

Workshop 2

Juan Carlos Quintero Esquivel (2225339)

Javier Alejandro Vergara Zorrilla

Universidad autónoma de occidente

**Santiago de Cali
2024**

Contexto

El WorkShop2 se centra en el análisis de dos conjuntos de datos en formato CSV. El primer conjunto contiene información detallada sobre canciones de 125 géneros musicales diferentes, con atributos que van desde el artista y álbum hasta características específicas de las canciones como popularidad, duración y elementos acústicos. A continuación, se detallan las columnas más importantes de este dataset:

track_id: Identificador único asignado por Spotify a cada canción.

artists: Listado de los artistas de la canción, separados por punto y coma si son varios.

album_name: Nombre del álbum en el que se incluye la canción.

track_name: Título de la canción.

popularity: Escala de popularidad de 0 a 100, donde 100 es la más popular.

duration_ms: Duración de la canción en milisegundos.

explicit: Indica si la canción contiene contenido explícito (True para sí, False para no).

danceability: Valor entre 0.0 y 1.0 que mide cuán bailable es la canción.

energy: Mide la intensidad de la canción en una escala de 0.0 a 1.0.

key: Tono musical de la canción (por ejemplo, C, D, E).

loudness: Nivel de volumen general de la canción en decibelios.

mode: Indica si la canción está en modo mayor (1) o menor (0).

speechiness: Proporción de palabras habladas en la canción, con valores cercanos a 1.0 indicando más palabras.

acousticness: Mide cuán acústica es la canción en una escala de 0.0 a 1.0.

instrumentalness: Predice si la canción es instrumental (valores cercanos a 1.0 indican mayor probabilidad).

liveness: Indica la probabilidad de que la canción se haya grabado en vivo (valores superiores a 0.8 sugieren grabaciones en vivo).

valence: Mide el nivel de positividad de la canción, de 0.0 (triste) a 1.0 (feliz).

tempo: Velocidad de la canción en beats por minuto (BPM).

time_signature: Indica el compás de la canción.

track_genre: Género musical de la canción.

El segundo dataset contiene información histórica sobre los premios Grammy desde 1959 hasta 2019, con los siguientes campos principales:

year: Año en que se celebró la ceremonia de los Grammy.

title: Título del evento, que incluye la edición y el año.

published_at: Fecha y hora en que se publicó la información.

updated_at: Última fecha de actualización de la información.

category: Categoría del premio (ejemplo: "Grabación del año").

nominee: Canción o proyecto nominado.

artist: Artista nominado.

workers: Lista de las personas involucradas en el proyecto (productores, ingenieros, etc.).

img: URL de la imagen relacionada con la nominación.

winner: Indica si el nominado ganó en su categoría (True para sí, False para no).

Proceso de análisis

Se lleva a cabo un análisis exploratorio de datos (EDA) en ambos conjuntos de datos para evaluar su estructura, realizar transformaciones necesarias y limpiar la información. El objetivo es preparar los datos para su análisis y posterior visualización.

Herramientas utilizadas:

Python: Se utilizó para crear los scripts necesarios para la carga de datos en la base de datos y la ejecución de tareas en Airflow.

Jupyter Notebooks: Los análisis EDA se realizaron en este entorno, donde se limpiaron y transformaron los datos, y se generaron gráficos que facilitaron la comprensión de la información.

Ubuntu: Se utilizó una máquina virtual con Ubuntu para ejecutar el proyecto en un entorno Linux, que es requerido por Airflow.

Apache Airflow: Orquestador de flujos de trabajo utilizado para automatizar la extracción, transformación y carga (ETL) de los datos.

venv: Gestor de dependencias y entornos virtuales de Python para asegurar que todas las librerías necesarias estuvieran disponibles.

Git y GitHub: Utilizados para el control de versiones del código y la colaboración en el proyecto.

PowerBI: Herramienta utilizada para crear las visualizaciones más complejas y compartir los resultados del análisis.

SQLAlchemy: Librería de Python que facilitó la conexión a la base de datos y la actualización de datos post-EDA.

Pandas: Se utilizó para manipular y analizar los datos, así como para realizar transformaciones y limpiezas.

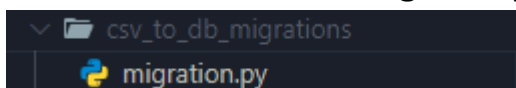
Dotenv: Librería empleada para manejar las credenciales de la base de datos de forma segura, evitando que se expongan en el código.

PostgreSQL: Base de datos relacional utilizada para almacenar y gestionar los datos después de su transformación.

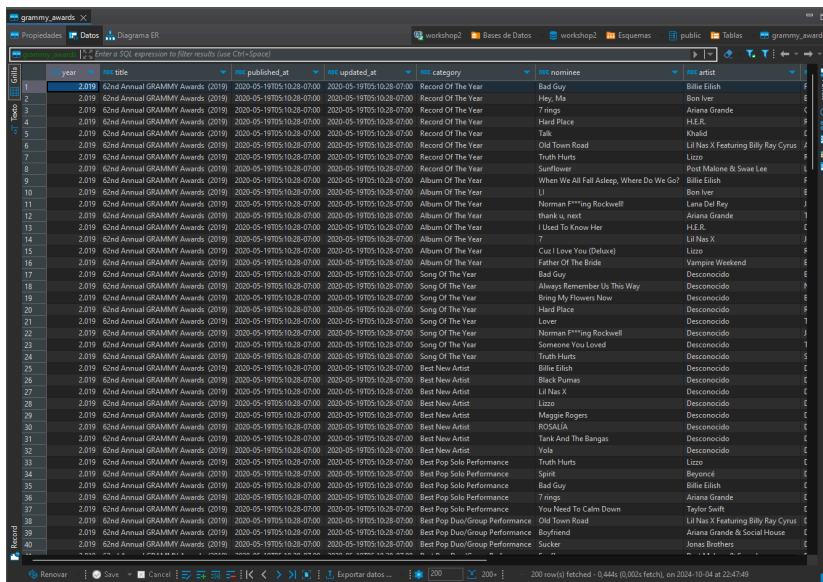
Ngrok: Se empleó para crear un túnel que expusiera la base de datos local a la máquina Ubuntu para que pudiera ser accesible desde el exterior.

Este flujo de trabajo asegura la correcta transformación, análisis y visualización de los datos, facilitando la obtención de insights valiosos tanto del contenido musical como de la historia de los premios Grammy.

Subida de el dataset de grammy a la base de datos:



Se usó un script de python para subir el csv a la base de datos.



	year	title	published_at	updated_at	category	nominee	artist
1	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	Bad Guy	Billie Eilish
2	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	My, My	Jonas Brothers
3	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	7 rings	Ariana Grande
4	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	Hard Place	H.E.R.
5	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	Talk	Khalid
6	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	Old Town Road	Lil Nas X Featuring Billy Ray Cyrus
7	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	Truth Hurts	Lizzo
8	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	Surfswear	Post Malone & Swae Lee
9	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Album Of The Year	When We All Fall Asleep, Where Do We Go?	Billie Eilish
10	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Album Of The Year	Li	Jonas Brothers
11	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Album Of The Year	Norman F***ing Rockwell	Lana Del Rey
12	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Album Of The Year	thank u, next	Ariana Grande
13	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Album Of The Year	I Used To Know Her	H.E.R.
14	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Album Of The Year	7	Lil Nas X
15	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Album Of The Year	Cuz I Love You (Deluxe)	Lizzo
16	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Album Of The Year	Father Of The Bride	Vampire Weekend
17	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Song Of The Year	Bad Guy	Desconocido
18	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Song Of The Year	Always Remember Us This Way	Desconocido
19	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Song Of The Year	Bring My Flowers Now	Desconocido
20	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Song Of The Year	Hard Place	Desconocido
21	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Song Of The Year	Love	Desconocido
22	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Song Of The Year	Norman F***ing Rockwell	Desconocido
23	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Song Of The Year	Someone You Loved	Desconocido
24	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Song Of The Year	Truth Hurts	Desconocido
25	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Best New Artist	Billie Eilish	Desconocido
26	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Best New Artist	Black Pumas	Desconocido
27	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Best New Artist	Lil Nas X	Desconocido
28	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Best New Artist	Lizzo	Desconocido
29	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Best New Artist	Maggie Rogers	Desconocido
30	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Best New Artist	ROSALÍA	Desconocido
31	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Best New Artist	Tank And The Bangas	Desconocido
32	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Best New Artist	Yola	Desconocido
33	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Best Pop Solo Performance	Truth Hurts	Lizzo
34	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Best Pop Solo Performance	Spirit	Bejovinc
35	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Best Pop Solo Performance	Bad Guy	Billie Eilish
36	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Best Pop Solo Performance	7 rings	Ariana Grande
37	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Best Pop Solo Performance	You Need To Calm Down	Taylor Swift
38	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Best Pop Duo/Group Performance	Old Town Road	Lil Nas X Featuring Billy Ray Cyrus
39	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Best Pop Duo/Group Performance	Boyz n the Hood	Ariana Grande & Social House
40	2019	62nd Annual GRAMMY Awards	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Best Pop Duo/Group Performance	Sucker	Jonas Brothers

EDAs:

EDA_grammy:

En este Notebook se hará todo el proceso de transformación, limpieza y carga de datos del dataset de grammy.

```
from sqlalchemy import create_engine
import pandas as pd
import os
from dotenv import load_dotenv
from sqlalchemy import text
```

Primero se importan todas las librerías necesarias para hacer el EDA y conectar con la base de datos.

```
load_dotenv()

db_connection_url = f"postgresql://{os.getenv('DB_USERNAME')}:{os.getenv('DB_PASSWORD')}@localhost:5432/grammy"
engine = create_engine(db_connection_url)

# using SQLAlchemy to read the data directly from the database
with engine.connect() as connection:
    result = connection.execute(text("SELECT * FROM grammy_awards"))
    grammy_df = pd.DataFrame(result.fetchall(), columns=result.keys())

grammy_df_head = grammy_df.head()
grammy_df_info = grammy_df.info()

(grammy_df_head, grammy_df_info)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4810 entries, 0 to 4809
Data columns (total 10 columns):
#   Column          Non-Null Count  Dtype
---  -
0   year            4810 non-null  int64
1   title           4810 non-null  object
2   published_at    4810 non-null  object
3   updated_at     4810 non-null  object
4   category        4810 non-null  object
5   nominee        4810 non-null  object
6   artist         4810 non-null  object
7   workers        4810 non-null  object
8   img            4810 non-null  object
9   winner         4810 non-null  bool
dtypes: bool(1), int64(1), object(8)
memory usage: 343.0+ KB
Out[5]:
```

```

dtypes: bool(1), int64(1), object(8)
memory usage: 343.0+ KB
Out[5]:
(   year      title      published_at \
0  2019  62nd Annual GRAMMY Awards (2019) 2020-05-19T05:10:28-07:00
1  2019  62nd Annual GRAMMY Awards (2019) 2020-05-19T05:10:28-07:00
2  2019  62nd Annual GRAMMY Awards (2019) 2020-05-19T05:10:28-07:00
3  2019  62nd Annual GRAMMY Awards (2019) 2020-05-19T05:10:28-07:00
4  2019  62nd Annual GRAMMY Awards (2019) 2020-05-19T05:10:28-07:00

      updated_at      category      nominee      artist \
0  2020-05-19T05:10:28-07:00  Record Of The Year  Bad Guy  Billie Eilish
1  2020-05-19T05:10:28-07:00  Record Of The Year  Hey, Ma  Bon Iver
2  2020-05-19T05:10:28-07:00  Record Of The Year  7 rings  Ariana Grande
3  2020-05-19T05:10:28-07:00  Record Of The Year  Hard Place  H.E.R.
4  2020-05-19T05:10:28-07:00  Record Of The Year  Talk  Khalid

      workers \
0  Finneas O'Connell, producer; Rob Kineliski & Fi...
1  BJ Burton, Brad Cook, Chris Messina & Justin V...
2  Charles Anderson, Tommy Brown, Michael Foster ...
3  Rodney "Darkchild" Jerkins, producer; Joseph H...
4  Disclosure & Denis Kosiak, producers; Ingmar C...

```

Columna	Descripción
year	Año en que se otorgó el premio.
title	Título de la ceremonia de premiación.
published_at	Fecha de publicación de los resultados.
updated_at	Fecha de la última actualización de los datos.
category	Categoría del premio.
nominee	Nombre de la obra o persona nominada.
artist	Artista o grupo que recibió la nominación.
workers	Personas que contribuyeron a la obra nominada.
img	Enlace a la imagen relacionada con la nominación.
winner	Indica si la nominación resultó ganadora (True = ganador, False = no ganador).

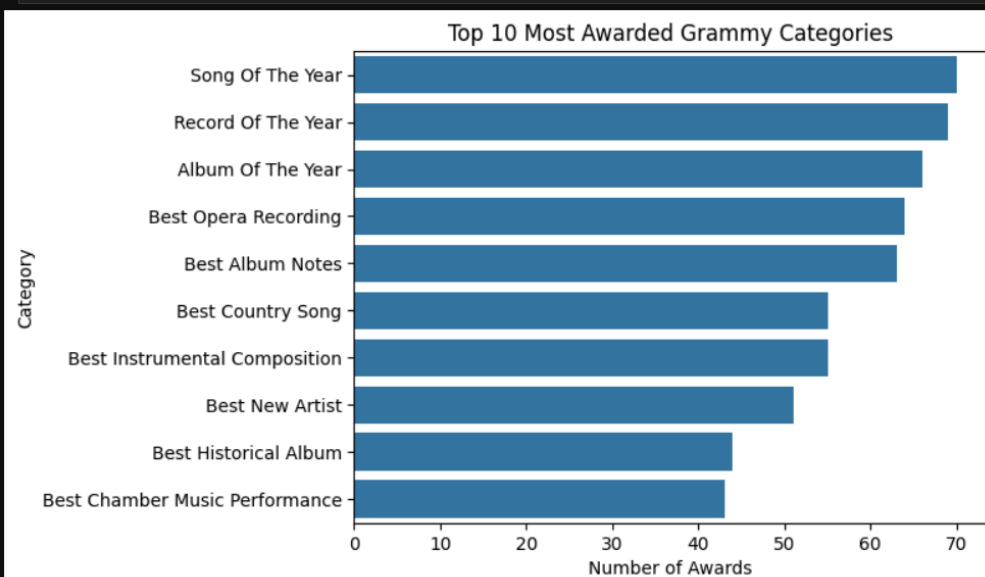
Se trae la base de datos y observamos rápidamente los datos que contiene.

```
In [6]:
```

```
category_counts = grammy_df['category'].value_counts().head(10)  
print(category_counts)
```

```
category  
Song Of The Year          70  
Record Of The Year        69  
Album Of The Year         66  
Best Opera Recording       64  
Best Album Notes          63  
Best Country Song         55  
Best Instrumental Composition 55  
Best New Artist           51  
Best Historical Album      44  
Best Chamber Music Performance 43  
Name: count, dtype: int64
```

```
import matplotlib.pyplot as plt  
import seaborn as sns  
  
sns.barplot(x=category_counts.values, y=category_counts.index)  
plt.title('Top 10 Most Awarded Grammy Categories')  
plt.xlabel('Number of Awards')  
plt.ylabel('Category')  
plt.show()
```



En estas visualizaciones se muestran las categorías de los premios Grammy que han tenido más galardones a lo largo de los años. Primero, se contó la frecuencia de las categorías usando `value_counts()` y se graficaron las 10 categorías con más premios mediante un gráfico de barras. Esto ayuda a identificar las categorías más recurrentes, como "Song Of The Year" y "Record Of The Year".

```

3]: # Count awards per year to identify any unusual spikes
awards_per_year = grammy_df['year'].value_counts()
print(awards_per_year)

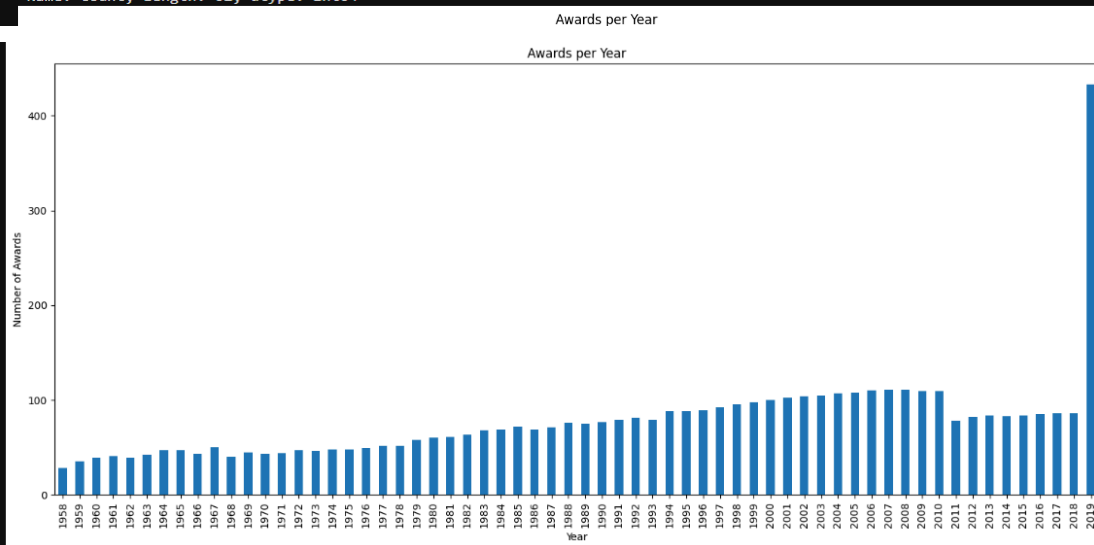
awards_per_year.sort_index().plot(kind='bar', figsize=(15, 7)) # Ajusta el tamaño de la figura
plt.title('Awards per Year')
plt.xlabel('Year')
plt.ylabel('Number of Awards')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()

```

```

year
2019    433
2007    111
2008    111
2006    110
2010    109
...
1968     40
1962     39
1960     39
1959     35
1958     28
Name: count, Length: 62, dtype: int64

```



The majority of the data (awards per year), comprising 433 entries, is from 2019.

Aquí se analiza la distribución de premios por año. Usando un gráfico de barras, se puede observar un aumento significativo en la cantidad de premios entregados en 2019. El conteo se realiza sobre la columna year y se grafica el número de premios entregados por año, revelando un crecimiento a lo largo del tiempo, con algunos picos importantes.


```
In [9]: top_nominees = grammy_df['nominee'].value_counts().head(10)
top_artists = grammy_df[grammy_df['winner'] == True]['artist'].value_counts().head(10)

print("Top Nominees:\n", top_nominees)
print("Top Winning Artists:\n", top_artists)
```

```
Top Nominees:
nominee
Berlioz: Requiem      7
Bridge Over Troubled Water  7
Steven Epstein        7
Robert Woods         7
Up, Up And Away      6
Britten: War Requiem  6
Desconocido          6
David Frost          6
A Taste Of Honey     6
Berlioz: Les Troyens  5
Name: count, dtype: int64
Top Winning Artists:
artist
Desconocido      1840
(Various Artists)  66
U2               18
Aretha Franklin  16
Ella Fitzgerald  13
Bruce Springsteen 13
Beyoncé         13
Stevie Wonder    13
Tony Bennett     12
Dixie Chicks     12
Name: count, dtype: int64
```

Este análisis presenta los nominados y artistas más premiados en los Grammy. Primero, se contó la cantidad de nominaciones por proyecto y, luego, los artistas ganadores más frecuentes. Entre los artistas más destacados se encuentran U2, Aretha Franklin y Beyoncé, mientras que categorías como "Berlioz: Requiem" y "Bridge Over Troubled Water" tienen varias nominaciones.

Cleaning

Duplicates

```
0]: # Remove duplicates
grammy_df = grammy_df.drop_duplicates()

# Confirm removal
print(f"Duplicates removed, new dataset size: {grammy_df.shape}")
```

Duplicates removed, new dataset size: (4810, 10)

Missing Values

```
1]: # Check for missing values
print(grammy_df.isnull().sum())

# For categorical data, you might use a placeholder like 'Unknown'
grammy_df.fillna('Unknown', inplace=True)
```

```
year      0
title     0
published_at  0
updated_at  0
category  0
nominee   0
artist    0
workers   0
img       0
winner    0
dtype: int64
```

En esta sección, se aplican técnicas de limpieza de datos. Primero, se eliminan las entradas duplicadas del DataFrame usando `drop_duplicates()`, lo que reduce el tamaño del dataset. Después, se realiza un chequeo de valores nulos y se reemplazan con 'Unknown' en las categorías donde se identificaron vacíos, garantizando la consistencia en los valores.

Fields Cleaning

```
2]: # Standardizing the 'artist' field
grammy_df['artist'] = grammy_df['artist'].str.title()

# Remove any leading/trailing whitespaces
grammy_df['artist'] = grammy_df['artist'].str.strip()

print(grammy_df['artist'].head())
```

```
0    Billie Eilish
1      Bon Iver
2    Ariana Grande
3       H.E.R.
4       Khalid
Name: artist, dtype: object
```

Analyze Award Year

```
3]: #Ensure 'year' is integer type
grammy_df['year'] = grammy_df['year'].astype(int)
valid_years = grammy_df['year'].between(1958, 2024)
grammy_df = grammy_df[valid_years]
```

Aquí se estandarizan los nombres de los artistas, transformando el texto a título (`title()`) y eliminando espacios en blanco innecesarios con `strip()`. Esto asegura que los nombres estén en un formato coherente para análisis futuros.

Además, se valida que la columna year contenga valores numéricos entre los años válidos de los premios Grammy, filtrando entradas no deseadas.

Inspect Categorical Values

```
In [14]: # Standardize category names to lower case and strip extra whitespace
grammy_df['category'] = grammy_df['category'].str.lower().str.strip()
print(grammy_df['category'].unique()) # Check for inconsistencies

['record of the year' 'album of the year' 'song of the year'
 'best new artist' 'best pop solo performance'
 'best pop duo/group performance' 'best traditional pop vocal album'
 'best pop vocal album' 'best dance recording'
 'best dance/electronic album' 'best contemporary instrumental album'
 'best rock performance' 'best metal performance' 'best rock song'
 'best rock album' 'best alternative music album' 'best new age album'
 'best r&b performance' 'best traditional r&b performance' 'best r&b song'
 'best urban contemporary album' 'best r&b album' 'best country song'
 'best rap performance' 'best rap/sung performance' 'best rap song'
 'best rap album' 'best pop gospel album' 'best country solo performance'
 'best country duo/group performance' 'best country album'
 'best improvised jazz solo' 'best jazz vocal album'
 'best jazz instrumental album' 'best large jazz ensemble album'
 'best latin jazz album' 'best gospel performance/song'
 'best contemporary christian music performance/song' 'best gospel album'
 'best contemporary christian music album' 'best roots gospel album'
 'best latin pop album' 'best latin rock, urban or alternative album'
 'best regional mexican music album (including tejano)'
 'best tropical latin album' 'best album notes'
 'best american roots performance' 'best american roots song'
 'best americana album' 'best bluegrass album'
 'best traditional blues album' 'best pop instrumental performance'
 'best contemporary blues album' 'best folk album'
 'best regional roots music album' 'best reggae album'
 'best world music album' 'best children's music album'
 'best spoken word album (includes poetry, audio books & storytelling)'
 'best comedy album' 'best musical theater album']
```

Para mantener consistencia en los valores categóricos, se convierten los nombres de las categorías a minúsculas y se eliminan los espacios en blanco adicionales. Esto ayuda a evitar discrepancias en los nombres de las categorías cuando se realicen análisis o agrupaciones más adelante.

Check Unusual Entries

```
[15]: # Check for short or null entries in 'nominee' and 'artist'
grammy_df['nominee'] = grammy_df['nominee'].fillna('Unknown')
grammy_df['artist'] = grammy_df['artist'].fillna('Unknown')
grammy_df['nominee_length'] = grammy_df['nominee'].str.len()
short_entries = grammy_df[grammy_df['nominee_length'] < 5] # Assuming 5 as a threshold
print(short_entries)
```

	year		title	published_at	\
4	2019	62nd Annual GRAMMY Awards	(2019)	2020-05-19T05:10:28-07:00	
9	2019	62nd Annual GRAMMY Awards	(2019)	2020-05-19T05:10:28-07:00	
13	2019	62nd Annual GRAMMY Awards	(2019)	2020-05-19T05:10:28-07:00	
31	2019	62nd Annual GRAMMY Awards	(2019)	2020-05-19T05:10:28-07:00	
43	2019	62nd Annual GRAMMY Awards	(2019)	2020-05-19T05:10:28-07:00	
...	
4078	1975	18th Annual GRAMMY Awards	(1975)	2017-11-28T00:03:45-08:00	
4388	1968	11th Annual GRAMMY Awards	(1968)	2017-11-28T00:03:45-08:00	
4461	1966	9th Annual GRAMMY Awards	(1966)	2017-11-28T00:03:45-08:00	
4599	1963	6th Annual GRAMMY Awards	(1963)	2017-11-28T00:03:45-08:00	
4801	1958	1st Annual GRAMMY Awards	(1958)	2017-11-28T00:03:45-08:00	
		updated_at	\		
4		2020-05-19T05:10:28-07:00			
9		2020-05-19T05:10:28-07:00			
13		2020-05-19T05:10:28-07:00			
31		2020-05-19T05:10:28-07:00			
43		2020-05-19T05:10:28-07:00			
...		...			
4078	2019-09-10T01:06:59-07:00				
4388	2019-09-10T01:11:09-07:00				
4461	2019-09-10T01:07:37-07:00				
4599	2019-09-10T01:11:09-07:00				
4801	2019-09-10T01:11:09-07:00				

En esta fase, se identifican entradas inusuales o sospechosamente cortas en las columnas nominee y artist. Se asigna 'Unknown' a los valores nulos, y se filtran aquellos con nombres de longitud inferior a cinco caracteres para evaluar si son válidos o requerirán corrección.

```
grammy_df.to_csv('../data/grammy_dataset_cleaned.csv', index=False)
```

En esta última etapa, el DataFrame resultante del proceso de limpieza y estandarización se exporta a un archivo CSV. La función `to_csv()` guarda el DataFrame como un archivo llamado `grammy_dataset_cleaned.csv` en la carpeta designada. Se especifica el argumento `index=False` para asegurarse de que los índices no se incluyan en el archivo exportado, ya que no son necesarios en este caso para el análisis posterior.

EDA_spotify:

En este Notebook se hará todo el proceso de transformación, limpieza y carga de datos del dataset de spotify.

```

from sqlalchemy import create_engine
import pandas as pd
import os
from dotenv import load_dotenv
from sqlalchemy import text

```

Primero se importan todas las librerías necesarias para hacer el EDA y conectar con la base de datos.

```

df_spotify = pd.read_csv('../raw_data/spotify_dataset.csv')

df_spotify_head = df_spotify.head()

df_spotify_info = df_spotify.info()

(df_spotify_head, df_spotify_info)

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 114000 entries, 0 to 113999
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Unnamed: 0             114000 non-null  int64
1   track_id               114000 non-null  object
2   artists               113999 non-null  object
3   album_name            113999 non-null  object
4   track_name            113999 non-null  object
5   popularity            114000 non-null  int64
6   duration_ms           114000 non-null  int64
7   explicit              114000 non-null  bool
8   danceability          114000 non-null  float64
9   energy                114000 non-null  float64
10  key                   114000 non-null  int64
11  loudness              114000 non-null  float64
12  mode                  114000 non-null  int64
13  speechiness           114000 non-null  float64
14  acousticness          114000 non-null  float64
15  instrumentalness       114000 non-null  float64
16  liveness              114000 non-null  float64
17  valence               114000 non-null  float64
18  tempo                 114000 non-null  float64
19  time_signature         114000 non-null  int64
20  track_genre           114000 non-null  object
dtypes: bool(1), float64(9), int64(6), object(5)
memory usage: 17.5+ MB

```

```

Out[2]: (   Unnamed: 0      track_id      artists \
0         0  5Su0ikwiRyPMVoIQDJUgSV      Gen Hoshino
1         1  4qPNDBW1i3p13qLct0Ki3A      Ben Woodward
2         2  1iJBSr7s7jYXzM8EGcbK5b  Ingrid Michaelson;ZAYN
3         3  6lfxq3CG4xtTiEg7opyCyx      Kina Grannis
4         4  5vjLSffimiIP26QG5WcN2K      Chord Overstreet

          album_name \
0              Comedy
1          Ghost (Acoustic)
2          To Begin Again
3  Crazy Rich Asians (Original Motion Picture Sou...
4              Hold On

      track_name  popularity  duration_ms  explicit \
0              Comedy          73      230666      False
1      Ghost - Acoustic          55      149610      False
2      To Begin Again          57      210826      False
3  Can't Help Falling In Love          71      201933      False
4      Hold On          82      198853      False

  danceability  energy  ...  loudness  mode  speechiness  acousticness \
0         0.676  0.4610  ...    -6.746    0         0.1430         0.0322
1         0.420  0.1660  ...   -17.235    1         0.0763         0.9240
2         0.438  0.3590  ...    -9.734    1         0.0557         0.2100
3         0.266  0.0596  ...   -18.515    1         0.0363         0.9050
4         0.618  0.4430  ...    -9.681    1         0.0526         0.4690

  instrumentalness  liveness  valence  tempo  time_signature  track_genre
0         0.000001    0.3580    0.715   87.917              4      acoustic
1         0.000006    0.1010    0.267   77.489              4      acoustic
2         0.000000    0.1170    0.120   76.332              4      acoustic
3         0.000071    0.1320    0.143  181.740              3      acoustic
4         0.000000    0.0829    0.167  119.949              4      acoustic

[5 rows x 21 columns],

```

El primer paso es la lectura del archivo CSV que contiene los datos de Spotify mediante `pd.read_csv()`. Posteriormente, se muestran las primeras cinco filas del dataset con `head()`, y se utiliza `info()` para obtener una visión general del número de filas, columnas y tipos de datos. Este paso permite entender la estructura inicial del dataset y comprobar si hay valores nulos o inconsistencias.

Columna	Descripción
track_id	Identificador único de la canción.
track_name	Nombre de la canción.
track_artist	Artista o grupo que interpreta la canción.
track_popularity	Popularidad de la canción.
track_album_id	Identificador único del álbum de la canción.
track_album_name	Nombre del álbum de la canción.
track_album_release_date	Fecha de lanzamiento del álbum.
playlist_name	Nombre de la lista de reproducción en la que aparece la canción.
playlist_id	Identificador único de la lista de reproducción.
playlist_genre	Género de la lista de reproducción.
playlist_subgenre	Subgénero de la lista de reproducción.
danceability	Medida de qué tan adecuada es una canción para bailar.
energy	Medida de la intensidad y actividad de una canción.
key	La tonalidad en la que está la canción.
loudness	Volumen general de una canción en decibelios.
mode	Modalidad de la canción (mayor o menor).
speechiness	Medida de la presencia de palabras habladas en una canción.
acousticness	Medida de qué tan acústica es una canción.
instrumentalness	Medida de qué tan instrumental es una canción.
liveness	Medida de la presencia de audiencia en la grabación de una canción.

valence	Medida de la positividad que transmite una canción.
tempo	Tempo de la canción en pulsos por minuto.
duration_ms	Duración de la canción en milisegundos.
time_signature	Compás de la canción.

Se incluye una descripción detallada de cada columna en el dataset de Spotify. Estas columnas incluyen información relevante sobre las canciones, como el identificador único de cada canción (track_id), los artistas

involucrados, el nombre del álbum, la popularidad de la canción (popularity), así como características musicales como la danceability (medida de lo bailable de una canción) y la energy (intensidad). Esta tabla sirve como referencia para comprender mejor los datos a ser analizados.

Cleaning

```
[3]: # Remove any duplicate entries
df_spotify.drop_duplicates(inplace=True)

# Check for and handle missing values
print(df_spotify.isnull().sum())

df_spotify = df_spotify.dropna()

# Ensure data types are correct
print(df_spotify.dtypes)
```

```
Unnamed: 0      0
track_id        0
artists         1
album_name      1
track_name      1
popularity      0
duration_ms     0
explicit        0
danceability    0
energy          0
key             0
loudness        0
mode            0
speechiness     0
acousticness    0
instrumentalness 0
liveness        0
valence         0
tempo           0
time_signature  0
```

Las columnas de texto, como artists, album_name, track_name, y track_genre, se estandarizan utilizando el método str.title() para asegurar que todas las entradas tengan un formato consistente, con las palabras iniciando en mayúsculas. También se eliminan espacios en blanco innecesarios al principio o al final de los textos con str.strip()).


```

# Convert all text columns to a consistent format
text_columns = ['artists', 'album_name', 'track_name', 'track_genre']
for col in text_columns:
    df_spotify[col] = df_spotify[col].str.title()

# Check for and handle missing values
df_spotify.fillna('Unknown', inplace=True)

# Convert explicit column to boolean
df_spotify['explicit'] = df_spotify['explicit'].astype(bool)

# Ensure numerical columns are the correct data type
numerical_columns = ['popularity', 'duration_ms', 'danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness', 'acousticness']
for col in numerical_columns:
    df_spotify[col] = pd.to_numeric(df_spotify[col], errors='coerce') # Coerce any errors to NaN

# Handle any NaN values that could be introduced by conversion errors
df_spotify.dropna(inplace=True)

# Check the cleaned dataset
print(df_spotify.info())
print(df_spotify.head())

```

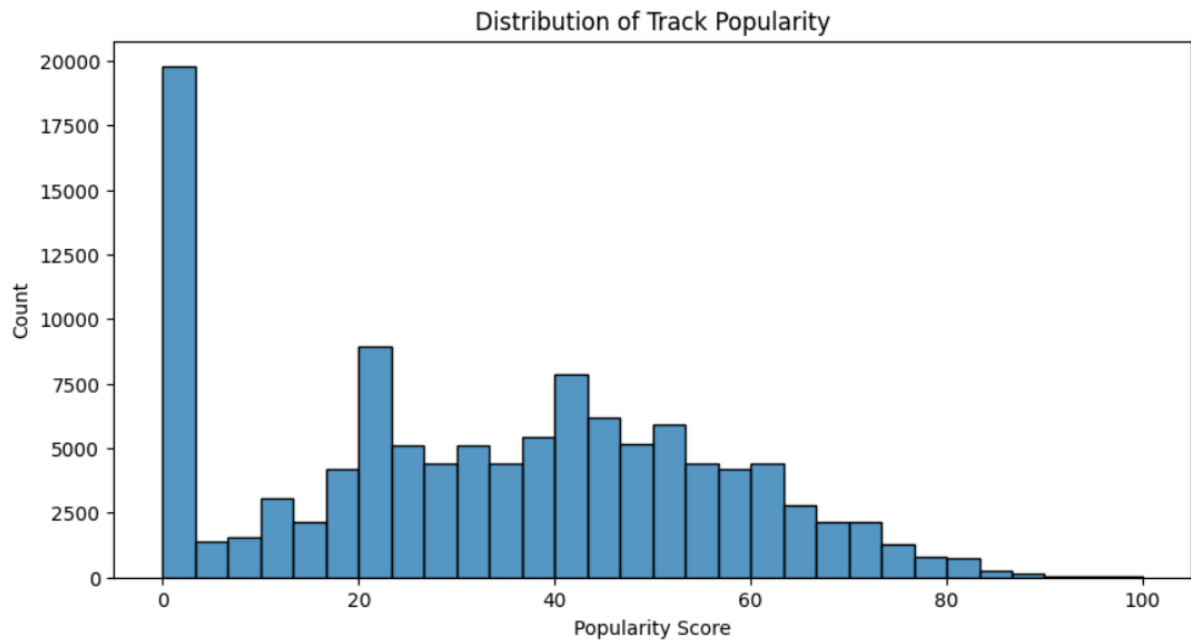
```

<class 'pandas.core.frame.DataFrame'>
Index: 113999 entries, 0 to 113999
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Unnamed: 0             113999 non-null  int64
1   track_id               113999 non-null  object
2   artists                113999 non-null  object
3   album_name             113999 non-null  object
4   track_name             113999 non-null  object
5   popularity             113999 non-null  int64
6   duration_ms            113999 non-null  int64
7   explicit               113999 non-null  bool
8   danceability           113999 non-null  float64
9   energy                 113999 non-null  float64
10  ...                    ...
11  ...                    ...

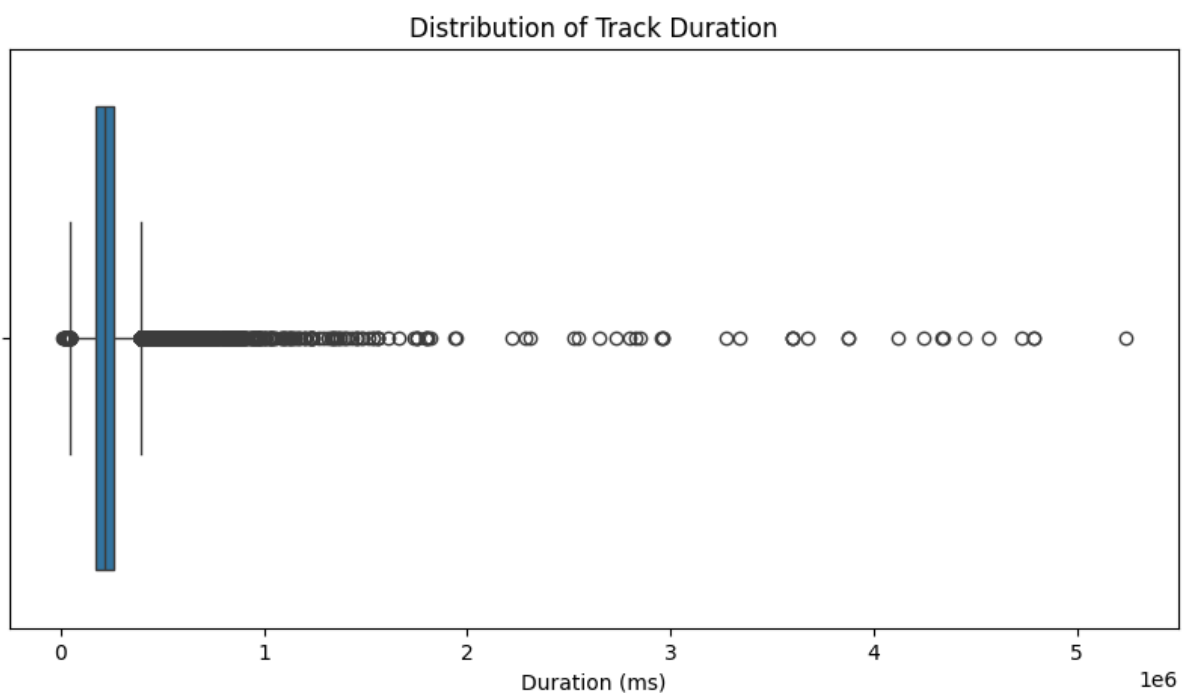
```

La columna explicit se convierte a tipo booleano, asegurando que las canciones con contenido explícito tengan valores True o False. Las columnas numéricas como popularity, duration_ms, y otras relacionadas a las características musicales se convierten al tipo float64 usando pd.to_numeric(). Además, se manejan valores nulos en estas columnas para evitar errores de tipo.

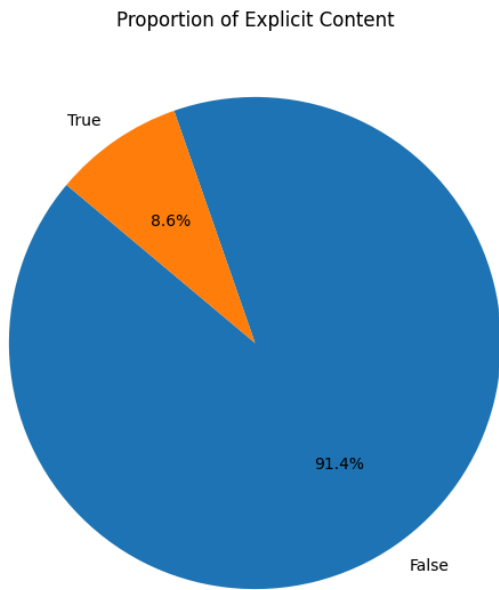
Una vez que se han limpiado los datos, se revisa el dataset resultante utilizando nuevamente info() y head() para verificar que los cambios realizados hayan sido aplicados correctamente. Se confirma que los duplicados y valores nulos han sido tratados, y que los tipos de datos son los adecuados para continuar con el análisis.



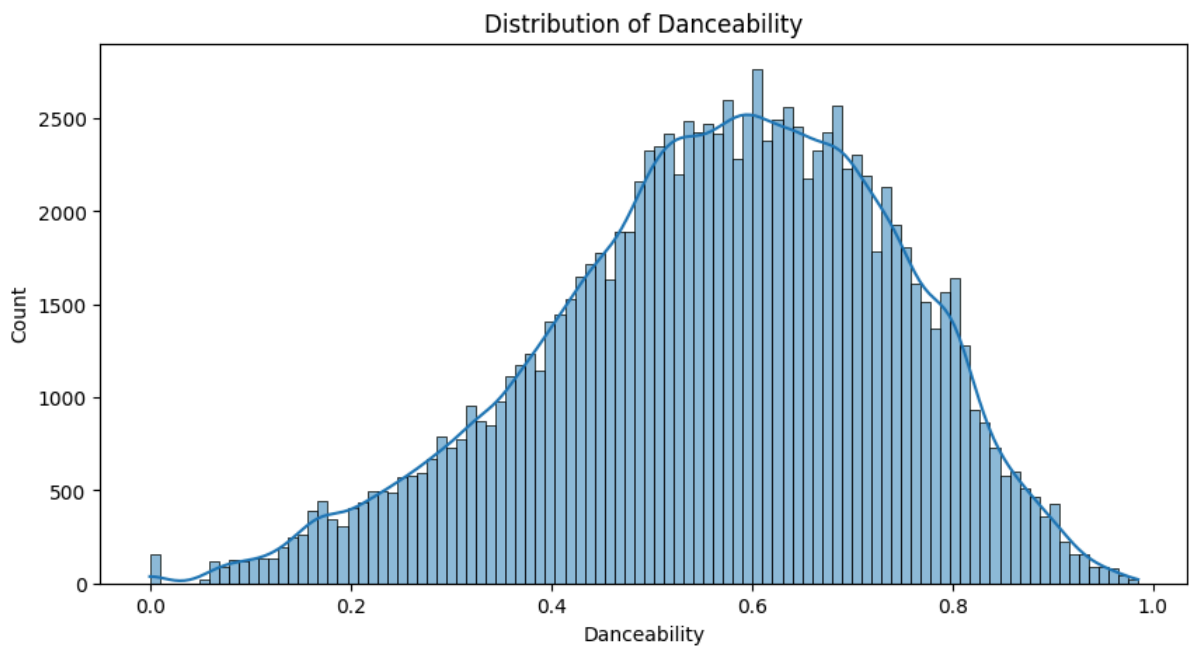
Este histograma muestra cómo se distribuye la popularidad de las canciones en la base de datos. La mayoría de las canciones tienen un puntaje de popularidad muy bajo, alrededor de 0, mientras que una cantidad significativa tiene una popularidad entre 20 y 60.



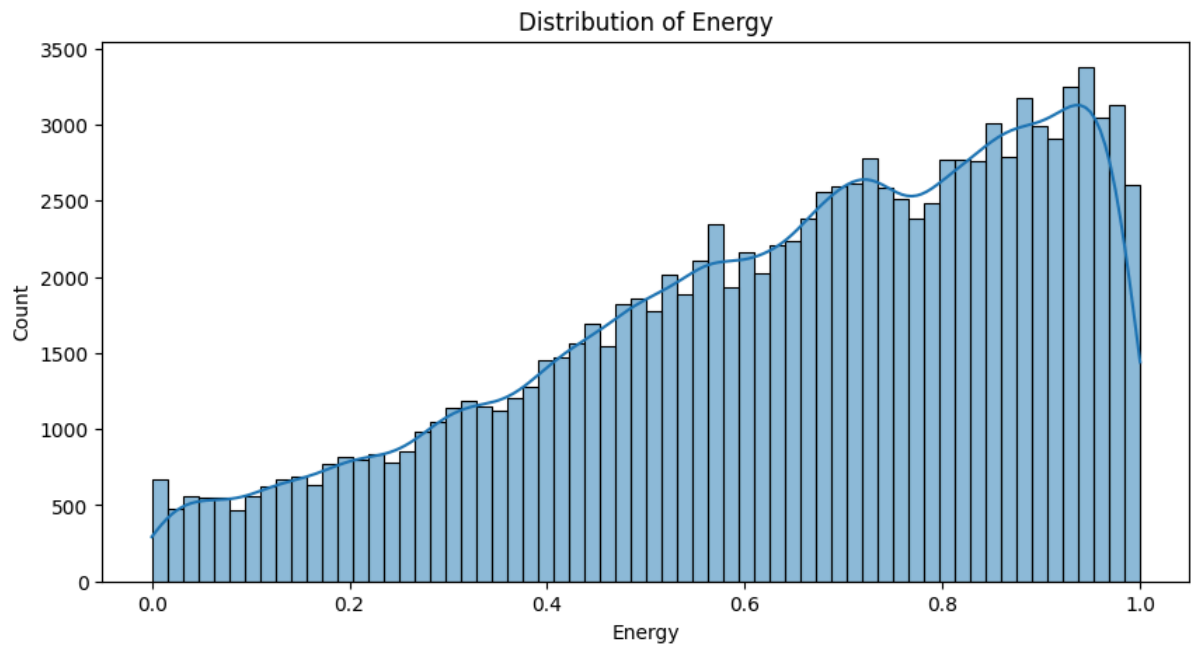
El gráfico de caja revela que la mayoría de las canciones tienen una duración bastante normal, aunque hay algunos valores atípicos que superan los 2 millones de milisegundos, lo cual es inusual.



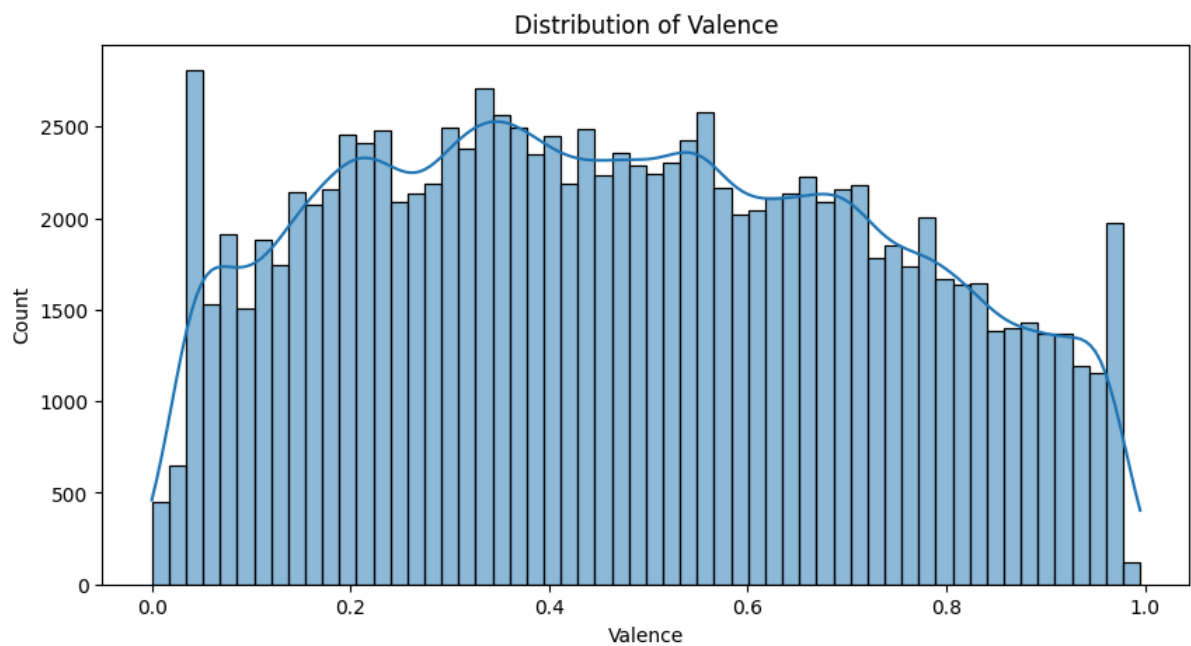
Este gráfico circular muestra que el 91.4% de las canciones no tienen contenido explícito, mientras que solo el 8.6% son explícitas.



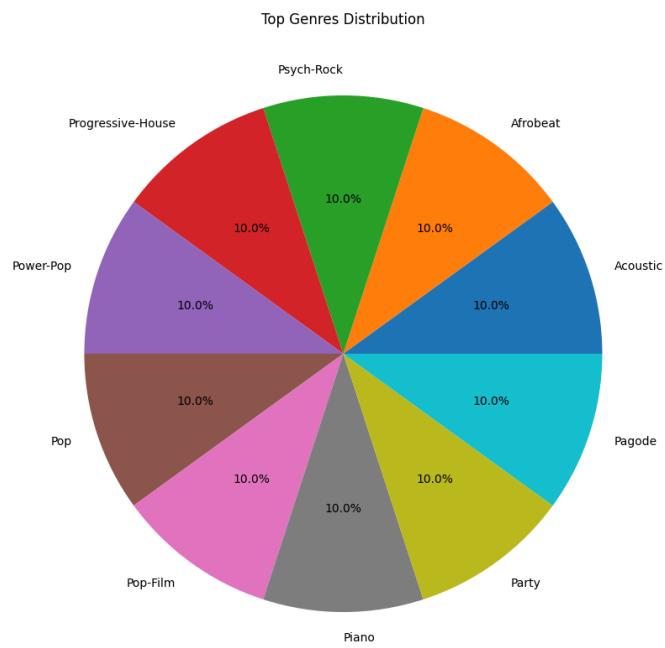
La distribución de la bailabilidad de las canciones sigue una curva casi normal, con la mayoría de las canciones teniendo un puntaje entre 0.5 y 0.8, lo que indica que son moderadamente bailables.



La energía de las canciones tiende a ser alta, con la mayoría de las canciones ubicándose en valores entre 0.5 y 0.9. Esto sugiere que muchas canciones son enérgicas y rápidas.



El gráfico muestra que las canciones tienen una amplia gama de emociones, con un equilibrio entre canciones con un valence bajo (tristeza) y alto (alegría).



Este gráfico circular muestra los géneros más comunes en el dataset. Cada género principal (como Pop, Acoustic, y Afrobeat) representa un 10% de las canciones analizadas.

```
# Zero Popularity Peak Analysis
zero_popularity = df_spotify[df_spotify['popularity'] == 0]
# Output some sample data for tracks with zero popularity
zero_popularity_sample = zero_popularity.sample(5)

# Outliers in Track Duration Analysis
Q1 = df_spotify['duration_ms'].quantile(0.25)
Q3 = df_spotify['duration_ms'].quantile(0.75)
IQR = Q3 - Q1
duration_outliers = df_spotify[(df_spotify['duration_ms'] < (Q1 - 1.5 * IQR)) | (df_spotify['duration_ms'] > (Q3 + 1.5 * IQR))]
# Output some sample data for duration outliers
duration_outliers_sample = duration_outliers.sample(5)

# Audio Features and Genre Relationship Analysis
top_genres = df_spotify['track_genre'].value_counts().nlargest(10).index
top_genres_df = df_spotify[df_spotify['track_genre'].isin(top_genres)]
# Output correlation matrix for top genres
audio_features = ['energy', 'danceability', 'valence']
correlation_matrix = top_genres_df[audio_features].corr()

# Output the count of tracks by genre
genre_distribution = df_spotify['track_genre'].value_counts()

print("Sample Tracks with Zero Popularity:")
print(zero_popularity_sample)

duration_outliers_sample = duration_outliers.sample(5)

# Audio Features and Genre Relationship Analysis
top_genres = df_spotify['track_genre'].value_counts().nlargest(10).index
top_genres_df = df_spotify[df_spotify['track_genre'].isin(top_genres)]
# Output correlation matrix for top genres
audio_features = ['energy', 'danceability', 'valence']
correlation_matrix = top_genres_df[audio_features].corr()

# Output the count of tracks by genre
genre_distribution = df_spotify['track_genre'].value_counts()

print("Sample Tracks with Zero Popularity:")
print(zero_popularity_sample)

print("\nSample Duration Outliers:")
print(duration_outliers_sample)

print("\nAudio Features Correlation Matrix for Top Genres:")
print(correlation_matrix)

print("\nGenre Distribution Count:")
print(genre_distribution)
```

```
Sample Tracks with Zero Popularity:
   Unnamed: 0  track_id \
19128      19128  2GfuDjHEPDpQMAe4mTgQ49
67882      67882  6hp0VLqSgyltDUa7g5CGS8
3849        3849  2oxkdzljjiFWj8lfsZkaCg
64698      64698  29eo4Bc2tFUrS7VoApleeK
59921      59921  2yQ5Ni8Z126C70S3tYQNe0

   artists \
19128  Bailey Zimmerman
67882  Chino & Nacho;Gente De Zona;Los Cadillac'S
3849    La Mosca Tse-Tse
64698  Oscar Peterson
59921  Eternal Griefs

   album_name  track_name  popularity  duration_ms \
19128  Give You Love - Cozy Hits  Fall In Love      0      232058
67882  Halloween 2022 Perreo Vol. 3  Tú Me Quemas      0      269840
3849    Carrete Familiar  Baila Para Mi      0      205386
64698  Ultimate Calm Christmas Jazz  White Christmas      0      228466
59921    Life Is Pain      Is      0      192219

   explicit  danceability  energy  ...  loudness  mode  speechiness \
19128     False      0.524  0.6430  ...    -6.055      1      0.0297
67882     False      0.737  0.9440  ...    -1.436      0      0.0373
...
Electro      1000
World-Music  1000
K-Pop        999
Name: count, Length: 114, dtype: int64
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Distribución de la popularidad de las canciones: En esta gráfica de barras, se observa que una gran cantidad de canciones tienen un puntaje de popularidad de cero, lo cual podría deberse a que estas canciones no han sido reproducidas o no son conocidas. A medida que el puntaje de popularidad aumenta, la frecuencia de canciones disminuye gradualmente.

Distribución de la duración de las canciones: Este diagrama de caja muestra la distribución de la duración de las canciones en milisegundos. Se observa que la mayoría de las canciones tienen una duración dentro de un rango aceptable, pero hay algunos valores atípicos significativos que representan canciones inusualmente largas.

Proporción de contenido explícito: El gráfico de torta ilustra que el 91.4% de las canciones no contienen contenido explícito, mientras que solo el 8.6% sí lo tienen. Esto indica que la mayoría de las canciones son aptas para todo público.

Distribución de la bailabilidad (danceability): La gráfica muestra que la mayoría de las canciones tienen un valor de bailabilidad que oscila alrededor de 0.6, lo que sugiere que la mayoría de las canciones en el dataset son moderadamente bailables.

Distribución de la energía: Aquí observamos que la energía de las canciones tiene una tendencia creciente hacia los valores más altos, lo que indica que muchas canciones son intensas y activas en términos de su energía.

Distribución de la positividad (valence): La distribución de la positividad (valence) presenta una curva que sugiere que las canciones suelen situarse en un rango medio de felicidad o positividad, sin una inclinación fuerte hacia los extremos de ser extremadamente tristes o extremadamente felices.

Distribución de los géneros más populares: El gráfico circular representa los géneros más comunes en el dataset, todos con un 10% de distribución, lo que sugiere que estos géneros están bien distribuidos en el conjunto de datos.

Análisis de canciones con popularidad cero: En el análisis del código, se seleccionan algunas canciones con un puntaje de popularidad cero, lo que ayuda a investigar por qué estas canciones no tienen reproducciones o no son populares. También se analizan los valores atípicos en la duración de las canciones y se examina la correlación entre características de audio y géneros musicales.

Este análisis es útil para comprender cómo las diferentes características musicales, como la duración, la bailabilidad, la energía y la popularidad, se distribuyen en el dataset. Además, resalta la presencia de valores atípicos

que podrían necesitar mayor atención durante el proceso de limpieza y transformación de los datos.

Merge:

La función `combine_csv_files` toma los datasets en formato JSON y los normaliza a DataFrames. A continuación, se utiliza la función `merge` con los parámetros especificados de la siguiente manera: `pd.merge(df1, df2, left_on='nominee', right_on='track_name', how='inner')`. En este proceso, se utilizan las columnas "nominee" y "track_name", combinando los valores que coinciden, ya que ambas representan el nombre del artista. De esta forma, solo se conservan los artistas nominados para el análisis.

```
def combine_csv_files(**context):
    try:
        # Cargar los datos desde XCom
        ti = context["ti"]

        json_data_db = json.loads(ti.xcom_pull(task_ids="validate_grammy_data"))
        df_database = pd.json_normalize(data=json_data_db)

        json_data_csv = json.loads(ti.xcom_pull(task_ids="validate_spotify_csv"))
        df_csv = pd.json_normalize(data=json_data_csv)

        logging.info("CSV data loaded successfully.")

        # Combinar los DataFrames
        combined_df = pd.merge(df_database, df_csv, left_on='nominee', right_on='track_name', how='inner')
        logging.info(f"DataFrames merged successfully. Final shape: {combined_df.shape}")

        # Verificar duplicados
        initial_duplicates = combined_df['track_id'].duplicated().sum()
        logging.info(f"Initial duplicates in 'track_id': {initial_duplicates}")

        combined_df = combined_df.drop_duplicates(subset=['track_id'], keep='first')

        final_duplicates = combined_df['track_id'].duplicated().sum()
        logging.info(f"Remaining duplicates in 'track_id': {final_duplicates}")

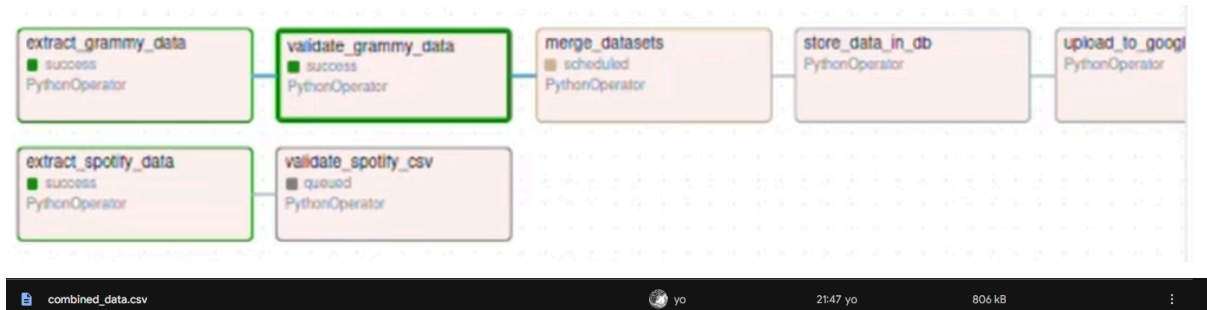
        logging.info("Data cleaned and ready for next step.")

    return combined_df.to_json(orient='records')
```

Airflow:

El proceso en Airflow es un flujo ETL completo, que incluye la extracción de datos tanto desde una base de datos de postgresql como desde un archivo

CSV. Posteriormente, se realiza la transformación, donde se verifican posibles datos nulos o duplicados. Finalmente, el dataset resultante se sube nuevamente a la base de datos y se almacena en Drive, tal como se muestra en la imagen de evidencia.



Conexión a la base de datos usando ngrok para power BI:

Base de datos

Base de datos PostgreSQL

0.tcp.ngrok.io:12930;workshop2

Nombre de usuario
postgres

Contraseña
••••

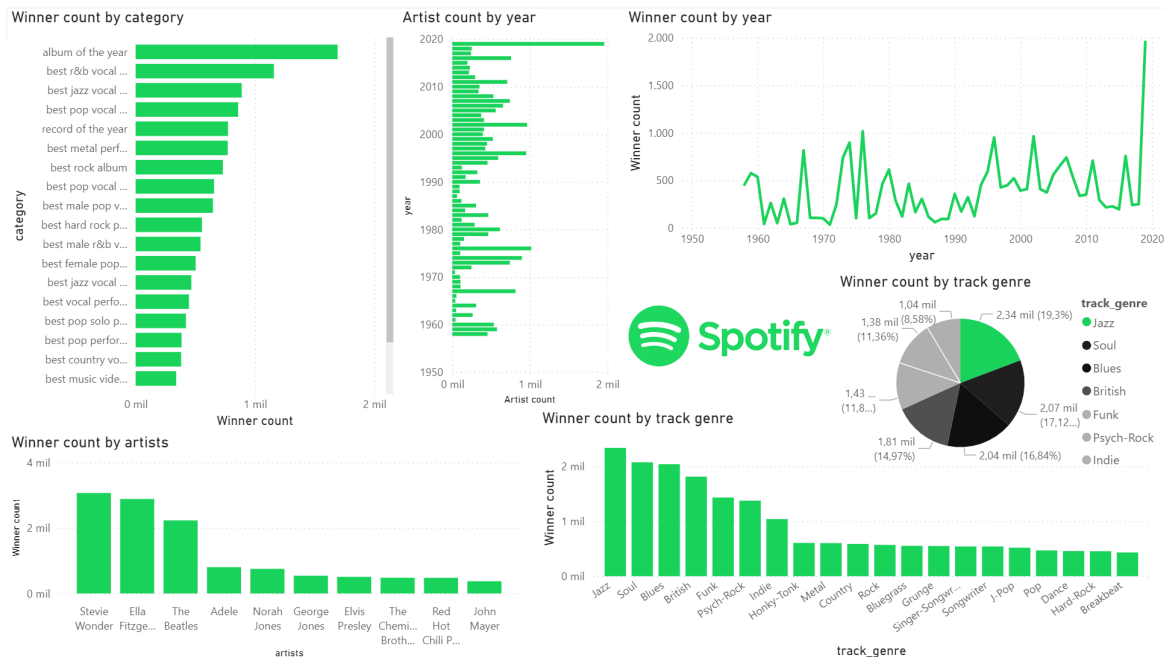
Seleccionar en qué nivel hay que aplicar esta configuración
0.tcp.ngrok.io:12930

Atrás

Conectar

Cancelar

Dashboard:



Winner count by category (Conteo de ganadores por categoría): Las categorías con más ganadores incluyen "Album of the Year", "Best R&B Vocal", "Best Jazz Vocal", y "Best Pop Vocal". Esto refleja que estas categorías son las más competitivas y recurrentes en cuanto a premiaciones en los Grammy, destacando géneros muy populares como el pop y el jazz.

Artist count by year (Conteo de artistas por año): Se observa un aumento progresivo en el número de artistas nominados o ganadores a medida que avanzan los años, con un pico notable en los años 2000 y 2020. Esto podría indicar una diversificación de los artistas y una mayor cantidad de categorías en las premiaciones con el paso del tiempo.

Winner count by year (Conteo de ganadores por año): Se evidencia una tendencia de aumento en el número de ganadores por año, con variaciones significativas, especialmente entre los años 1950 y 1980. A partir del año 2000, hay una estabilización en el número de ganadores, hasta un notable aumento en el 2020. Esto puede sugerir un incremento en las premiaciones o la creación de nuevas categorías en la ceremonia de los Grammy.

Winner count by artists (Conteo de ganadores por artista): Artistas legendarios como Stevie Wonder, Ella Fitzgerald, The Beatles y Adele son quienes han recibido la mayor cantidad de premios Grammy. La presencia de estos artistas destaca su longevidad y relevancia en la industria musical a lo largo de los años.

Winner count by track genre (Conteo de ganadores por género musical):

Los géneros más premiados incluyen "Jazz", "Soul", "Blues" y "Funk". Esto resalta la relevancia y el reconocimiento de estos géneros en la música contemporánea, especialmente en los Grammy.

Winner count by track genre (Pie chart) (Distribución de géneros musicales por número de ganadores): El gráfico circular muestra la distribución proporcional de ganadores por género musical, destacando "Jazz" con el mayor porcentaje de ganadores (19.3%), seguido de "Soul" y "Blues". Este gráfico ofrece una visión más clara de cómo se distribuyen los premios entre los diferentes géneros musicales, reafirmando la prominencia de ciertos géneros en los premios Grammy.