLARA generó 10 clústeres(véase Figura [**fig-3.4**]), cada uno representado por un medoide. Los promedios de las características en cada clúster nos dicen que:

- Los clústeres con alta `popularity_norm`, como el **0** y el **7**, contienen canciones populares y que son más escuchadas en general por el usuario medio.
- Valores altos en `danceability_norm` y `tempo_norm` en el clúster **4** sugieren canciones rápidas y bailables.
- Clústeres con valores altos de `acousticness_norm` y bajos en `energy_norm`, como el **3**, representan canciones más acústicas y tranquilas.

4.1.3.2. Función para obtener medoides del algoritmo CLARA

A continuación se describe la función añadida al código que teníamos relativo al Clustering para obtener los medoides que genera el algoritmo CLARA.

Listing 4.1: Función que muestra los medoides

```
def obtener_medoides(self):
    print("Obteniendo medoides de cada cluster...")
    clara_model = KMedoids(n_clusters=10, method='pam', random_state=42)
    clara_model.fit(self.dataset[self.numeric_features])

    medoides_indices = clara_model.medoid_indices_
    medoides_info = []
    for idx in medoides_indices:
        medoide = self.dataset.iloc[idx]
        medoides_info.append({
            "Cluster": clara_model.labels_[idx],
            "Track Name": medoide.get("track_name", "Desconocido"),
            "Popularity": medoide["popularity_norm"],
            "Danceability": medoide["danceability_norm"],
            "Energy": medoide["energy_norm"],
            "Valence": medoide["valence_norm"],
            "Acousticness": medoide["acousticness_norm"],
            "Tempo": medoide["tempo_norm"],
            "Liveness": medoide["liveness_norm"],
            "Speechiness": medoide["speechiness_norm"],
        })

    medoides_file_path = os.path.join(self.output_dir, "clara_medoides.csv")
    medoides_df = pd.DataFrame(medoides_info)
    medoides_df.to_csv(medoides_file_path, index=False)
    print(f"Archivo de medoides guardado en: {medoides_file_path}")
    return medoides_info
```

En la función 4.1 se identifican los medoides generados por el algoritmo CLARA. Cada medoide representa el punto más cercano al centro del clúster correspondiente, y sus características promedio se extraen y almacenan si está disponible la pista. Los resultados se guardan en un archivo en formato .csv.

4.1.3.3. Resultados de los medoides

La siguiente tabla muestra las canciones identificadas para cada uno de los medoides:

Cluster	Track Name	Popularity	Danceability	Energy	Valence	Acousticness	Tempo	Liveness	Speechiness
0	Mi Buenos Aires Querido	0.00	0.60	0.74	0.57	0.21	0.59	0.86	0.45
1	Chainsaw man main theme	0.00	0.58	0.89	0.22	0.13	0.59	0.24	0.43
2	Waipi'o Paea	0.00	0.47	0.26	0.39	0.55	0.15	0.37	0.29
3	Los que marcan el camino	0.00	0.62	0.34	0.43	0.35	0.10	0.29	0.29
4	Se Deus Disser	0.00	0.39	0.41	0.35	0.75	0.46	0.82	0.29
5	Struna	0.00	0.57	0.79	0.71	0.69	0.60	0.21	0.35
6	Swept Away in Wonder	1.00	0.48	0.36	0.39	0.52	0.05	0.31	0.31
7	Something New	1.00	0.48	0.36	0.71	0.52	0.05	0.31	0.31
8	indieedycool	0.00	0.60	0.83	0.72	0.16	0.59	0.18	0.37
9	Heart	1.00	0.54	0.72	0.47	0.19	0.61	0.26	0.36

Cuadro 4.1: Medoides generados por el algoritmo CLARA.

Estas canciones destacan por ser las más representativas de los grupos formados.

4.1.3.4. Descripción de los clústeres destacados

A continuación, se destacan algunos de los clústeres más relevantes:

- **Cluster 1:** representado por la canción *Chainsaw man main theme but it's lofi hiphop*, este clúster muestra una alta `danceability` (0.89) y valor (0.0) `popularity` . Agrupa canciones muy bailables, típicas de, por ejemplo, el hip-hop.

- **Cluster 4:** incluye la canción *Se Deus Disser*, con valor alto de `energy` (0.75) y moderado de `valence` (0.46). Este clúster caracteriza canciones intensas y enérgicas.

- **Cluster 5:** con *Struna* como representante, este grupo tiene una popularidad moderada (0.57) y valores equilibrados de `danceability` (0.60) y `energy` (0.69), son por tanto canciones accesibles y agradables para la audiencia general.

- **Cluster 9:** representado por *Heart*, con alta `popularity` (1.0) y `energy` (0.72). Este clúster agrupa canciones dinámicas y populares.

Estos clústeres nos muestran la diversidad de las canciones en función de sus características, lo que facilitará posteriormente ser más precisos con las recomendaciones personalizadas.

4.1.4. Resultados del Clustering DBSCAN

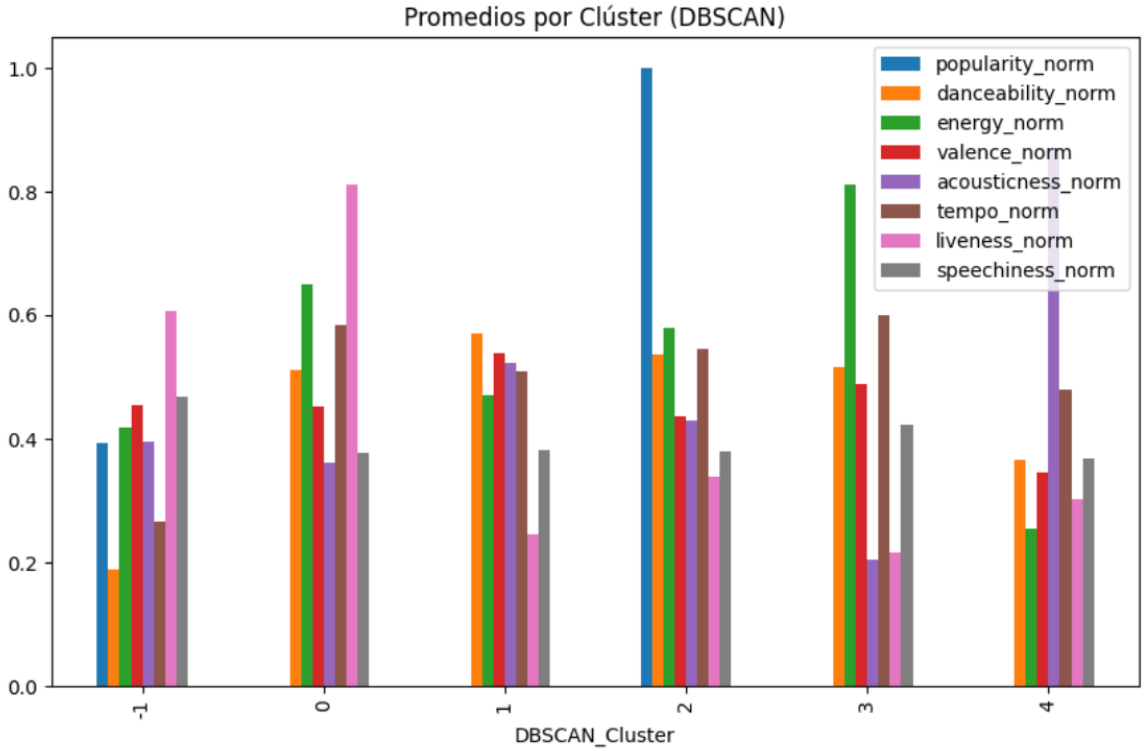

Figura 4.2: Clustering DBSCAN

Como se puede apreciar en la figura 4.2, DBSCAN identificó cinco clústeres principales y un conjunto de outliers (`-1`). De estos resultados podemos destacar que:

- Los **outliers (clúster -1)** reflejan canciones con características inusuales, útiles para recomendar contenido específico.

- El clúster **2** incluye canciones con alta `energy_norm` y `tempo_norm`, lo cual indica canciones más energicas y rápidas en ritmo.

4.1.5. Resultados del Clustering Jerárquico

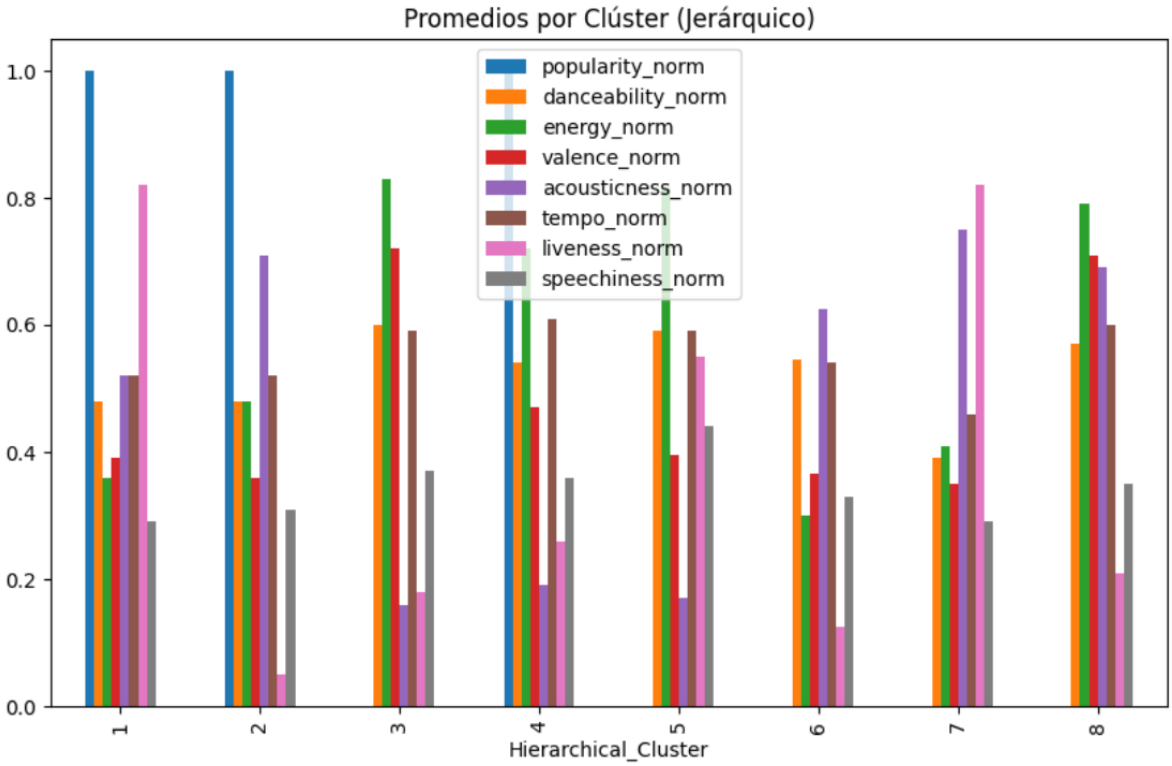

Figura 4.3: Clustering Jerárquico

El número de clústeres generados por el algoritmo jerárquico no siempre coincide con el parámetro `n_clusters`. Esto puede deberse a:

- **Redundancia en los datos**: si los puntos de datos están demasiado cercanos entre sí, el modelo opta por agruparlos juntos, reduciendo el número de clústeres únicos.
- **Umbral en el dendrograma**: el dendrograma agrupa los puntos de datos en función de un umbral de similitud, y esto puede impedir que se formen todos los grupos deseados según la estructura del árbol.

[Die21]

Basándonos en los medoids de CLARA, el clustering jerárquico (vease la figura 4.3) permitió observar:

- Relaciones entre clústeres: clústeres cercanos, como el **4** y el **5**, comparten características similares de (`popularity_norm` y de `energy_norm`).
- Subdivisiones claras en clústeres con diferentes perfiles musicales, útiles para las posteriores recomendaciones.

El dendograma generado muestra cómo los clústeres se fusionan progresivamente.

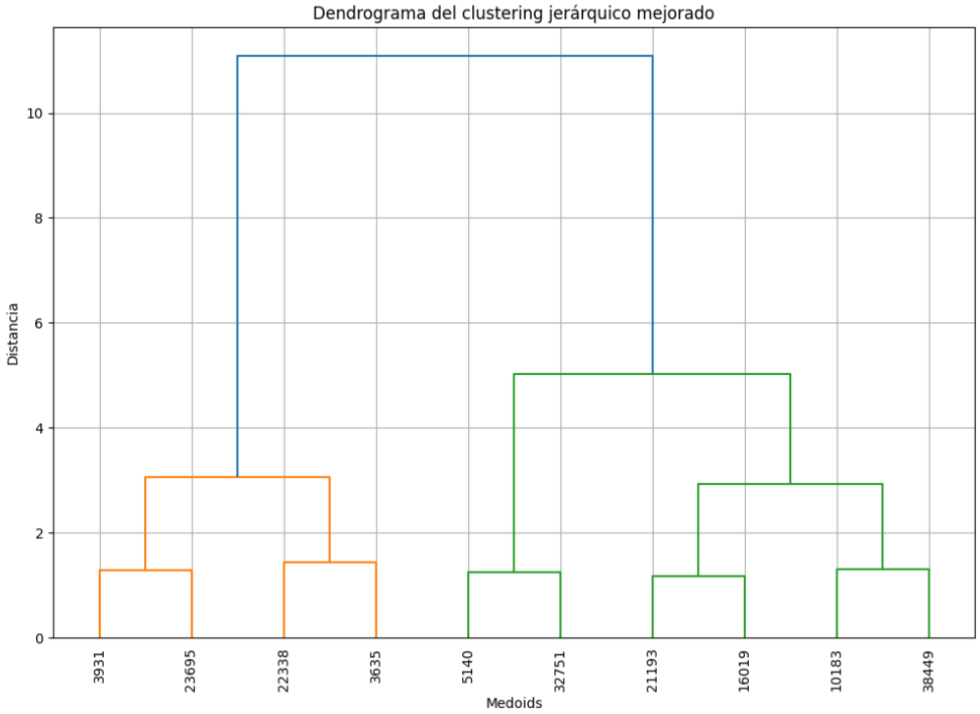

Figura 4.4: Dendograma

Medoide	Popularity	Danceability	Energy	Valence	Acousticness	Tempo	Liveness	Speechiness
3931	0.0	0.50	0.66	0.69	0.08	0.58	0.47	0.14
23695	0.0	0.78	0.80	0.49	0.28	0.61	0.76	0.34
22338	0.0	0.55	0.62	0.07	0.05	0.75	0.22	0.33
3635	0.0	0.06	0.42	0.53	0.70	0.53	0.21	0.34
5140	1.0	0.73	0.45	0.11	0.59	0.06	0.01	0.53
32751	1.0	0.29	0.64	0.53	0.18	0.91	0.10	0.55
21193	1.0	0.43	0.11	0.11	0.84	0.30	0.97	0.51
16019	0.0	0.65	0.32	0.38	0.81	0.76	0.81	0.72
10183	0.0	0.43	0.56	0.45	0.15	0.10	0.53	0.04
38449	0.0	0.07	0.08	0.63	0.11	0.52	0.47	0.03

Cuadro 4.2: Medoids identificados en el dendograma del clustering jerárquico, incluyendo las características principales de cada uno.

4.1.6. Clases Implementadas

- **Clase `Clustering.py`:** es en la que se aplican los algoritmos CLARA, DBSCAN y jerárquico. Además va guardando los resultados en archivos CSV.
- **Clase `estadisticas.py`:** calcula los promedios de las variables de cada algoritmo de clustering y genera las gráficas.

4.2. Implementación del prototipo de Sistema de Recomendación

Desarrollamos un prototipo de sistema de recomendación musical para hacer visible la efectividad que tienen los algoritmos de clustering y los muchos algoritmos de

similitud que se utilizan en los sistemas que ya conocemos, que permiten ofrecer recomendaciones personalizadas basadas en datos. En nuestro caso, los datos (el historial de usuarios y los propios usuarios) serían artificiales, pero por eso es un prototipo y por eso es una simulación. A continuación, se detallan las decisiones tomadas, la lógica implementada, y el funcionamiento del sistema.

4.2.1. Decisiones de Diseño y Lógica Implementada

El sistema está diseñado para trabajar con un conjunto de canciones de nuestro dataset concreto previamente limpiado y preparado. Las canciones, además, han sido clasificadas segun los algoritmos de clustering que fueron explicados en la anterior sección. A partir de estas agrupaciones, asignamos perfiles musicales a un grupo de 15 usuarios ficticios.

Cada usuario tiene un perfil musical que define sus preferencias a nivel de características como *danceability_norm*, *energy_norm*, *valence_norm*, y otras características relevantes para determinar un perfil dentro del dataset.

4.2.2. Asignación de Perfiles y Matriz de Usuarios

Los perfiles se asignaron a partir de características específicas del dataset de la siguiente manera:

Listing 4.2: Filtrado de canciones según el perfil del usuario

```
if profile == "Fan de Pop":
    filtered_songs = self.dataset[
        (self.dataset["danceability_norm"] > 0.7) &
        (self.dataset["valence_norm"] > 0.7)
    ]
elif profile == "Rockero":
    filtered_songs = self.dataset[
        (self.dataset["energy_norm"] > 0.7) &
        (self.dataset["valence_norm"] > 0.5)
    ]
    # ...etc
```

Como estábamos trabajando con una muestra de 5000 datos, decidí que 5 perfiles era un numero adecuado para que hubiese heterogeneidad en el conjunto de datos, siendo uno de los perfiles: generalista (que es aleatorio).

- **Fan de Pop:** canciones con altos valores en *danceability_norm* y *valence_norm*.
- **Rockero:** canciones con altos valores en *energy_norm* y valores moderados en *valence_norm*.
- **Amante de Clásica:** canciones con altos valores en *acousticness_norm* y *liveness_norm*.
- **Fan de Electrónica:** canciones rápidas y enérgicas con altos valores en *tempo_norm* y *energy_norm*.
- **Generalista:** canciones seleccionadas uniformemente de todos los clústeres.

Despues de tener los perfiles marcados, generamos una matriz de usuarios y canciones. Cada fila de la matriz representa un usuario, y cada columna indica si el usuario ha escuchado una canción (`1`) o no (`0`). La matriz se asigna 20 canciones a

cada usuario, seleccionadas del subconjunto de canciones que mejor se ajustan a su perfil.

4.2.3. Similitud Coseno para Recomendaciones

Finalmente para las recomendaciones, utilizamos la métrica de similitud coseno, porque está en muchas páginas y foros y quedó claro que era la que utilizaban diversos sistemas de recomendación de todo tipo(ventas, música, etc). Mide el grado de similitud entre dos usuarios considerando el historial de canciones que ha escuchado cada uno. La fórmula es:

$$\text{Similitud Coseno} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \cdot \|\mathbf{B}\|}$$

Donde:

- $\mathbf{A}$ y $\mathbf{B}$ son los vectores binarios de canciones escuchadas de dos usuarios.
- $\|\mathbf{A}\|$ y $\|\mathbf{B}\|$ son las magnitudes de los vectores.
- $\mathbf{A} \cdot \mathbf{B}$ es el producto escalar entre los vectores.

Lo interesante de la Similitud Coseno es que da igual la magnitud del vector, lo que nos importa es como de cercana es la dirección entre los vectores

El sistema calcula esta similitud entre el un usuario y el resto de usuarios, después ordena los usuarios más similares y usa el historial de canciones de estos usuarios como base para generar recomendaciones. Las canciones ya escuchadas por el usuario seleccionado son excluidas para que no se repitan (este último filtro también es fundamental).

4.2.4. Interfaz Gráfica de Usuario

A continuación, se muestra una captura de pantalla de la interfaz implementada, donde se puede observar cómo un usuario selecciona su perfil y recibe una lista de canciones recomendadas.

Figura 4.5: Ejemplo de lista de recomendaciones para uno de los usuarios

Se diseñó una interfaz gráfica muy simple utilizando `Tkinter`[Fou25], la biblioteca estándar de Python para crear interfaces gráficas de usuario (GUI) que tiene diversas herramientas para diseñar elementos interactivos como botones, menús, etc. de forma simple y es sencilla de utilizar a la vez que eficiente.
Está integrada con las funciones del código:

- **Selector de Usuario:** es un menú desplegable con la lista de usuarios ficticios que se han generado previamente en la clase `SistemaRecomendacion`. La lista está almacenada en el atributo `self.user_names` y se pasa a la interfaz en la inicialización:

Listing 4.3: Inicialización de un desplegable en Tkinter para seleccionar un usuario.

```
self.usuario_var = tk.StringVar(master)
self.usuario_var.set(self.sistema.user_names[0])
self.menu_usuarios = ttk.Combobox(
    master, textvariable=self.usuario_var,
    values=self.sistema.user_names, state="readonly"
)
```

  El usuario que seleccionemos en el menú será el que se utilizará para determinar qué índice de usuario le corresponde en la matriz de usuarios `self.user_song_matrix`, así de esta manera cada recomendación es exclusiva para cada usuario.

- **Botón de Recomendaciones:** está vinculado a la función `mostrar_recomendaciones` mediante el argumento `command`. Al hacer click en él, la función, a la función le llega el usuario que hemos seleccionado del desplegable y busca su índice en la

lista de usuarios. Finalmente llama al método `recomendar_canciones`.

```
usuario_seleccionado = self.usuario_var.get()
user_index = self.sistema.user_names.index(usuario_seleccionado)
recomendaciones = self.sistema.recomendar_canciones(user_index)
```

El cual devuelve la lista de canciones recomendadas en un recuadro de texto a parte.

- **Conexión con la Clase `SistemaRecomendacion`:** La interfaz recibe una instancia del sistema de recomendación al inicializarse:

```
gui = SistemaRecomendacionGUI(root, sistema)
```

La idea es que todos los pasos sean accesibles desde la interfaz.

- **Datos Dinámicos en Tiempo de Ejecución:** la interfaz también depende de los datos cargados dinámicamente desde el archivo de clustering `clara_clustering.csv` por lo que cualquier actualización de este contenido de datos afectará en las recomendaciones mostradas en la interfaz.

4.2.5. Resultados y Ejemplo de Ejecución

Se probó el sistema usando un conjunto de 15 usuarios y 5000 canciones y habiendole pasado los algoritmos de clustering. A continuación, un ejemplo de ejecución tanto del clustering como de la recomendación en si.

4.2.5.1. Comandos para el Clustering

El comando utilizado para ejecutar el clustering es:

```
python clustering.py
```

Este proceso genera una serie de archivos de salida dentro de la carpeta `outputs/clustering_results` que incluye:

- `clara_clustering.csv` - Canciones agrupadas con el algoritmo CLARA.
- `dbscan_clustering.csv` - Resultados del clustering DBSCAN.
- `hierarchical_clustering.csv` - Agrupación jerárquica basada en los medoides generados por CLARA.

4.2.5.2. Ejecución del Sistema de Recomendación

El comando es:

```
python SistemaRecomendacionGUI.py
```

El programa carga los datos (que ya han sido tratados por los algoritmos de clustering) y genera la matriz. Después, aparece la interfaz para permitirnos interactuar y cambiar de usuarios.

4.2.5.3. Ejemplo de Ejecución

Un ejemplo de la ejecución y salida del sistema es:

```
Generada matriz de usuarios ficticios con 15 usuarios y 5000 canciones.
Usuario seleccionado: Juan (Fan de Pop)
Recomendaciones generadas:
1. "Heartbeats" - The Knife
2. "Sweet Jane - Full Length Version; 2015 Remaster" - The Velvet Underground
3. "KEEP IT TO YOURSELF" - Sarah Vaughan
...
```

Aqui se ve que el sistema ha generado recomendaciones para el usuario `Juan`,que tiene perfil de `Fan de Pop`. Las recomendaciones tienen en cuenta las variables del perfil asignado (que para este perfil de ejemplo son altos valores en `danceability_norm` y `valence_norm`) y la similitud con otros usuarios que ha sido calculada con la *similitud coseno*. Por último se muestran en la interfaz confirmando que el backend coincide con el frontend [RRS11l].

4.2.6. Interpretación del Resultado

Hay que tener en cuenta que, como el prototipo está combinando varias técnicas de *clustering*, asignación de perfiles basados en variables musicales y cálculo de similitudes entre usuarios, las recomendaciones que resultan pueden no parecer a priori tan precisas, pero sí que lo son en cuanto a que se centra en que sean similares las características del historial con la del historial de un usuario similar, intentando así que el usuario recomendado pueda disfutar del mayor número de canciones posibles de esa lista de recomendaciones.

Otra opción que fue valorada por un instante era la de filtrar y recomendar por género únicamente, pero esto limitaba mucho y no tenía el sentido que se supone que debe tener una recomendación que trata de primar la diversidad en las escuchas.

Capítulo 5

Resultados y conclusiones

5.1. Resultados

Los resultados obtenidos durante este trabajo han sido expuestos y comentados a lo largo del desarrollo del proyecto en cada sección correspondiente.

En líneas generales y como valoración personal, los resultados han sido coherentes con la experiencia que tiene un usuario de plataformas de streaming como yo (en nuestro caso Spotify). Las características musicales analizadas, como la *danceability*, la *valence*, o los géneros predominantes, han mostrado resultados que coinciden con lo que esperaba.

El sistema de recomendación prototipo desarrollado también ha ofrecido resultados satisfactorios al sugerir canciones alineadas con los perfiles de los usuarios ficticios definidos. Este proceso demuestra la utilidad del clustering y la personalización basada en características específicas.

Este proyecto no solo ha sido una oportunidad para explorar la tecnología detrás de los sistemas de recomendación, sino también para reflexionar sobre sus implicaciones. La necesidad de equilibrio entre precisión tecnológica y diversidad cultural sigue siendo un desafío que merece atención en futuros desarrollos.

En conclusión, los objetivos planteados al inicio de este trabajo han sido alcanzados, y los resultados obtenidos no solo validan las decisiones metodológicas tomadas, sino que también sirven como base para futuras investigaciones y mejoras en sistemas de recomendación.

5.2. Trabajo futuro

Uno de los aspectos que podría haberse completado pero que es un feedback que debe venir de parte del grado de satisfacción de los usuarios, es la evaluación de la precisión del sistema. Como trabajo futuro se podrían analizar las métricas del sistema, como por ejemplo la exactitud de las recomendaciones, el promedio de precisión de todos los usuarios, el ratio de conversión satisfactoria y un largo etcétera.

Pero como decía anteriormente, no se considera tarea de quien realiza el algoritmo sino de quien lo utiliza, lo implementa en su plataforma y verifica si los clientes están

satisfechos o las ventas del sitio aumentan, eso sí dará una clara respuesta de si es realmente efectivo.

El enfoque futuro del sistema debería centrarse en integrar el sistema en una plataforma real y medir su impacto en la satisfacción de los usuarios, así como realizar estudios de usuario que evalúen cómo perciben las diversas recomendaciones y si aumenta su interés en dicha plataforma.

5.3. Objetivos Cubiertos con Respecto a los planteados

La mayoría de los objetivos que me propuse completar, pero es posible que algunos aspectos podrían ser profundizados con un análisis adicional en un futuro.

5.3.1. Segmentación de usuarios

Conseguimos identificar bien los conjuntos de usuarios con comportamientos similares gracias a los algoritmos de clustering, esto ayudó sin duda a tener más precisión a la hora de recomendar, es la base de todo. Se clasificaron las variables clave como género, características de audio y popularidad.

5.3.2. Correlación y asociación entre variables

Se hizo hincapié en el análisis sobre cómo diferentes variables influyen desde hace años en la música y las preferencias de los usuarios.

- **Análisis de Popularidad por Año y Género:** examinamos cómo la popularidad de los géneros ha cambiado a lo largo de los años.
- **Análisis de Características de Audio:** comprobamos la relación entre las características de audio (*danceability, energy, valence*, etc.) y la popularidad de las canciones.
- **Análisis por Franjas de Años:** agrupamos los datos en franjas temporales, lo que me permitió ver cambios en las preferencias según la época.
- **Análisis de Canciones Explícitas vs. No Explícitas:** confirmamos que la popularidad de las canciones con contenido explícito ha ido en aumento en los últimos años.

5.3.3. Mejoras en sistemas de recomendación

La propuesta ha sido un sistema de recomendación híbrido que combine características variables de la música con similitudes entre usuarios con el objetivo de evitar patrones repetitivos de recomendaciones y ganar en diversidad a la hora de recomendar.

5.3.4. Desarrollo de una interfaz

Se diseñó y se implementó una interfaz gráfica funcional que permitiese explorar las recomendaciones de forma dinámica.

5.3.5. Recomendaciones basadas en estado de ánimo y actividad

Aunque sería mejorable este aspecto, uno de los objetivos era poder ser capaces de que el sistema respondiese al usuario teniendo en cuenta su histórico de estados de ánimo(medido por *valence* y *energy*), generando playlists adaptadas a dichos parámetros.

5.3.6. Objetivos Pendientes

5.3.6.1. Evaluación de precisión del sistema

Aunque el sistema implementado cumple su propósito, no llegué a hacer una evaluación exhaustiva de su precisión mediante métricas como *Precision@K* o *Recall*.

En resumen, se ha logrado cubrir prácticamente todos los objetivos planteados al inicio del proyecto, cumpliendo con requisitos establecidos y siempre tratando de aportar avances así como teniendo la sensación de haber dejado una base sólida sobre la que, por supuesto, pueden mejorarse infinidad de aspectos.

5.4. Conclusiones personales

Este sistema, además de tratar de conseguir una mejora en la calidad de las recomendaciones, también quiere promover una experiencia más enriquecedora para los usuarios y contribuir a la diversidad cultural. Sin embargo, no podría permitirme ignorar los dilemas éticos asociados a este tipo de sistemas.

Uno de los principales desafíos que vivimos en relación con este tipo de sistemas está directamente relacionado con las leyes de protección de datos. Los sistemas de recomendación suelen basarse en un análisis exhaustivo de nuestros comportamientos, gustos, patrones de consumo, etcétera. Pero... ¿hasta qué punto sabemos qué datos nuestros están siendo utilizados y para qué? ¿Es justo que se construyan perfiles tan detallados de nuestras vidas para fines comerciales?

Además, los sistemas de recomendación, como el que he tratado de implementar para perfeccionar algunos aspectos, tienen un gran potencial para incrementar aún más los sesgos existentes de los cuales inevitablemente somos víctimas. Las recomendaciones, por ejemplo, pueden querer reforzar tendencias repetitivas, promoviendo contenidos populares o intereses de personas concretas que quieran manipular los patrones de las escuchas, lo cual dificultaría la diversidad real en lo que consumimos. Esto puede fomentar un comportamiento de rebaño, donde se nos empuja constantemente hacia lo que otros están disfrutando o hacia lo que quieren que forzosamente disfrutemos entre muchas comillas.

Este conflicto interno ha estado presente durante el desarrollo del proyecto. Mientras mi intención era crear un sistema eficaz, no podía no cuestionarme si es saludable que nuestra interacción con el mundo esté tan influida por algoritmos que condicionan nuestro comportamiento. Esta tecnología puede ser una herramienta para el descubrimiento cultural, o una potente herramienta de censura que recuerda al siglo pasado. Incluso es más grave cuando se aplican técnicas parecidas a otras áreas más delicadas como la información o la política.

Una posible solución a este problema ya se está viendo implementada en algunas

plataformas, como por ejemplo BlueSky, que permite al usuario decidir si deja que sus datos sean utilizados para entrenar la IA que tiene detrás sustentando su tecnología de recomendación, de esta manera, el usuario construirá su propio algoritmo de recomendación sin ser conducido por nada externo a los contenidos que se le muestren. Tengo la sensación de que poco a poco tendrá que regularse este aspecto y debería ser normal que fuesemos nosotros los que decidimos si queremos ser dueños de nuestros datos.

En última instancia, mi intención con este trabajo para nada es fomentar que se nos conduzca por recomendaciones que a una entidad le puedan convenir o que haya sesgos, todo lo contrario, he tratado de diseñar un sistema que respete la diversidad y fomente el descubrimiento, utilizando los datos disponibles de forma ética bajo mi punto de vista y mi juicio moral. El objetivo principal era claro: mejorar las tecnologías relacionadas con los sistemas de recomendación, pero sin dejar atrás este capítulo de perspectiva y reflexión sobre su impacto en nuestras vidas y en la sociedad.

5.5. Impacto del Trabajo

5.5.1. Impacto general

Un sistema de recomendación es siempre una herraminenta potencialmente útil, es tanto que influye en las personas en varios aspectos, desde la experiencia del usuario que interactúa con el, hasta la promoción de cualquier cosa, artículo, producto, en nuestro caso: música (algo que nos mueve a todos)

5.5.1.1. Impacto en la tecnología

Este prototipo que hemos desarrollado usa técnicas modernas de agrupamiento (clustering) y similitudes entre usuarios para ofrecer recomendaciones personalizadas, lo cual soluciona pequeños dilemas cotidianos muy comunes como la elección de la música que escuchamos. Además, el diseño del sistema lo considero fácilmente adaptable a otros contextos, lo que nos abriría nuevas posibilidades para su uso en plataformas existentes.

5.5.1.2. Impacto en las personas

Para los usuarios, el sistema pretende y consigue mejorar siempre la experiencia musical dentro de lo posible al ofrecer recomendaciones más personalizadas, más variadas y que enriquezcan el historial de esuchas de cada uno. Esto lo consigue evitando la monotonía de las recomendaciones que solo estan basadas en generos o en popularidad, ayudando a descubrir canciones nuevas o menos conocidas. Especialmente valioso si el usuario pretende ampliar su horizonte y descubir nuevas escuhas que pueda disfrutar.

5.5.1.3. Impacto cultural y social

Desde una perspectiva cultural, este trabajo pretende contribuir (siempre he estado a favor) a dar visibilidad a artistas emergentes y géneros musicales menos populares. Fomentar una mayor diversidad en las escuchas también acerca a personas y enriquece y crea relaciones distintas.

5.5.1.4. Decisiones tomadas de cara al impacto del trabajo

Durante el desarrollo del sistema, tomamos diversas decisiones o caminos que permitiesen maximizar la utilidad del mismo o el impacto positivo, como por ejemplo:

- Seleccionar algoritmos que permitiesen un análisis eficiente.
- Diseñar una interfaz gráfica fácil de usar para cualquier persona.
- Priorizar características que promuevan la diversidad y el descubrimiento musical.

5.5.2. Objetivos de Desarrollo Sostenible

El sistema se alinea con varios Objetivos de Desarrollo Sostenible (ODS) como:

5.5.2.1. ODS 9: Industria, Innovación e Infraestructura

El proyecto promueve innovación ya que combina tecnología avanzada con una aplicación a modelos actuales de plataformas de streaming y puede ser integrado en las ya existentes para mejorarlas.

5.5.2.2. ODS 10: Reducción de las Desigualdades

Al recomendar canciones basadas en características musicales y no solo en popularidad, el sistema permite dar visibilidad a artistas o géneros menos conocidos, haciendoles más fácil el acceso al público y eliminando ciertas desigualdades que pueden venir del sesgo que se produce a la hora de promocionar solo a los grandes.

5.5.2.3. ODS 12: Producción y Consumo Responsables

El sistema anima a que se escuche música variada, lo que promueve un consumo más consciente de la cultura musical.

Bibliografía

[TSK06] P-N Tan, M Steinbach y V Kumar. *Introduction to Data Mining*. Pearson, 2006.

[HKP11] J Han, M Kamber y J Pei. *Data Mining: Concepts and Techniques*. Elsevier, 2011.

[RRS11a] Francesco Ricci, Lior Rokach y Bracha Shapira. *Recommender Systems Handbook*. Springer US, 2011. Cap. 1: Recommender Systems: Introduction and Challenges, págs. 1-36.

[RRS11b] Francesco Ricci, Lior Rokach y Bracha Shapira. *Recommender Systems Handbook*. Springer US, 2011. Cap. 7: Data Mining Methods for Recommender Systems, págs. 227-264.

[RRS11c] Francesco Ricci, Lior Rokach y Bracha Shapira. *Recommender Systems Handbook*. Springer US, 2011. Cap. 6: Context-Aware Recommender Systems, págs. 191-226.

[RRS11d] Francesco Ricci, Lior Rokach y Bracha Shapira. *Recommender Systems Handbook*. Springer US, 2011. Cap. 8: Evaluating Recommender Systems, págs. 265-307.

[RRS11e] Francesco Ricci, Lior Rokach y Bracha Shapira. *Recommender Systems Handbook*. Springer US, 2011. Cap. 25: Diversity in Recommender Systems, págs. 677-703.

[RRS11f] Francesco Ricci, Lior Rokach y Bracha Shapira. *Recommender Systems Handbook*. Springer US, 2011. Cap. 11: Recommender Systems in Industry: A Netflix Case Study, págs. 385-419.

[RRS11g] Francesco Ricci, Lior Rokach y Bracha Shapira. *Recommender Systems Handbook*. Springer US, 2011. Cap. 3: Advances in Collaborative Filtering, págs. 77-117.

[RRS11h] Francesco Ricci, Lior Rokach y Bracha Shapira. *Recommender Systems Handbook*. Springer US, 2011. Cap. 4: Semantics-Aware Content-Based Recommender Systems, págs. 119-159.

[RRS11i] Francesco Ricci, Lior Rokach y Bracha Shapira. *Recommender Systems Handbook*. Springer US, 2011. Cap. 8: Evaluating Recommender Systems, págs. 265-307.

[RRS11j] Francesco Ricci, Lior Rokach y Bracha Shapira. *Recommender Systems Handbook*. Springer US, 2011. Cap. 4: Semantics-Aware Content-Based Recommender Systems, págs. 119-159.

[RRS11k] Francesco Ricci, Lior Rokach y Bracha Shapira. *Recommender Systems Handbook*. Springer US, 2011. Cap. 25: Diversity in Recommender Systems, págs. 677-703.

[RRS11l] Francesco Ricci, Lior Rokach y Bracha Shapira. *Recommender Systems Handbook*. Springer US, 2011. Cap. 11: Recommender Systems in Industry: A Netflix Case Study, págs. 385-419.

[Die21] Diego Calvo. *Cluster Jerárquicos y No Jeráruqicos*. https://www.diegocalvo.es/cluster-jerarquicos-y-no-jerarquicos/. 2021.

[Pan23] Maharshi Pandya. *Spotify Tracks Dataset*. https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset?resource=download. 2023.

[Sci] *Scikit-learn Documentation*. 2023.

[Fou25] Python Software Foundation. *Tkinter — Python Interface to Tcl/Tk*. 2025.

[Mic25] Microsoft Corporation. *Microsoft Power BI*. Último acceso: 9 de enero de 2025. 2025.

[Ama] Amazon. *Amazon*. https://www.amazon.com/. Último acceso: 9 de enero de 2025.

[Spo] Spotify. *Spotify*. https://open.spotify.com/. Último acceso: 9 de enero de 2025.

Apéndice A

Anexo

A.1. Palabras Clave

A continuación, un glosario con palabras clave utilizadas en este TFG:

- **Clustering**: Técnica de agrupamiento utilizada para organizar datos en grupos (o clústeres) basados en sus similitudes
- **Sistema de recomendación**: Tecnología que sugiere contenido personalizado a los usuarios basándose en patrones de comportamiento, preferencias y características de los datos disponibles.
- **Similitud coseno**: Métrica que mide la similitud entre dos vectores, usada en el proyecto para comparar las preferencias de usuarios ficticios.
- **Características de audio**: Propiedades cuantificables de las canciones, como *danceability* (bailabilidad), *energy* (energía), *valence* (positividad), entre otras.
- **Estado de ánimo**: Indicador relacionado con las características *valence* y *energy*, que se utiliza para explorar recomendaciones basadas en estados de ánimo concretos.
- **Burbuja de contenido**: Fenómeno donde las recomendaciones sólo refuerzan gustos que ya estaban presentes en el histórico del usuario, lo cual limita la diversidad del contenido ofrecido.
- **CLARA (Clustering Large Applications)**: Algoritmo de clustering basado en *k-medoids* que se utiliza para analizar grandes conjuntos de datos seleccionando una muestra representativa.
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**: Algoritmo de clustering que agrupa datos basándose en la densidad de puntos, identificando clústeres y valores atípicos (*outliers*).
- **Interfaz gráfica**: Componente visual de un sistema que permite a los usuarios interactuar con las funciones que ofrece el código, facilitando así en este caso la comprensión y el uso del sistema.
- **Normalización**: Proceso de escalar los datos para que estén en un rango comparable, utilizado en este proyecto como preparación de los datos para su posterior

correcto análisis.

A.2. Informe de originalidad generado por la herramienta Turnitin

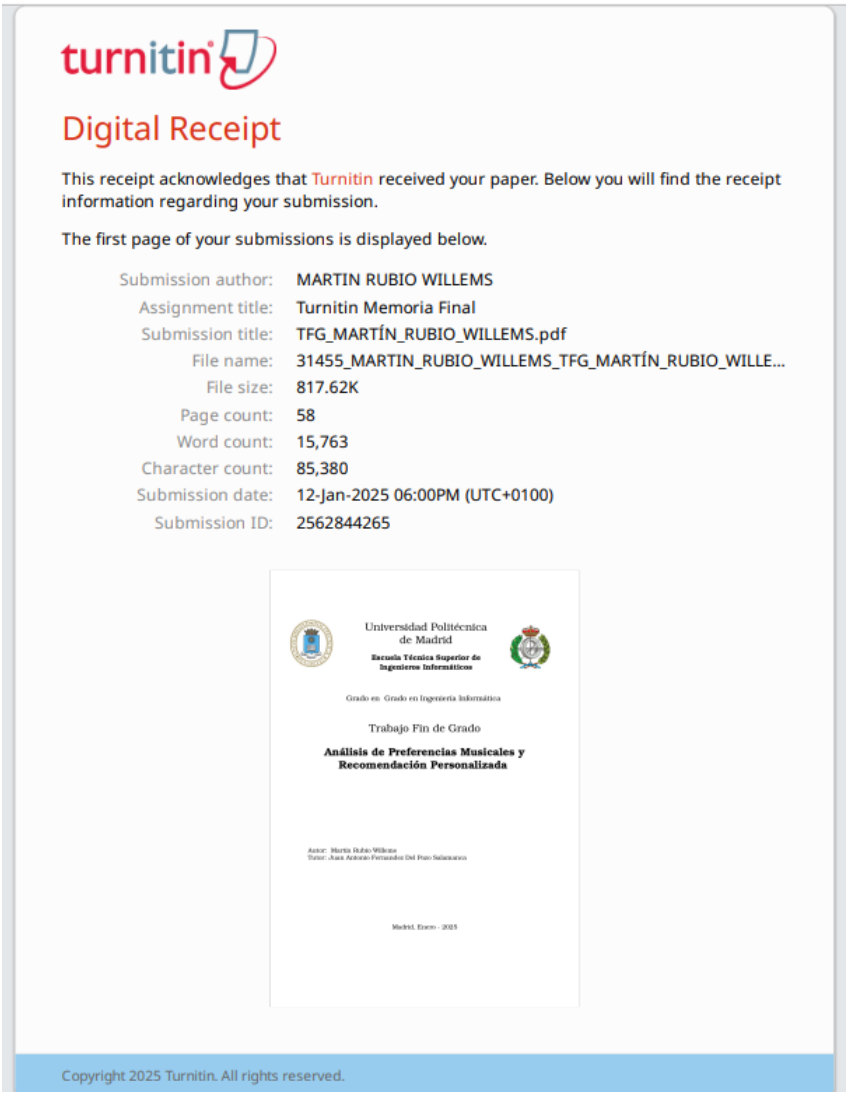

Figura A.1: Turnitin TFGMARTÍNRUBIOWILLEMS verificación coincidencia 2 %

Este documento esta firmado por

	**Firmante**	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
	Fecha/Hora	Mon Jan 13 18:12:38 CET 2025
	Emisor del Certificado	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
	Numero de Serie	561
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)