Técnicas de Agrupamiento Aplicadas a Sistemas de Recomendación Musical: Un Estudio Académico Basado en el Reto de Spotify “Million Playlist Dataset Challenge”

Sara Durango Ceballos
Cristian Camilo Caballero Ayala

Monografía presentada para optar al título de Especialista en Analítica y Ciencia de Datos

Asesor
Javier Fernando Botía Valderrama

Universidad de Antioquia
Facultad de Ingeniería
Especialización en Analítica y Ciencia de Datos
Medellín, Antioquia, Colombia
2024

Cita	(Sara Durango Ceballos & Cristian Camilo Caballero Ayala, 2024),
Referencia **Estilo APA 7 (2020)**	Durango Ceballos, S., & Caballero Ayala, C. C. (2024). Técnicas de Agrupamiento Aplicadas a Sistemas de Recomendación Musical: Un Estudio Académico Basado en el Reto de Spotify Million Playlist Dataset Challenge]. Universidad de Antioquia, Medellín, Colombia.

Especialización en Analítica y Ciencia de Datos, Cohorte VI.

Centro de Investigación Ambientales y de Ingeniería (CIA).

UNIVERSIDAD DE ANTIOQUIA
Vicerrectoría de Docencia

Sistema de Bibliotecas

Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes
Decano: Julio César Saldarriaga Molina
Jefe departamento: Danny Alexandro Múnera Ramírez

Dedicatoria

A nuestras familias por el apoyo recibido

Agradecimientos

A los profesores que nos formaron durante esta especialización

Tabla de contenido

Lista de tablas

Lista de figuras

Siglas, acrónimos y abreviaturas

API	Interfaz de Programación de Aplicaciones
JSON	JavaScript Object Notation - Notación de objeto de JavaScript
LOF	Factor de Localización de Outliers
CSV	Comma-Separated Values
RBF	Radial Basis Function
MSE	Mean Squared Error – Error cuadrático Medio

Resumen

La constante evolución de las plataformas digitales de entretenimiento, como Spotify, ha generado desafíos en la mejora continua de sus sistemas de recomendación musical, los cuales se ven exacerbados por la diversidad de gustos de los usuarios, los cambios en el estado de ánimo y el contexto de éstos, la necesidad de descubrir nueva música y la competencia en la industria del streaming. Con el objetivo de abordar estos desafíos, Spotify ha lanzado el concurso "The Spotify Million Playlist Dataset Challenge", promoviendo la investigación en algoritmos de recomendación musical para aumentar la retención de usuarios, la satisfacción del cliente y el tiempo de uso en la plataforma.

En el marco del concurso, este estudio se apoya en los datos proporcionados por Spotify para desarrollar un modelo predictivo que permita recomendar canciones a partir de patrones de escucha de los usuarios, teniendo en cuenta variables como la duración de las canciones, los niveles de bailabilidad, energía, instrumentalidad y acústica, entre otros; con el propósito de mejorar la personalización y satisfacción del usuario al usar la plataforma.

Los datos utilizados provienen del dataset proporcionado por Spotify para el concurso, el cual comprende más de un millón de playlists y dos millones de pistas. Para abordar esta problemática, se tomó una muestra de mil canciones, seleccionadas para representar una diversidad de géneros y características musicales. Cuando un usuario introduce una playlist, el sistema utiliza algoritmos de agrupamiento para analizar las características de las canciones y generar recomendaciones personalizadas. Este método permite agrupar canciones con atributos similares y sugerir nuevas pistas que se ajustan a los gustos y preferencias del usuario.

Para lograr recomendaciones musicales efectivas, se optó por un modelo de agrupamiento K-means con seis clústers, el cual ha demostrado ser eficaz, alcanzando un error cuadrático medio (MSE) de 0.08%, lo que indica una alta precisión y un bajo sesgo en las recomendaciones generadas, asegurando éstas son tanto relevantes como personalizadas para los usuarios de la plataforma.

Palabras clave — Spotify Million Playlist Challenge, predicción musical, recomendación personalizada, datos, agrupamiento.

https://github.com/SaraDurango/Proyecto-Recomendacion-Musical-Spotify

https://github.com/SaraDurango/Proyecto-Recomendacion-Musical-SpotifyAbstract

The constant evolution of digital entertainment platforms like Spotify has generated challenges in the ongoing improvement of their music recommendation systems. These challenges are exacerbated by the diversity of user tastes, changes in mood and context, the need to discover new music, and competition in the streaming industry. To address these challenges, Spotify has launched the 'The Spotify Million Playlist Dataset Challenge,' promoting research into music recommendation algorithms to increase user retention, customer satisfaction, and time spent on the platform.

In the context of this competition, this study leverages the data provided by Spotify to develop a predictive model that can recommend songs based on user listening patterns, considering variables such as song duration, levels of danceability, energy, instrumentality, and acousticness, among others; with the aim of improving personalization and user satisfaction on the platform.

The data used come from the dataset provided by Spotify for the challenge, which includes over a million playlists and two million tracks. To address this issue, a sample of a thousand songs was taken, selected to represent a diversity of genres and musical characteristics. When a user submits a playlist, the system uses clustering algorithms to analyze the songs' characteristics and generate personalized recommendations. This method groups songs with similar attributes and suggests new tracks that match the user's tastes and preferences.

To achieve effective musical recommendations, a K-means clustering model with six clústers was chosen, which has proven effective, achieving a Mean Squared Error (MSE) of 0.08%. This indicates high precision and low bias in the generated recommendations, ensuring they are both relevant and personalized for platform users.

Keywords — Spotify Million Playlist Challenge, musical prediction, personalized recommendation, data, clustering.

https://github.com/SaraDurango/Proyecto-Recomendacion-Musical-Spotify

1. Descripción del problema

Las plataformas digitales como Spotify enfrentan un desafío constante para mejorar su sistema de recomendación de canciones para los usuarios debido la diversidad de gustos, los cambios de ánimo y de contexto en el que se desenvuelven los usuarios, el descubrimiento de nueva música, la necesidad de generar una experiencia cada vez más personalizada y la competencia de la industria (Arango Archila, 2016)

Dado esto, Spotify propone promover la investigación en los algoritmos de recomendaciones musicales mediante un concurso abierto al público, con el fin de mejorar el sistema de recomendación de canciones buscando aumentar la retención de usuarios, la satisfacción del cliente, el tiempo de uso de la plataforma y el éxito continuo en un mercado altamente competitivo y en constante evolución (Antenucci et al., 2018).

1.1. Problema de negocio

Spotify enfrenta un desafío constante para mejorar su sistema de recomendación de canciones para los usuarios debido a varias razones:

• Diversidad de gustos: Los usuarios de Spotify tienen una amplia gama de preferencias musicales. Algunos pueden disfrutar de varios géneros y artistas, lo que hace que sea un desafío proporcionar recomendaciones precisas que se adapten a gustos cambiantes y diversos.

• Descubrimiento de nueva música: Los usuarios esperan que el sistema de recomendación no sólo les ofrezca música que ya conocen y les gusta, sino que también les ayude a descubrir nueva música que sea relevante para sus gustos.

• Experiencia personalizada: La competencia en el mercado de la transmisión de música es intensa, y los servicios como Spotify buscan diferenciarse ofreciendo una experiencia altamente personalizada. Esto implica la necesidad de refinar constantemente los algoritmos de recomendación para adaptarse a las preferencias de cada usuario.

• Evitar la fatiga del usuario: Si un usuario recibe recomendaciones irrelevantes con frecuencia, puede llevar a la insatisfacción y, en última instancia, a la pérdida de interés en el servicio. Por lo tanto, mejorar la precisión de las recomendaciones es fundamental para mantener a los usuarios satisfechos.

• Competencia en la industria: La competencia en el mercado de la música en streaming es feroz. Para retener y atraer a más usuarios, Spotify debe ofrecer un sistema de recomendación superior que no solo se base en lo que ya conocen los usuarios, sino que también brinde valor al descubrir nueva música de manera efectiva.

Dado esto, mejorar el sistema de recomendación de canciones es esencial para la retención de usuarios, la satisfacción del cliente y el éxito continuo de Spotify en un mercado altamente competitivo y en constante evolución.

1.2.Aproximación desde la analítica de datos

Un modelo de Machine Learning puede mejorar la predicción de canciones a añadir automáticamente a una lista de reproducción de acuerdo con diversos patrones de comportamiento del usuario en la plataforma. Para desarrollar dicho modelo se pueden explorar diferentes técnicas hasta encontrar la que mejor pronostique la recomendación de reproducción automática de acuerdo con los gustos de los usuarios.

Para esto se desarrollarán las siguientes fases:

• Preparación de datos:

El dataset del reto está disponible en el sitio web de AIcrowd https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge, y tiene un tamaño total aproximado de 5 GB. Dado los extensos requisitos y los significativos recursos computacionales necesarios para participar en el reto completo, este proyecto se propone como una aproximación académica optando por trabajar sólo con una fracción seleccionada del conjunto de datos.

Adicionalmente, se propone complementar la información proporcionada mediante una consulta de características musicales a través de una API de Spotify para desarrolladores, la cual permite extraer información adicional relevante acerca de las canciones del dataset y se encuentra en el siguiente enlace https://developer.spotify.com/documentation/web-api/reference/get-playlist. Esta aproximación es escalable, permitiendo la expansión del dataset de recomendaciones según sea necesario, en función de los futuros requerimientos y decisiones de la academia. Por lo anterior, el modelo funcionará como un acercamiento académico a una porción del desafío inicial.

• **Creación de modelo de aprendizaje no supervisado:** Se utilizarán las características musicales consultadas de las listas de reproducción y las pistas extraídas como base de

recomendación para ser cruzadas con los datos identificados en la playlist entregada por el usuario. Por lo que antes de usar el modelo se debe asegurar la equiparación de ambos dataset para que contengan los mismos componentes y variables, garantizando así la correcta comparación y procesamiento de la información.

• **Generación de recomendaciones:** Cuando un usuario proporciona una lista de reproducción, el algoritmo puede identificar las canciones con mayor similitud en función de sus características musicales. Dichas canciones pueden recomendarse como continuación automática para la lista de reproducción del usuario.

• **Ajuste de hiperparámetros:** A medida que se realizan iteraciones sobre los datasets disponibles, se deben ajustar los hiperparámetros definidos para asegurar que el modelo alcanza el rendimiento óptimo y responde adecuadamente a las necesidades específicas del problema.

• **Evaluación y validación:** Evaluar el rendimiento del modelo utilizando tanto métricas de agrupamiento como métricas de negocio para verificar la calidad de las recomendaciones generadas. Estas métricas se profundizarán más adelante.

1.3. Origen de los datos

Como se mencionó anteriormente, la fuente de datos primaria corresponde al dataset preparado por Spotify de acceso público para el reto de recomendación automática. Dicho dataset comprende 1 millón de listas de reproducción, consta de más de 2 millones de pistas únicas de casi 300 000 artistas y representa el conjunto de datos público de listas de reproducción de música más grande del mundo. Para este proyecto, se tomará un slice de mil canciones de dicha base de datos; cada slice tiene una estructura en *formato JSON* entregada como diccionario donde una playlist representa una agrupación de primer nivel que se comienza a expandir a medida que se profundiza en la composición. La extensión de cada playlist varía en función del número de canciones que contiene, el cual puede oscilar desde uno hasta 'n' canciones, dependiendo de lo almacenado por el usuario en el archivo original. Un extracto de playlist almacenado en el formato original se puede ver de la siguiente manera, (C.W. Chen, P. Lamere, M. Schedl, & H. Zamani, 2018):


```
{
    "name": "musical",
    "collaborative": "false",
    "pid": 5,
    "modified_at": 1493424000,
    "num_albums": 7,
```

```
    "num_tracks": 12,
    "num_followers": 1,
    "num_edits": 2,
    "duration_ms": 2657366,
    "num_artists": 6,
    "tracks": [
      {
        "pos": 0,
        "artist_name": "Degiheugi",
        "track_uri": "spotify:track:7vqa3sDmtEaVJ2gcvxtRID",
        "artist_uri": "spotify:artist:3V2paBXEoZIAhfZRJmo2jL",
        "track_name": "Finalement",
        "album_uri": "spotify:album:2KrRMJ9z7Xjoz1Az4O6UML",
        "duration_ms": 166264,
        "album_name": "Dancing Chords and Fireflies"
      },
      {
        "pos": 1,
        "artist_name": "Degiheugi",
        "track_uri": "spotify:track:23EOmJivOZ88WJPUbIPjh6",
        "artist_uri": "spotify:artist:3V2paBXEoZIAhfZRJmo2jL",
        "track_name": "Betty",
        "album_uri": "spotify:album:3lUSlvjUoHNA8IkNTqURqd",
        "duration_ms": 235534,
        "album_name": "Endless Smile"
      }
    ],
}
```


Puede observarse que la información inicial obtenida con los archivos *JSON* preprocesados no resulta suficiente para el modelo, pues proporciona variables que son poco representativas, en términos de caracterización musical de las canciones. De ahí el planteamiento de expandir la información, añadiendo características musicales más detalladas que amplíen el espectro de recomendación para cada canción en el slice de playlist procesadas, mediante la consulta en la API de Spotify para desarrolladores mencionada anteriormente. Para incorporar estas características adicionales, se utilizará la librería Spotipy, una herramienta para Python que actúa como un puente local hacia la API de Spotify. Esta librería permite interpretar y extender la información a la codificación requerida para el modelo. Cabe destacar que Spotipy no puede operar de manera independiente; requiere de dos variables de autenticación proporcionadas por el módulo de

desarrolladores de Spotify, denominadas “Client_Secret” y “Client_Id”. Estas claves habilitan a Spotipy para realizar consultas directas a Spotify.

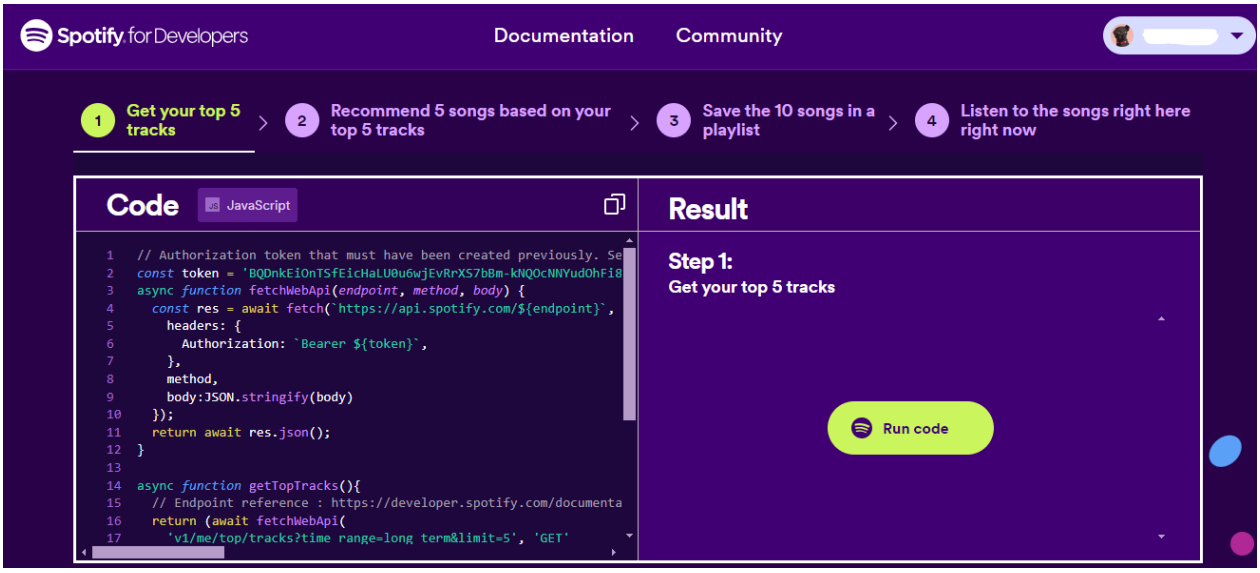


Figura 1.Interfaz API Spotify para desarrolladores

1.4. Métricas de desempeño

1.4.1. Métricas de evaluación del modelo o Métricas de clustering

Para asegurar la calidad de los clústers generados por el modelo de recomendación musical, se emplearán diversas métricas de evaluación:

- **Coherencia de Clústers:** Se mide mediante la varianza intra-clúster de características clave como bailabilidad, energía, tempo, etc.
- **Silhouette Score:** Esta métrica evalúa cuán bien una canción se ajusta a su clúster en comparación con otros clústers.
- **Davies-Bouldin Score:** evalúa la calidad del clustering calculando el ratio de dispersión dentro de cada clúster y la separación entre clústers.
- **Calinski-Harabasz Score:** mide la relación entre la dispersión dentro de los clústers y la dispersión entre ellos.

1.4.2. Métricas de recomendación

- **Similitud entre características:** Para medir la precisión en la recomendación musical, se calculará la similitud entre las canciones recomendadas y las de la playlist del usuario a través de la **distancia entre las características de las canciones recomendadas y las originales**. Una canción recomendada se considera relevante si la distancia entre su perfil de características y el rango o promedio de las características de las canciones en la playlist del usuario es pequeña.
- **Precisión en la recomendación:** está dada por el MSE (error cuadrático medio), el cual calcula la diferencia cuadrada promedio entre las distancias de las canciones recomendadas y las canciones en la playlist del usuario, proporcionando un indicador cuantitativo del rendimiento del modelo.

$$\boldsymbol{MSE} = \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (\boldsymbol{y_i} - \boldsymbol{\hat{y}_i})^{\mathbf{2}}$$

Ecuación1. Error Cuadrático Medio

Donde:

- n es el número de canciones recomendadas
- $\boldsymbol{y_i}$ es el valor real de la característica de la i-ésima canción procesada de la playlist del usuario.
- y$\boldsymbol{\hat{y}_i}$ es la a i-ésima canción recomendada.
- $(\boldsymbol{y_i} - \boldsymbol{\hat{y}_i})^{\mathbf{2}}$ es el cuadrado de la diferencia entre el valor real y el valor de la recomendación.

2. Objetivos

2.1.Objetivo general

Desarrollar un algoritmo de recomendación musical utilizando la base de datos "The Spotify Million Playlist Dataset Challenge" para generar recomendaciones personalizadas a partir de patrones de escucha de los usuarios, teniendo en cuenta variables como energía, bailabilidad, tempo, instrumentalidad y acústica, entre otras; proporcionando una experiencia de usuario más personalizada y satisfactoria.

2.2.Objetivos específicos

- Aplicar diferentes algoritmos de agrupamiento de datos para desarrollar el modelo de recomendación musical, experimentando con diversos parámetros y configuraciones de modelos para identificar la configuración que maximice la precisión y relevancia de las recomendaciones.
- Explorar diferentes técnicas de balanceo de datos para asegurar una representación equitativa de las diversas características musicales en el modelo de recomendación, incluyendo métodos como sobre muestreo y submuestreo, entre otras técnicas, para optimizar la distribución de las clases y mejorar la precisión de las recomendaciones.
- Implementar tanto métricas estándar de agrupamiento como métricas de precisión en la recomendación para evaluar la efectividad del sistema de recomendación musical y determinar cómo las canciones recomendadas cumplen con las expectativas y preferencias de los usuarios con base en el análisis de las canciones procesadas.

3. Datos

3.1. Datos originales

Los datos iniciales parten del preprocesamiento del archivo entregado en el desafío planteado por Spotify, correspondiente a un archivo con extensión *.zip* de aproximadamente 5 GB (Spotify_Million_Playlist_Dataset). Debido al gran tamaño del dataset, no es posible procesarlo en entornos en la nube, por lo que se opta por realizar el procesamiento de manera local utilizando *Jupyter Lab*. Este enfoque permite la apertura, procesamiento y extracción inicial de los datos. En consecuencia, en el repositorio de GitHub se incluye únicamente el resultado final del procesamiento y no el dataset original, para optimizar el uso de recursos y facilitar el acceso a la información. Sin embargo, se proporciona el enlace del desafío, en el cual se puede descargar el dataset original si así se desea https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge#challenge-dataset.

- **Descripción de variables**

En la siguiente tabla se describen las variables que comprenden los datos proporcionados por Spotify

Variable	Definición
Nombre Playlist	Nombre de una playlist de Spotify
Número de Pistas	Número de canciones dentro de una playlist de Spotify
Duración Playlist	Duración total de toda la lista de reproducción (en ms)
Número de Seguidores	Número de seguidores de la playlist
Nombre Pista	Nombre de la canción o pista de la playlist
Nombre Artista	Nombre del artista
Nombre Álbum Musical	Nombre del Álbum Musical al que corresponde la pista
Duración Pista	Duración de una pista de la lista de reproducción (en ms)
Posición	Posición de la pista dentro de la playlist
Track URI	Identificación o ID de la pista dentro de Spotify
Artist URI	Identificación o ID del artista dentro de Spotify
Álbum URI	Identificación o ID del álbum musical dentro de Spotify

Tabla 1. Descripción de variables Dataset Spotify

En la siguiente tabla se muestran las variables añadidas mediante la consulta en la API y las cuales corresponden a las características musicales de las canciones del dataset

Variable	Significado	Descripción
Danceability	Bailabilidad	Aptitud de una canción para el baile basada en una combinación de elementos musicales
Energy	Energía	Medida perceptiva de intensidad y actividad en una canción
Key	Tonalidad	Tonalidad en la que está compuesta la canción
Loudness	Volumen	Volumen general de una canción, medido en decibeles
Mode	Modalidad	Modalidad (mayor o menor) de una canción, donde mayor es 1 y menor es 0
Speechiness	Vocalidad	Presencia de palabras habladas en una canción
Acousticness	Acústica	Medida de la acústica de una canción
Instrumentalness	Instrumentalidad	Nivel de instrumentalidad
Liveness	Presencia en Vivo	Presencia de audiencia en la grabación
Valence	Valencia	Positividad musical transmitida por una canción
Tempo	Tempo	Tempo general de una canción en pulsos por minuto (BPM)
Time_signature	Compás	Medida del compás de una canción
Genre	Género	Género musical asociado a cada canción, derivado del artista.

Tabla 2. Descripción de variables añadidas mediante API de Spotify

3.2. Datasets – Transformación de datos

Se cargaron las playlist extraídas mediante el preprocesamiento *JSON* mencionado anteriormente, para extraer información relevante y almacenarla en archivos *CSV* y, posteriormente, se realiza una transformación inicial que consta de:

- Eliminación de columnas que no aportan valor, como la fecha de última actualización de la playlist, el número de modificaciones y columnas en blanco
- Renombrado de algunas columnas para hacerlas más legibles
- Eliminación de listas de reproducción vacías o sin ninguna pista añadida
- Conversión de las variables de tiempo de milisegundos a segundos
- Escalamiento de variables numéricas
- Acotamiento de los datos mediante de selección de mil canciones
- Eliminación de outliers combinando los métodos Z-score ajustado y Factor de Localización de Outliers (LOF).

3.3. Analítica descriptiva

Ranking de artistas

```
Ranking para 'artist_name':
Drake                   4876
Kanye West              2589
Kendrick Lamar          1900
Rihanna                 1733
The Weeknd              1644
                        ...
Goodbye Tomorrow           1
Jay & The Techniques       1
D-Devils                   1
Cover Queens               1
Ashley DuBose              1
Name: artist_name, Length: 14034, dtype: int64
```

Figura 2. Ranking de artistas en playlist de Spotify

De acuerdo con la figura 2, el artista con mayor participación en las playlist de Spotify es Drake, seguido de Kanye West, Kendrick Lamar, Rihanna y The Weekend. Mientras que los álbumes cuyas canciones tienen mayor presencia en playlist de Spotify son Views del artista Drake, Coloring Book de Chance the Rapper, Stoney More Life de Post Malone y The Life of Pablo de Kanye West, como se ilustra en la figura 3.

```
Ranking para 'album_name':
Views                              1126
Coloring Book                       805
Stoney                              762
More Life                           743
The Life Of Pablo                   713
                                   ...
In Blue                               1
Decompositions                        1
Mr. Put It Down (Remixes)             1
Leyendas Solamente Los Mejores        1
Be You                                1
Name: album_name, Length: 28009, dtype: int64
```

Figura 3.Ranking de álbumes musicales en playlist de Spotify

Distribución de variables numéricas

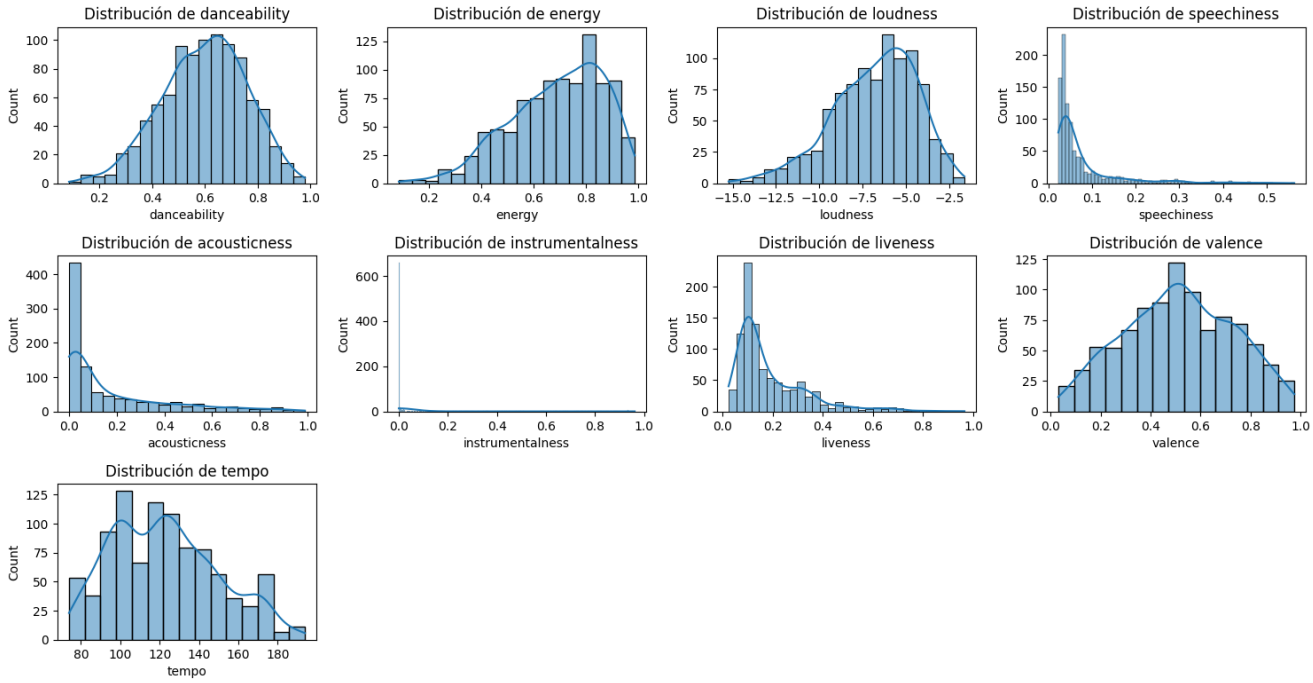
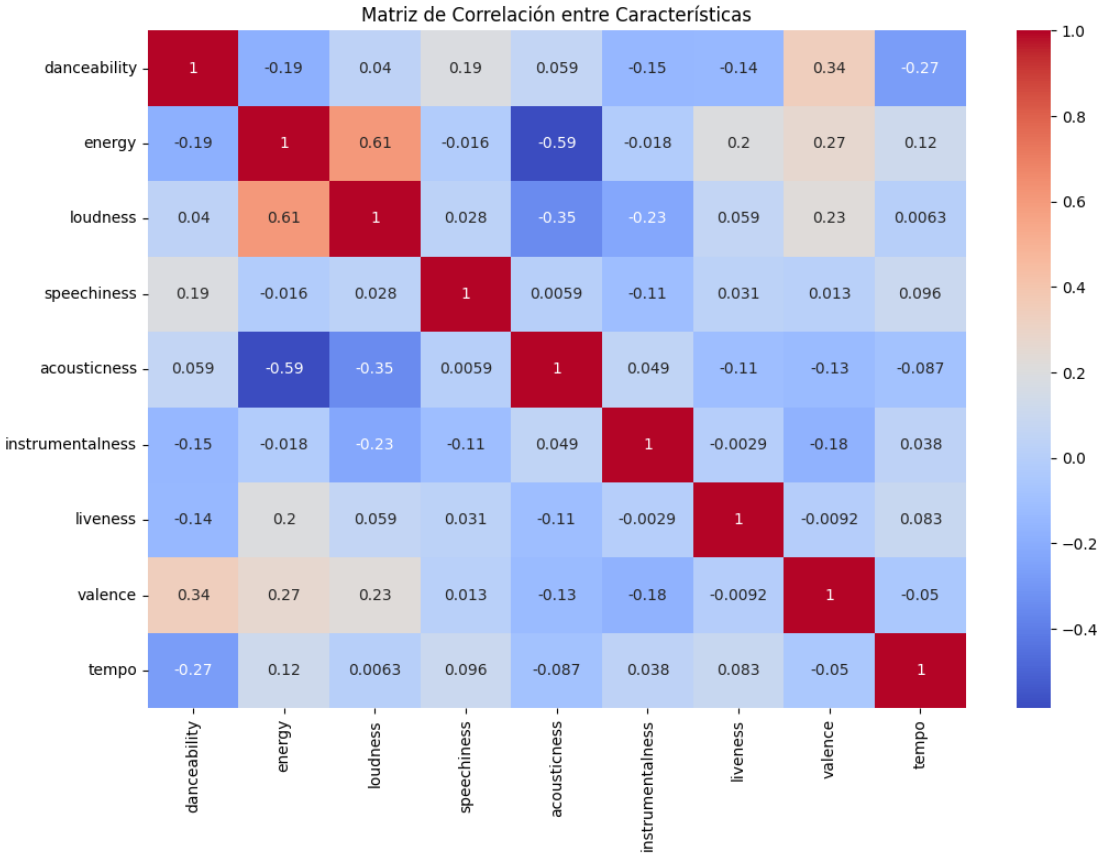


Figura 4. Distribución de variables numéricas

De acuerdo con las distribuciones observadas, puede inferirse que la bailabilidad tiene una distribución aproximadamente normal centrada alrededor de 0.5, indicando que, en el conjunto de datos, las canciones tienden a tener una bailabilidad intermedia. Similar a esta variable, los niveles de energía y valencia siguen una distribución normal con tendencia a valores altos, sugiriendo que la mayoría de las canciones poseen un nivel de energía moderadamente alto y con percepción de transmitir positividad.

Por otro lado, el tempo tiene una distribución multimodal, con varios picos distintos, lo que sugiere diferentes agrupaciones de canciones por su tempo en BPM (pulsos por minuto), mientras que en las variables acústica, instrumentalidad y “liveness” (en vivo), se observan distribuciones sesgadas hacia valores bajos, lo que significa que hay muchas canciones en el conjunto de datos que presentan niveles bajos para estas características,

Correlación entre variables



El gráfico anterior evidencia una correlación positiva moderada (0.61) entre energía y sonoridad, lo que sugiere que las canciones con mayor energía tienden a ser más sonoras. Por otro lado, la energía y la acústica tienen una correlación negativa moderada (-0.59), indicando que las canciones más enérgicas tienden a ser menos acústicas. Esto se alinea con la idea de que las canciones electrónicas o de rock, por ejemplo, suelen ser más enérgicas y menos acústicas en comparación con géneros como el clásico.

También, se puede afirmar que existe una correlación moderadamente positiva (0.34) entre la valencia y la bailabilidad. Esto significa que las canciones que generan emociones más positivas (alta valencia) también suelen ser más aptas para bailar.

Finalmente, no se evidencia una correlación altamente significativa entre variables, lo que sugiere una relación de independencia entre las mismas e indica que no son variables redundantes.

4. Proceso de analítica

4.1. Pipeline principal

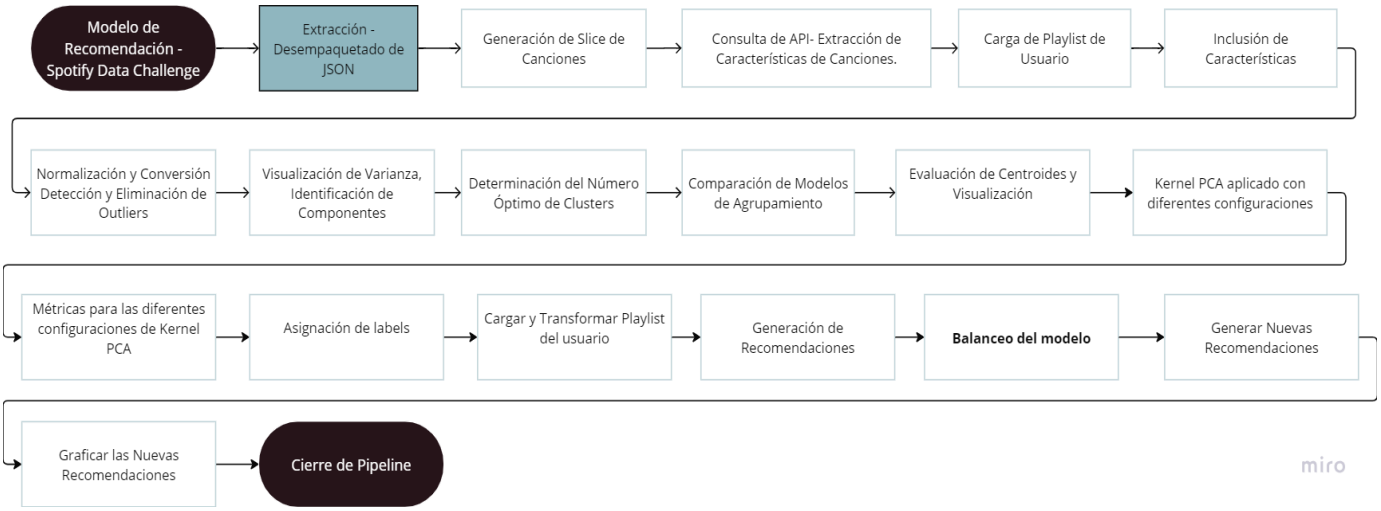


Figura 5. Pipeline Principal

La figura 5 ilustra el diagrama de flujo implementado en el proyecto, el cual se describe a continuación

4.2. Preprocesamiento

En primera instancia, se desempaquetan los archivos *JSON* para comenzar con la depuración de los datos brutos, utilizando la biblioteca *JSON* para generar los correspondientes slices y facilitar la lectura. Para asegurar la viabilidad computacional y evitar el sobre procesamiento, dado el enfoque académico del proyecto, se implementa una función que limita la cantidad de datos leídos. La función `loop_slices` se define de la siguiente manera:

```
def loop_slices(file_path, limit=1000):
with open(file_path, 'r') as file:
playlists = json.load(file)
return playlists[:limit]
```

Posteriormente, se almacena la información extraída del *JSON* en su estado original, creando así un primer archivo *CSV*. Este archivo sirve como punto de partida para convertir los datos del *JSON* en formato *CSV*, conteniendo las playlists y canciones, pero aún sin las características adicionales

- **Complemento de Datos Originales - API -Spotipy:**

A través de la biblioteca Spotipy se entregan las siguientes credenciales de autenticación para el acceso a la API:

Spotify credentials
os.environ["SPOTIPY_CLIENT_ID"]
os.environ["SPOTIPY_CLIENT_SECRET"]
os.environ['SPOTIPY_REDIRECT_URI']
sp = spotipy.Spotify(client_credentials_manager=SpotifyClientCredentials())

Tabla 3. Credenciales para autenticación en API

Una vez se tienen las credenciales, se puede iniciar con el proceso de consulta de características musicales mediante la API. Cabe resaltar que dicha consulta está limitada a 100 canciones por llamado, y que, si se sobrepasa este límite de uso diario se puede incurrir en bloqueos por parte de Spotify. Para gestionar eficientemente este límite y evitar la interrupción del acceso, se implementó un bucle permitiendo controlar el número de llamados realizados. Esta restricción fue uno de los desafíos iniciales del proyecto, requiriendo un manejo estratégico de las solicitudes a la API para mantener la continuidad en el procesamiento de los datos.

Posteriormente, se utilizó la función `sp.audio_features(list_uri_songs)` de la librería Spotify para extraer y almacenar las características detalladas de cada canción (bailabilidad, energía, tempo, etc., mencionadas anteriormente).

Una vez asignadas las variables a las canciones procesadas, se genera el *CSV* inicial utilizado como archivo base de recomendación para el modelo y con el que se realiza el entrenamiento correspondiente.

- **Preparación de Playlist de Usuario**

Luego de preparar la base de recomendación es necesario el eje complementario que consta de la playlist del usuario a procesar, a partir de la cual se realiza la recomendación. Para esto es necesario utilizar una playlist ya existente de tipo pública para que pueda ser consultada también a través de la API de Spotify, haciendo el llamado utilizando un identificador único especificado del siguiente modo

```
playlist_uri = 'spotify:playlist:447Mg9UnLbA3OPYuaTWbN7'
```

De este modo, se genera un nuevo archivo *CSV* correspondiente a la playlist del usuario, el cual comprende las nuevas canciones consultadas y las mismas características que el dataset original, para tener archivos equiparables al momento del cruce.

- **Detección y eliminación de outliers**

-Método de Desviación Absoluta Mediana (MAD):

Con este método se identificaron outliers con base en la mediana de desviaciones absolutas de los datos, proporcionando un enfoque robusto especialmente en distribuciones no normales.

-Factor de Localización de Outliers (LOF):

Se analizó la densidad local de puntos para identificar aquellos que diferían significativamente en densidad respecto a sus vecinos.

Finalmente, se combinaron ambos métodos para una eliminación más robusta y precisa de outliers

- **Identificación de componentes principales**

Se graficó la varianza acumulada explicada para determinar el número óptimo de componentes. De acuerdo con la figura 6 y teniendo en cuenta el método del codo, se puede concluir que con 7 componentes se puede explicar un poco más del 90% de la varianza. Por tanto, para reducir complejidad del análisis, se trabajará con 7 componentes. Este resultado se usará posteriormente para reducción de dimensionalidad

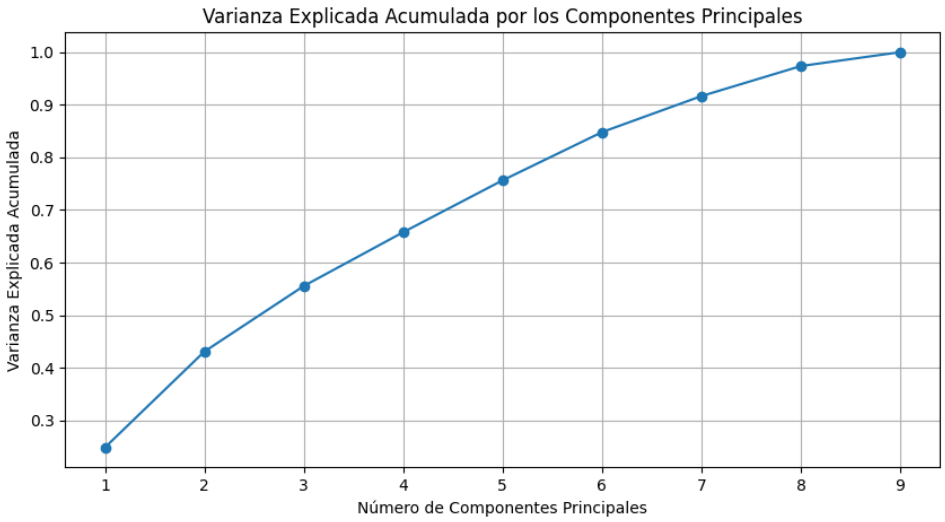


Figura 6. Varianza explicada acumulada por componentes principales

4.3.Modelos

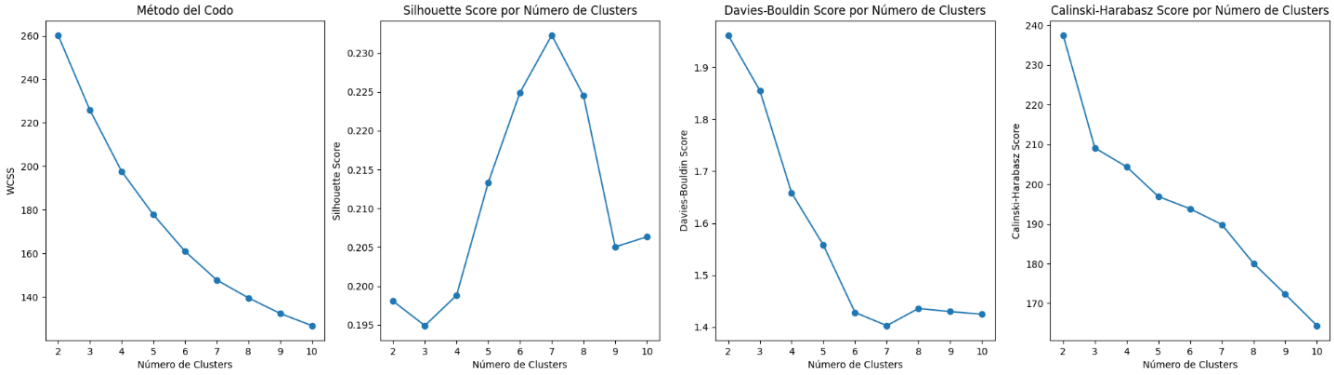


Figura 7.Visualización de número óptimo de clústers según diferentes métricas

Como se observa en la figura anterior, se realizó la validación del número óptimo de clústers utilizando diferentes métodos como el del codo, Silhouette Score, Davies-Bouldin y Calinski-Harabasz y, aunque de manera mayoritaria el número de clústeres definía en siete, el modelo Calinski generaba como óptimo dos clústers

Dado esto, es pertinente iterar entre diferentes configuraciones de número de clústers y modelos de agrupamiento.

Se proponen tres diferentes modelos de clústerización: K-means, agrupamiento jerárquico y agrupamiento espectral y se obtuvo el siguiente top 5 para los mejores modelos

Modelo	Configuración	Silhouette Score↑	Davies-Bouldin Score↓	Calinski-Harabasz Score↑	Cluster Coherence↓
KMeans	n_clústers=7, n_init=10	0.232298	1.402.806	189.821.956	0.022640
KMeans	n_clústers=7, n_init=20	0.232298	1.402.806	189.821.956	0.022640
KMeans	n_clústers=8, n_init=20	0.227322	1.420.615	180.702.652	0.021425
KMeans	n_clústers=6, n_init=10	0.224915	1.428.161	193.788.717	0.024368
KMeans	n_clústers=6, n_init=20	0.224915	1.428.161	193.788.717	0.024368

Tabla 4. Top 5 de mejores modelos de agrupamiento

De acuerdo con lo anterior, K-means con 7,6 y 8 clústers son las configuraciones con mejores resultados

- **Reducción de dimensionalidad**

Una vez identificado el mejor modelo, se aplicó reducción de dimensionalidad para trabajar con 7 componentes, de acuerdo con el resultado previo en el análisis de varianza explicada.

Inicialmente, se experimentó aplicando el modelo Kmeans con 7 clústers en tres escenarios: datos originales, datos con PCA y datos normalizados; posteriormente se graficaron los centroides de los clústers obtenidos

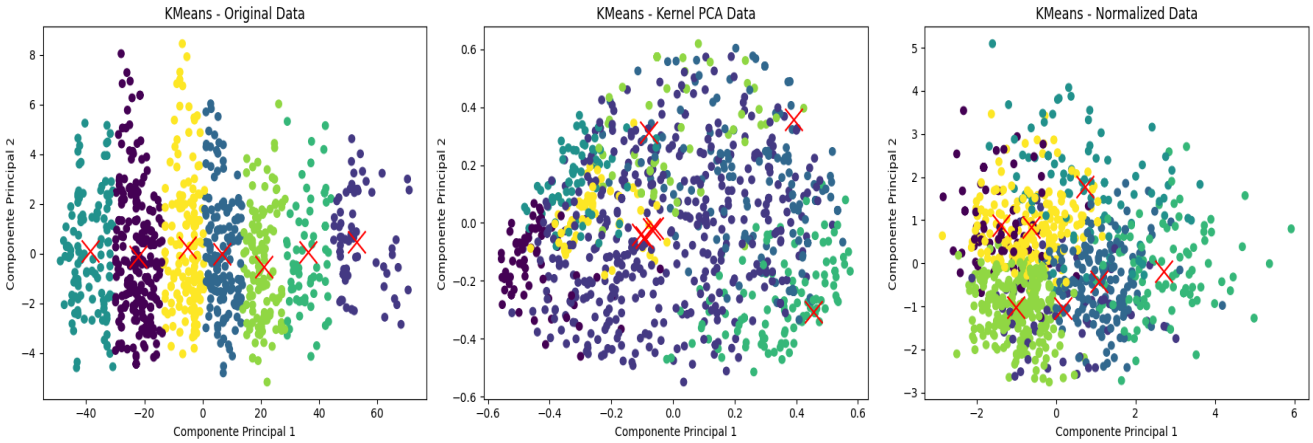


Figura 8.Visualización de centroides para Kmeans con datos originales, kernel PCA y datos normalizados

De acuerdo con la figura 8, la reducción de dimensionalidad no estaba generando suficiente separabilidad, así que se evaluaron diferentes configuraciones de kernel y parámetros para buscar el mejor resultado, tanto en separabilidad como en métricas para el modelo.

Dentro de las configuraciones se contemplaron varios kernels: RBF, polinomial de diferentes grados, sigmoidal, lineal y de coseno; iterando con diferentes valores de gamma y número de clústers.

Una vez generadas las iteraciones correspondientes, se concluye que **el modelo más adecuado es K-means con 6 clústers aplicando una reducción de dimensionalidad PCA con kernel RBF y gamma 0,9**, como se muestra en la tabla 5 y en la figura 9.

Tanto el modelo como la reducción de dimensionalidad mencionada y el método de escalamiento de los datos se guardaron en extensión *.joblib* para posterior aplicación a la playlist del usuario, garantizando replicabilidad en estos procedimientos.

Kernel	Parámetros	Silhouette Score	Davies-Bouldin Score	Calinski-Harabasz Score	Cluster Coherence
rbf	{'gamma': 0.1}	0.259464	1.475.121	196.699.577	0.028019
rbf	{'gamma': 0.9}	0.692613	0.633611	405.325.524	0.007214
poly	{'degree': 2, 'coef0': 1}	0.239954	1.506.649	180.188.585	0.158193
poly	{'degree': 3, 'coef0': 1}	0.255672	1.445.570	200.812.189	0.277199
sigmoid	{'gamma': 0.2, 'coef0': 0}	0.209561	1.573.649	151.736.430	0.089859
sigmoid	{'gamma': 0.5, 'coef0': 0}	0.201072	1.594.086	148.194.714	0.142666
linear	{}	0.213584	1.554.206	155.453.943	0.545757
cosine	{}	0.218811	1.546.051	158.068.423	0.077215

Tabla 5. Métricas para K-means con 6 clústers y diferentes configuraciones de Kernel PCA

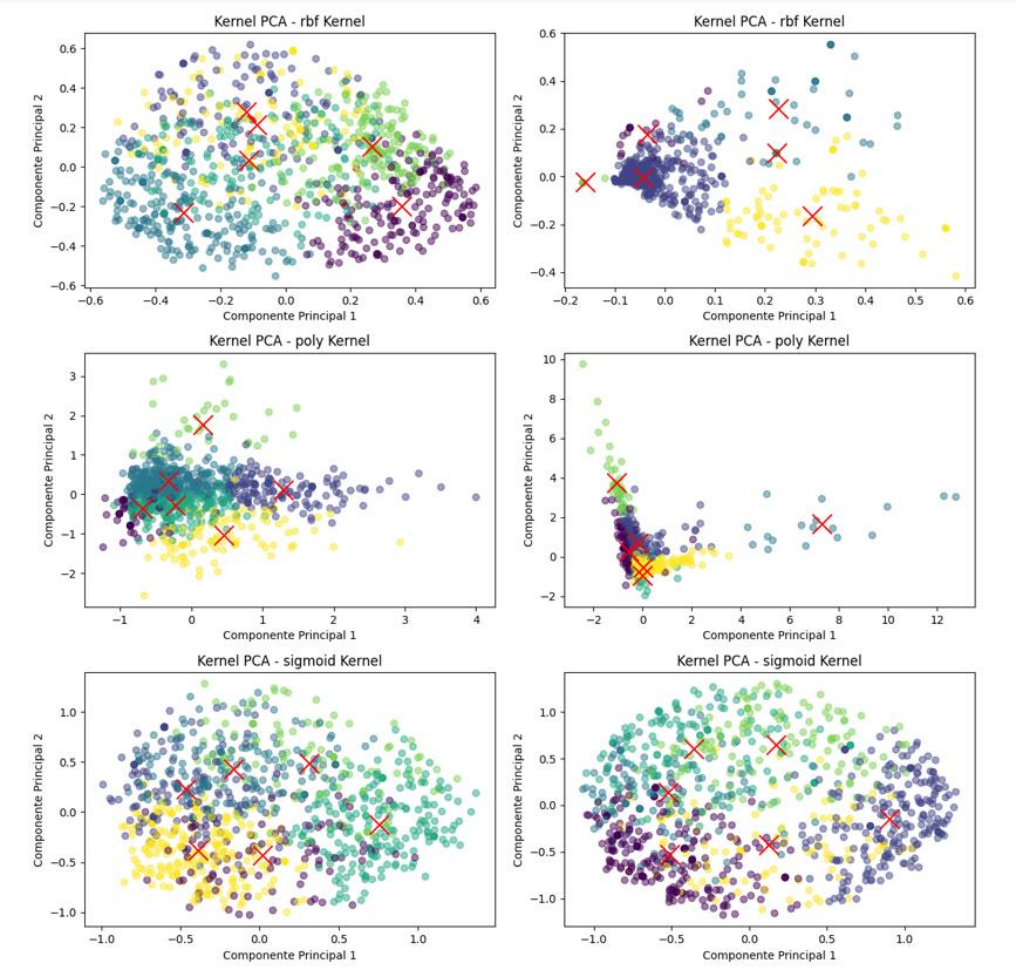


Figura 9. Visualización de Centroides para diferentes configuraciones de Kernel PCA

Una vez elegido el mejor modelo con su respectiva configuración, se establecen las siguientes etiquetas para los clústers generados, de acuerdo a las características principales de las canciones que los componen

Clúster ID	Nombre	Significado
0	High Energy, Moderate Danceability	Energía Alta, Bailabilidad Moderada
1	High Danceable, Moderate Energy	Bailabilidad Alta, Energía Moderada
2	Calm, Less Danceable	Calmada, Poco Bailable
3	High Intensity	Alta Intensidad, máximo volumen
4	Intense Tempo	Máximo Tempo, ritmos rápidos
5	High Valence, Positive Vibes	Alta Valencia, sonidos alegres con energía moderada

Tabla 6. Etiquetas para clústers generados

En la siguiente tabla se ilustran algunos ejemplos de canciones agrupadas en el clúster **Energizing, High Danceable**

Clúster 1 - High Danceable, Moderate Energy	
Track Name	**Artist Name**
Lose Control (feat. Ciara & Fat Man Scoop)	Missy Elliott
Toxic	Britney Spears
Crazy In Love	Beyoncé
Rock Your Body	Justin Timberlake
It Wasn't Me	Shaggy

Tabla 7.Ejemplos de canciones en el clúster High Danceable, Moderate Energy

Lo anterior permite observar coherencia en la interpretación de características musicales para el agrupamiento.

- **Procesamiento de Playlist de Entrada**

Se carga una playlist real de usuario en Spotify que será procesada y utilizada como punto de partida para la generación de las recomendaciones. Inicialmente se extrae la playlist de manera similar a los *JSON* iniciales y se equiparan las características necesarias para que el modelo las procese correctamente. De esta manera se tiene un archivo de entrada y de salida estructurado bajo el mismo modelo.

A través del modelo guardado previamente (`best_kmeans_kpca_model.joblib`) se procesa la playlist de entrada para realizar una primera recomendación.

4.4.Métricas

Utilizando las librerías Numpy y Scikit-learn, se realizaron validaciones del modelo seleccionado, enfocándose en diversas métricas de agrupamiento como el Silhouette Score, Davies-Bouldin Score, Calinski-Harabasz Score y Cluster Coherence. Este enfoque permitió optimizar tanto la ejecución como el desarrollo del proyecto, utilizándolas en las diferentes iteraciones realizadas.

Posteriormente, se evaluaron métricas de negocio centradas en la distancia promedio entre las canciones recomendadas y las procesadas, permitiendo determinar la cercanía de las recomendaciones con respecto a los datos originales. Finalmente, se utilizó el Mean Squared Error (MSE) para evaluar la precisión del modelo, comparando las canciones recomendadas con las procesadas.

5. Metodología

5.1. Baseline

Para establecer una línea base en el modelo de recomendación musical, se adoptó una metodología que inicia con la selección de un modelo preliminar basado en las métricas de agrupamiento definidas. Tras esta selección inicial, se procede a generar iteraciones, cambiando diferentes parámetros como número de clústers, configuraciones de reducción de dimensionalidad y métodos de balanceo de datos.

Cada nueva configuración se evalúa rigurosamente para determinar su eficacia y se compara con las anteriores, permitiendo refinar el modelo gradualmente hasta encontrar la configuración que genera los mejores resultados.

5.2. Validación

Una vez aplicado, el modelo fue sometido a validaciones de recomendación utilizando técnicas de validación cruzada *k-fold* para evaluar su robustez. Además, se dividió el conjunto de datos en partes de entrenamiento y prueba, asegurando que no se introdujeran sesgos al no usar la totalidad del dataset generado por la playlist. Esto garantizó que los datos de prueba no influyeran en el entrenamiento del modelo, manteniendo la integridad de la validación

5.3.Iteraciones y evolución

Como se mencionó anteriormente, el desarrollo del modelo de recomendación musical implicó varias fases de iteración y refinamiento para mejorar su precisión y rendimiento. A continuación, se describen los pasos clave y las mejoras implementadas en cada iteración:

- **Evaluación Inicial de Modelos de Agrupamiento:**

Las primeras iteraciones se centraron en evaluar diversos modelos de agrupamiento, variando sus parámetros para identificar la configuración que ofreciera los mejores resultados según las métricas de agrupamiento. Este proceso permitió seleccionar el modelo de agrupamiento más efectivo para las características del dataset.

- **Pruebas con Datos en Diferentes Formatos:**

Se experimentó con el modelo usando datos en su forma original, datos normalizados y datos transformados mediante Kernel PCA. Aunque inicialmente los datos normalizados mostraron una mejor separabilidad sin PCA, el modelo basado en estos datos produjo métricas de agrupamiento deficientes.

- **Optimización de la Reducción de Dimensionalidad:**

Ante los retos encontrados con los datos normalizados, se retomó la reducción de dimensionalidad, iterando sobre diversas configuraciones de Kernel, valores de gamma y otros parámetros relevantes del PCA hasta encontrar la configuración óptima que mejoró significativamente la separabilidad y las métricas de agrupamiento.

- **Corrección del Desbalance en los Datos:**

Se detectó un desbalance significativo en los datos que afectaba el rendimiento del modelo. Se exploraron diferentes técnicas de balanceo, ajustando el modelo hasta encontrar la estrategia que produjera los mejores resultados, equilibrando las clases de manera efectiva.

- **Reentrenamiento y Evaluación Final del Modelo:**

Con los datos balanceados y la configuración óptima de reducción de dimensionalidad y método de agrupamiento, se reentrenó el modelo y se verificaron las métricas definidas logrando la obtención del modelo final.

5.4 Herramientas

- Python: Lenguaje de programación principal utilizado para el desarrollo del modelo.
- JupyterLab: Herramienta para el preprocesamiento de las playlist en archivos *JSON* de la data cruda
- Pandas y NumPy: Bibliotecas para manipulación y análisis de datos.
- Scikit-learn: Biblioteca para implementación de algoritmos de aprendizaje automático y validación del modelo.
- Spotify API: Fuente de datos para obtener características musicales detalladas.
- Spotipy: librería como puente entre la API de Spotify y Python

La tabla 8 ilustra herramientas adicionales implementadas

Librería/Módulo	Descripción
Matplotlib.pyplot	Visualización de gráficos
Seaborn	Visualización de gráficos
Standardscaler	Normalización de datos eliminando la media y escalando a la varianza unitaria.
Kernelpca	Implementación de PCA con kernels para diferentes enfoques
Kmeans	Agrupación de datos en K grupos basados en sus características.
Agglomerativeclustering	Realización de clustering jerárquico basado en la similitud de los datos.
Spectralclustering	Utilización de grafo Laplaciano para clustering no lineal.
Silhouette_score	Evaluación de cohesión y separación de clústers.
Davies_bouldin_score	Cálculo de la calidad de separación entre clústers.
Calinski_harabasz_score	Cálculo de la varianza entre clústers comparada con la varianza dentro de clústers.
Pairwise_distances	Cálculo de la distancia entre cada par de puntos en dos conjuntos.
Pairwise_distances_argmin_min	Identificación de puntos más cercanos entre dos conjuntos de datos.
Kfold	Generación de k conjuntos para validación cruzada en el modelado.
Localoutlierfactor (LOF)	Detección de anomalías evaluando la densidad local.
Joblib	Serialización eficiente de modelos y estructuras de datos grandes.
Sklearn.clúster	Contiene diferentes modelos de clustering
Sklearn.metrics	Ofrece métricas como silhouette_score, davies_bouldin_score y calinski_harabasz_score.
Sklearn.neighbors	Implementa funcionalidades de aprendizaje basado en vecinos más cercanos, incluyendo LOF.

Tabla 8. Descripción de Módulos Utilizados

6. Resultados y discusión

El enfoque iterativo del proyecto implicó la generación y ajuste constante de nuevos modelos para perfeccionar y alinear el sistema de recomendación musical. A través de la fase de lectura e interpretación de playlists de usuarios, el modelo se evaluó y balanceó continuamente, ajustando parámetros para optimizar su rendimiento.

Inicialmente, se aplicó el modelo a diferentes subconjuntos de canciones, generando las primeras recomendaciones que indicaron la necesidad de rebalanceo debido a sesgos en la distribución de los clústers, pues se evidenció que, aunque cambiaran los subconjuntos de datos, las recomendaciones siempre eran las mismas, y se identificó un desbalanceo importante en los clústers generados inicialmente como se muestra en la figura 10

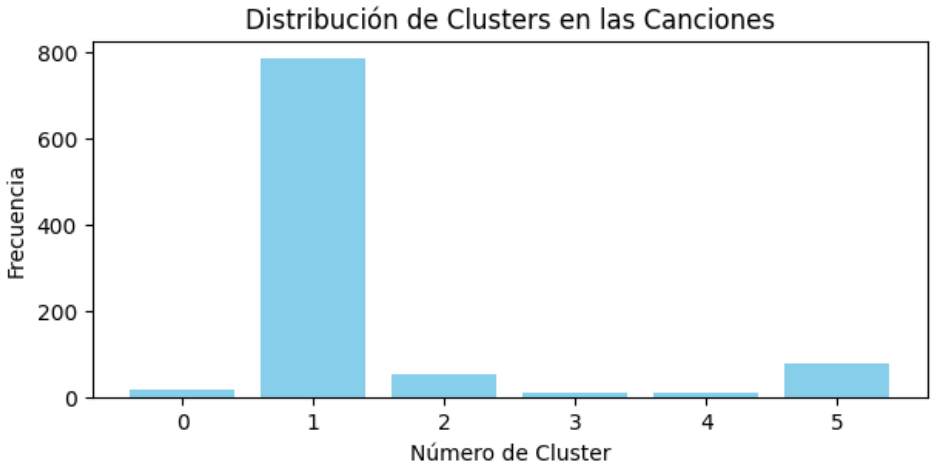


Figura 10. Distribución inicial de clústers

Para abordar el desbalanceo en los clústers, se implementaron varias técnicas, incluyendo el balanceo por promedio de muestras, mediana, factores mínimos y máximos, y ajustes de pesos según varianza normal y logarítmica. Estos métodos se probaron mediante *resample* de la biblioteca `sklearn.util`,

Finalmente, se seleccionó el método de balanceo por promedio de muestras, pues fue el que arrojó los mejores resultados. Posteriormente, se reentrenó el modelo y se dio lugar a la generación de recomendaciones.

La figura 11 ilustra una iteración donde se procesaron 35 pistas y se generaron las respectivas recomendaciones musicales

```
Ingrese el número de canciones a procesar: 35
Subconjunto de canciones procesadas:
                                          track_name         artist_name
394                                Best Day Of My Life    American Authors
964                                       Name For You           The Shins
589                                     Straightjacket          Quinn XCII
612                              Dreams and Nightmares           Meek Mill
768                                         Let Her Go           Passenger
153                                      Hold Me Tight                 BTS
251                                    Eight Miles High           Hüsker Dü
192                                         Talk To Me             Kopecky
99                                     GOOD (feat. ELO)                Loco
330                            Evacuate The Dancefloor             Cascada
647                                         Redeemed          DJ Shadow
77                                      Gimme Shelter  The Rolling Stones
822                                         Inside Out               Eve 6
1059                      Rather Be (feat. Jess Glynne)        Clean Bandit
656                             Change Is Gonna Come       Pretty Lights
322                                             Yeah!              Usher
32                                 Party In The U.S.A.        Miley Cyrus
995                                      In Cold Blood              alt-J
934                                   Mexican Jackpot           Flagship
741                 Building Steam With A Grain Of Salt          DJ Shadow
771                                 Hold Back The River          James Bay
136                                             FXXK IT             BIGBANG
815                                          Sugarhigh      Coyote Shivers
560                                             Don't       Bryson Tiller
643   Voodoo Child (Starring Afu Ra) [Dj Premier Remix]             DJ Cam
25                                  Yo (Excuse Me Miss)         Chris Brown
438                                         California                Niia
156                                       Anck Su Namum                YEZI
527                                         Highwayman       Willie Nelson
80                                  Foreplay / Long Time              Boston
613                       Real Hitta (feat. Kodak Black)               Plies
121                                       Just One Day                 BTS
672                                       Wicked Games          The Weeknd
1019                                         All of Me         John Legend
276                                          Chemistry          Jawbreaker
Recomendaciones de canciones:
              track_name artist_name  cluster
0          Elastic Heart         Sia        0
1             Can't Stop      CNBLUE        1
2                 Wasted      Tiësto        2
3  Bitter Sweet Symphony   The Verve        4
4            In Too Deep      Sum 41        5
Distancia promedio entre canciones recomendadas y procesadas: 0.27
```

Figura 11.Canciones Procesadas vs. Canciones Recomendadas

Puede observarse coherencia entre las canciones recomendadas y una medida baja de distancia entre las canciones recomendadas y las procesadas, permitiendo inferir que el modelo tiene un buen comportamiento para generar recomendaciones musicales que tienen gran afinidad con los gustos de los usuarios.

"Elastic Heart" de Sia (Clúster 0) coincide bien con su descripción de alta energía y bailabilidad moderada. De igual modo, "Can't Stop" de CNBLUE (Clúster 1) coincide con un clúster de alta bailabilidad y energía moderada.

Cabe resaltar que no hubo recomendaciones para el Clúster 3 (Alta Intensidad), lo cual es coherente dado que puede observarse que las canciones procesadas en la playlist del usuario no incluyen géneros de alta intensidad, como el metal pesado. Esto refleja una adecuada alineación del modelo con la naturaleza de los datos procesados, evitando forzar recomendaciones de un clúster que no corresponde con los patrones de escucha del usuario. De este modo, las recomendaciones observadas reflejan una correspondencia generalmente adecuada con los atributos de los clústers, mostrando que el modelo está bien sintonizado con las características de las canciones.

Sin embargo, es fundamental verificar el equilibrio del dataset para evitar sesgos significativos en las recomendaciones. Esto puede resultar en una limitante, especialmente si una persona tiene gustos musicales muy diversos o su playlist incluye géneros con características intensas, como el metal, ya que el modelo podría favorecer erróneamente ciertos géneros, lo que resultaría en recomendaciones que no reflejan fielmente las preferencias del usuario.

Adicionalmente, debe tenerse en cuenta que la base de datos del desafío de Spotify, está compuesta principalmente por playlists en inglés de EE. UU., lo cual restringe la eficacia del modelo al procesar géneros musicales o playlists en otros idiomas, como el español. Esto limita su aplicabilidad a un contexto global a menos que se expanda la base de datos para abarcar una mayor diversidad lingüística y cultural.

6.1.Métricas y Evaluación Cualitativa

- **Métricas de Agrupamiento**

Métrica	Valor
Silhouette ↑	0.555451
Davies-Bouldin ↓	0.910900
Calinski-Harabasz ↑	220.947.019
Cluster Coherence ↓	0.008264

Tabla 9. Resultados de Métricas de Agrupamiento

El modelo presenta un Silhouette Score de 0.555451, lo cual indica una separación y definición adecuada de los clústers, aunque hay espacio para mejorar hacia una diferenciación más

clara. Por su parte, el Davies-Bouldin Score de 0.910900 refleja positivamente en el modelo, ya que un valor bajo en esta métrica sugiere que los clústers no sólo son compactos sino también bien separados entre sí. Respecto al Calinski-Harabasz Score se tiene un valor alto, el cual confirma que los clústers están bien diferenciados, con una dispersión interna baja y una dispersión alta entre clústers vecinos, lo que indica que los grupos son densos y claramente distintos entre sí.

El valor bajo en "Cluster Coherence" es indicativo de que los clústers generados por el modelo son internamente consistentes y los puntos dentro de cada clúster son similares, lo que indica un signo adicional de un buen agrupamiento.

- **Métricas de Recomendación**

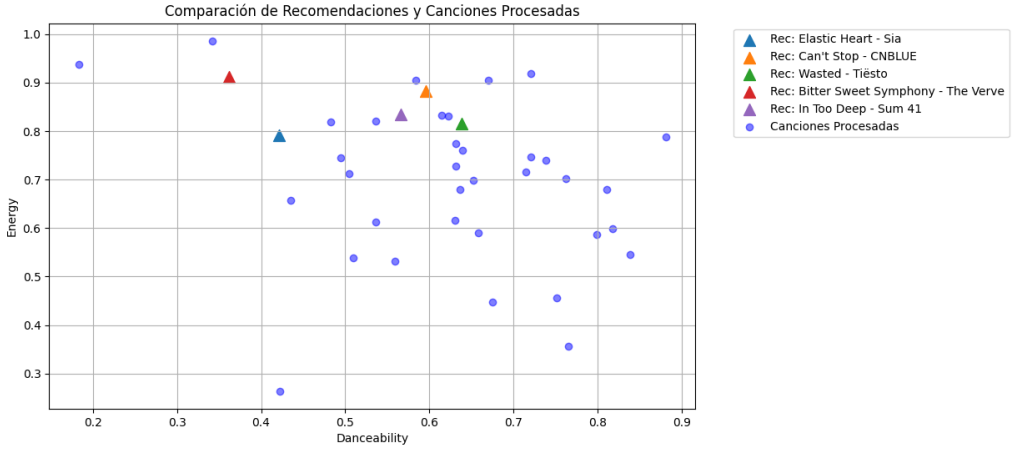


Figura 12. Comparación visual entre canciones recomendadas y cancions procesadas

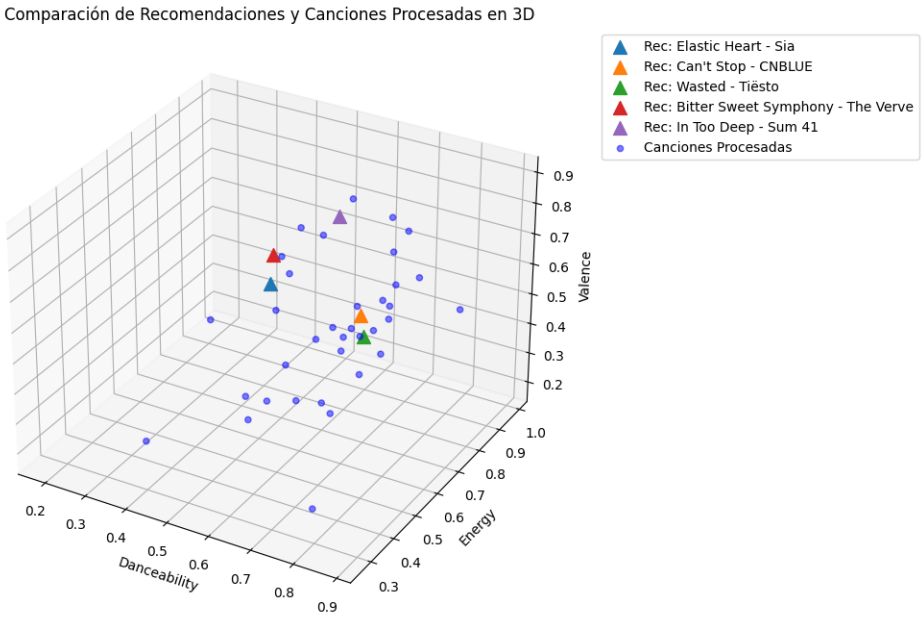


Figura 13. Comparación visual entre canciones recomendadas y canciones procesadas en 3D

Las figuras anteriores, muestran cómo las recomendaciones de canciones están distribuidas en relación con el conjunto general de canciones procesadas según tres características: bailabilidad, energía y valencia. Las recomendaciones están bien dispersas, cubriendo una amplia gama del espacio de características, lo que sugiere que el sistema de recomendación selecciona canciones que reflejan la diversidad presente en las canciones procesadas y que se alinean bien con las tendencias centrales de las características observadas en el conjunto más amplio, indicando que el modelo está funcionando eficazmente al elegir canciones representativas de las características musicales predominantes, demostrando un buen ajuste del sistema de recomendación a las características musicales del conjunto de datos.

Esto se ve respaldado por las métricas de precisión evaluadas. Partiendo del Mean Squared Error (MSE) entre las canciones recomendadas y procesadas con un valor de 0.08, sugiriendo que, en promedio, las diferencias entre las canciones recomendadas y las canciones escuchadas son pequeñas

```
Distancia promedio entre canciones recomendadas y procesadas: 0.27
Mean Squared Error (MSE) entre las canciones recomendadas y procesadas: 0.0845
```

Figura 14. Resultados para métricas de recomendación

Adicionalmente, la distancia promedio entre canciones recomendadas y procesadas es de 0.27, lo cual representa un valor adecuado y valida la precisión del modelo de recomendación musical.

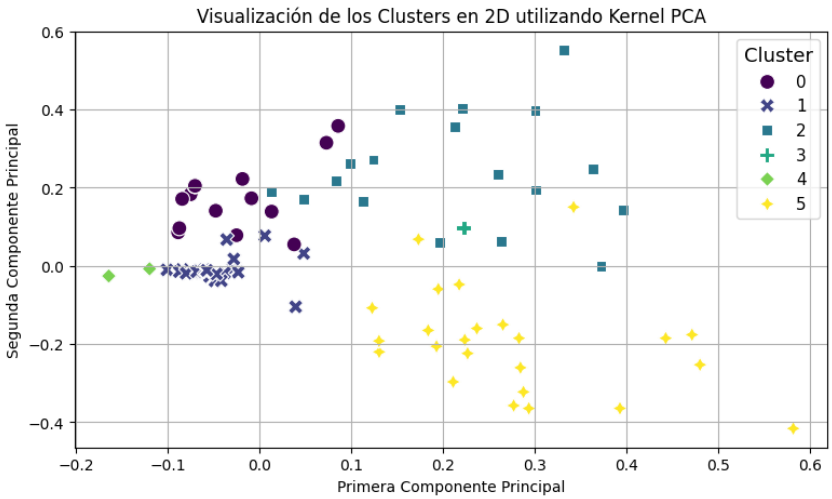


Figura 15. Visualización de clústers para el modelo elegido

Por otro lado, de acuerdo con la figura anterior, puede observarse buena diferenciación y separación entre los grupos, indicando que el modelo de clustering ha sido eficaz en agrupar datos similares. Los clústers 0 y 2 se destacan por su alta cohesión, mientras que los clústers 1 y 5, más dispersos y con cierto solapamiento, podrían requerir ajustes para mejorar su precisión.

El clúster 3 (Alta intensidad) tiene una representación muy pequeña, lo que sugiere evaluar la inclusión de nuevos datos correspondientes a música con características intensas para robustecer el dataset y lograr ampliar el espectro de géneros y niveles de intensidad considerados. Dado esto, se recomienda continuar optimizando el modelo para perfeccionar aún más la separabilidad, especialmente en las áreas donde los clústers se solapan o dispersan.

6.2. Consideraciones de producción

El modelo desarrollado es de naturaleza académica y no está diseñado para ser desplegado en un entorno de producción real sin ajustes adicionales. Actualmente, sirve como una herramienta educativa y experimental para explorar y entender las técnicas de clustering y recomendación musical. Sin embargo, si se considera la posibilidad de adaptar y escalar este modelo para un entorno de producción, sería esencial implementar varias mejoras. Entre éstas se incluyen el establecimiento de sistemas robustos para el monitoreo continuo del rendimiento del modelo, la integración con flujos de datos en tiempo real para actualizaciones dinámicas de las recomendaciones, y la consideración de aspectos de seguridad y privacidad al almacenar y procesar datos en servicios en la nube

7. Conclusiones

- El modelo de clustering implementado ha demostrado ser efectivo en agrupar canciones basándose en sus características musicales. Las visualizaciones de recomendaciones vs canciones procesadas en 2D y 3D han demostrado que los clústers formados son coherentes y distinguen claramente entre diferentes tipos de música, generando recomendaciones musicales que presentan buena alineación con las preferencias del usuario, lo que sugiere que el sistema puede hacer recomendaciones relevantes y personalizadas.

- El análisis de las métricas, como el Mean Squared Error (MSE), ha proporcionado evidencia de que las canciones recomendadas poseen similitudes significativas con las canciones del conjunto de datos, lo que refleja una precisión adecuada del modelo logrando alcanzar los objetivos planteados inicialmente en cuanto a desarrollar un algoritmo capaz de hacer recomendaciones personalizadas de música.

- Aunque el modelo ha funcionado bien en un entorno controlado, es importante reconocer sus limitaciones, como la necesidad de más diversidad en el dataset y la adaptación para manejar diferentes géneros y preferencias más amplias. La falta de recomendaciones en ciertos clústers sugiere que el modelo podría mejorarse para cubrir más completamente el espectro musical.

- En retrospectiva, el proyecto ha sido exitoso en demostrar cómo las técnicas de aprendizaje automático y análisis de datos pueden ser aplicadas efectivamente en el campo de la recomendación musical. Sin embargo, para avanzar hacia una aplicación práctica, sería esencial realizar ampliaciones del modelo para asegurar que las recomendaciones son culturalmente diversificadas.

8. Recomendaciones

Para futuras investigaciones y mejoras en el campo de la recomendación musical utilizando técnicas de clustering, se recomienda considerar las siguientes líneas de acción:

- **Reentrenamiento con Nuevos Datos**

  Periódicamente reentrenar el modelo con nuevos "slices" del dataset original de Spotify puede ayudar a mejorar la precisión y la relevancia de las recomendaciones. Esto permitirá que el modelo se adapte a nuevas tendencias musicales y cambios en las preferencias de los usuarios, manteniéndolo actualizado y eficaz.

- **Expansión de la Diversidad Musical**

  Es esencial ampliar el espectro de géneros musicales en el dataset. Incluir más playlists que no solo provengan de EE. UU., sino también de diferentes regiones del mundo, puede enriquecer el modelo, permitiendo que capture una mayor diversidad de gustos musicales y culturales. Esto podría mejorar la universalidad y la aceptación del sistema de recomendación en un contexto global.

- **Mejora de la Separabilidad de los Clústers**

  Continuar trabajando en la mejora de la separabilidad de los clústers es crucial. Se podría experimentar con diferentes algoritmos de clustering o ajustar los parámetros de los métodos existentes para aumentar la distinción entre los clústers. Esto mejorará la claridad y la utilidad de las categorizaciones del modelo, resultando en recomendaciones más precisas y coherentes.

- **Implementación de Métodos Adicionales de Reducción de Dimensionalidad**

  Explorar otros métodos de reducción de dimensionalidad además de PCA, como t-SNE o UMAP, podría proporcionar nuevas perspectivas sobre cómo las características de las canciones se agrupan y diferencian. Estas técnicas pueden revelar estructuras de datos más complejas y mejorar la formación de clústers.

Referencias

[1] S. Antenucci, S. Boglio, E. Chioso, E. Dervishaj, S. Kang, T. Scarlatti, and M. F. Dacrema. Artist-driven layering and user's behaviour impact on recommendations in a playlist continuation scenario. In Proceedings of the 2018 ACM Recommender Systems Challenge, RecSysChallenge '18, Vancouver, BC, Canada, 2018

[2] Arango Archila, F. (2016). El impacto de la tecnología digital en la industria discográfica. Dixit, 24(1), ISSN 1688-3497 (versión impresa) / ISSN 0797-3691 (versión en línea). Pontificia Universidad Javeriana, Bogotá, Colombia.

[3] CW Chen, P. Lamere, M. Schedl y H. Zamani. Recsys Challenge 2018: Continuación automática de listas de reproducción de música. En actas de la 12.ª Conferencia ACM sobre sistemas de recomendación (RecSys '18), 2018

[4] Spotify. (2022). *Spotify Playlist Million Dataset Challenge*. AIcrowd. https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge

[5] Spotify. (2023). *Get Playlist*: Consultar y descargar datos para el proyecto. Recuperado en marzo de 2023, consultado hasta junio de 2023, de https://developer.spotify.com/documentation/web-api/reference/get-playlist.

[6] Unisaacarroyov. (2022, mayo 24). Spotipy: Extraer características de canciones - tacosdedatos. tacosdedatos. Recuperado el 12 de junio de 2024, de https://www.tacosdedatos.com/unisaacarroyov/spotipy-extraer-caracteristicas-de-canciones-9km