

# Proyecto de organizar música en Spotify

**Por: Juan Fernando Soto**

El objetivo del proyecto es un programa que clasifique música de Spotify por sus características musicales, con el objetivo de agruparlas por semejanzas e incluso por géneros, para ello se utilizarán los mismos datos de Spotify y sus características:

- **Danceability:** Que tan cercana a ritmos bailables está la canción.
- **Energy:** Que tanta energía percibida transmite la canción. Una canción de metal transmite más energía que una obra de Bach.
- **Liveness:** Probabilidad de que exista una versión en vivo de la obra musical.
- **Loudness:** Los dB promedio de la canción.
- **Speechness:** Que tanta presencia hay de palabras habladas hay en una pista de Spotify. Mayor a 0.6 tiene más probabilidad de ser totalmente hablado, menor a 0.3 mayor probabilidad de no tener palabras.
- **Valence:** Que tipo de sentimiento transmite, más cercano a 1, transmite sentimientos positivos, más cercano a 0, transmite sentimientos negativos.
- **Acousticness:** Un valor más cercano a 1 tiene más componentes rudimentarios, mientras que valores cercanos a 0 tienen mayores modificaciones con medios electrónicos (guitarras eléctricas, sintetizadores, amplificadores).

El proyecto consiste en tres partes, la primera es la extracción de datos de Spotify, que son las características de las canciones por playlist, tales como cuanta cantidad de palabras tienen o su volumen promedio. El segundo es la selección del mejor modelo, en donde con ayuda de la función GridSearchCV, no solo para revisar entre los mejores parámetros sino también entre el mejor método. Y finalmente la selección de las playlists a las que pertenece o es más afín.

## Extracción de datos

En esta fase se utilizó tanto la documentación de Spotify (<https://developer.spotify.com/>), en donde fue necesario crear una aplicación a partir de la página, este nos daba una información para poder acceder a la aplicación y por lo tanto a los datos de Spotify. Esto con ayuda de la librería Spotipy.



Esta es la visualización de la aplicación y los datos para acceder a ella.

A partir de aquí se sacan las playlists del usuario, se hace una lista para filtrar las deseadas, y se empieza a examinar cada lista una por una extrayendo cada canción y dejando el nombre de la playlist como la etiqueta, después de generar este dataframe, se sacan todas las canciones del usuario, así como los nombres, el artista y el id, y así poder organizarlas.

## Mejor modelo y afinidad

Para la revisión del mejor modelo, se hizo el típico desarrollo de datos, en utilizando `train_test_split`, y un escalamiento, Después empleando una lista, en donde la primera columna son los modelos, la segunda los valores a iterar y la tercera el nombre del modelo, se va buscando cual es el mejor modelo para los datos, y una vez encontrado, se hace otro entrenamiento, con los todos los datos, y se vuelve a evaluar.

Finalmente, para la afinidad se desarrolló una técnica, para poder encontrar las listas con mayor afinidad por medio de la predicción de probabilidad por etiquetas, la idea en este, es que la suma de las mayores probabilidades, de lo justo para pasar un umbral, por ejemplo si el umbral es del

70% y una lista es del 80% solo se guarde esa lista, pero si las mayores probabilidades son del 35%, 34% y 20%, entonces se guardan esas tres playlists.

Además de la implementación de este sistema, también se evalúa si dentro de las listas sacadas está la que es correcta utilizando el `f1_score`. Finalmente se entrega un dataframe, que tiene las tres primeras columnas los datos de identificación de las canciones, las siguientes columnas las playlists afines, y después las probabilidades de esas playlists, claramente va a haber el numero de columnas acorde a la canción que más afín sea a varias canciones, y algunos datos quedarán vacíos.

## Resultados

Al final tanto las múltiples evaluaciones como el dataframe generado dan los resultados diciendo que el modelo es deficiente separando datos, el promedio de los mejores datos de cross validación fueron..., el mejor modelo fue..., los mejores parámetros..., el `f1_score` implementado para las playlists afines fue..., incluso su reporte de clasificación fue:

Sin embargo los datos (Se adjuntan también), dan como resultado los 5 primeros datos y los últimos 5, la misma disposición de etiquetas:

## Conclusiones

El resultado no fue el esperado, pues soltaba que todas (O casi todas) las canciones eran afines a la misma combinación de listas, esto creería que es más porque las características de las canciones algunas al ser tan subjetivas, y sin conocer de donde o como se calculan esos datos, puede que no sean datos verdaderamente diferenciadores, y a los ojos del modelo pues se está agrupando casi que la misma cosa, posiblemente definiendo muy bien solo algunas de las características o tener en cuenta otros aspectos que se puede pedir a Spotify (Como la popularidad, o los mercados en donde es más comercial esa canción u otros datos, puede que sí salga mejor, sin embargo con estos no sale muy bien.

Además, después de todo el tratamiento, y el trabajo dado, fue muy pulido, y salió muy bien, pues desde las iteraciones extrayendo los datos, hasta la organización de la afinidad se dio como se quería y esperaba.

## Links de YouTube y GitHub

<https://youtu.be/Q8HeID-icyw>

<https://github.com/Jufersoto/Organize-music>

Se cambiaron los clientes de la app de Spotify por temas de privacidad...