



Workshop #2 - Documentation

Martín García Chagüezá

[About the workshop](#)

[Goals of the project](#)

[Data flow](#)

[Process](#)

[Loading raw Grammys dataset](#)

[db_operations.py](#) - The details behind the DB connection and the load of the raw data

[Exploring the Data - Spotify dataset](#)

[The most popular songs tend to have high volume levels](#)

[Popular songs show high variability in terms of **danceability** and **energy**](#)

[Happy songs tend to be more energetic and danceable](#)

[Urban, Pop and Latin have the highest average popularity](#)

[Exploring the Data - Grammys dataset](#)

[The number of Grammy nominations was steadily increased over the decades, but...](#)

[Certain artists dominated the nominations in specific decades](#)

[Collaborative or compilation albums received the highest number of nominations](#)

[The structure of the Data Pipeline](#)

[Extraction of the data](#)

[Transformations in the *Spotify* dataset](#)

[Transformations in the *Grammys* dataset](#)

[Merging the datasets](#)

[Loading the data](#)

[Storing the data](#)

[Managing the data pipeline with Airflow](#)

[Analyzing the composition of the DAG](#)

[Visualizing the Data](#)

[Page 1: **Nominations**](#)

[Page 2: **Song Characteristics**](#)

[Conclusions](#)

About the workshop


In this workshop we will use two datasets (*spotify_dataset* and *the_grammys_awards*) that will be processed through Apache Airflow applying data cleaning, transformation and loading and storage, including a merge of both datasets. The result will culminate in visualizations on a dashboard that will give us important conclusions about this dataset.

The tools used are:

- Python 3.10 → [Download site](#)
- Jupyter Notebook → [VS Code tool for using notebooks](#)
- PostgreSQL → [Download site](#)
- Power BI (Desktop version) → [Download site](#)

The dependencies needed for Python are

- Apache Airflow
- Dotenv
- Pandas
- Matplotlib
- Seaborn
- SQLAlchemy
- PyDrive2

 Apache Airflow **only runs correctly in Linux environments**. If you have Windows, we recommend using a virtual machine or WSL.

These dependencies are included in the `requirements.txt` file of the Python project. The step-by-step installation is described in the README file.

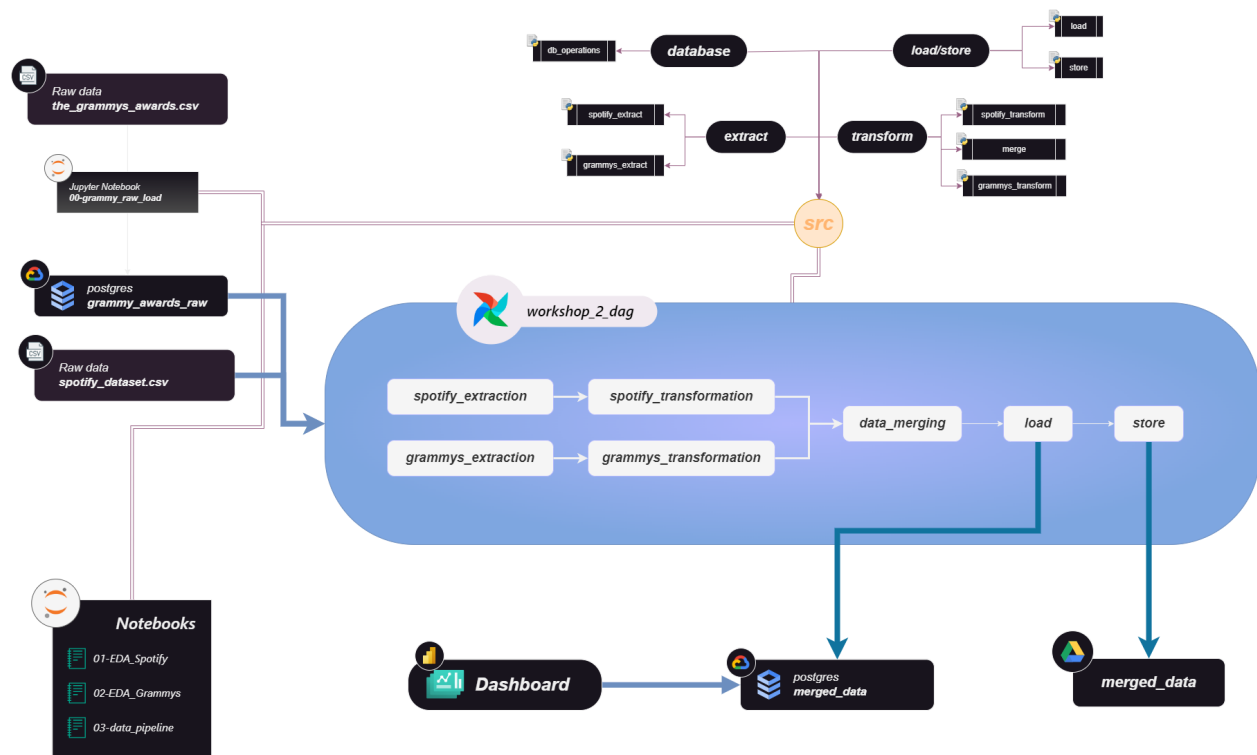
Goals of the project

Obtain the clean merged dataset for the creation of an analytical report using BI tools such as Power BI, which will be connected through a PostgreSQL database that will contain the transformed data set.

The analytical report will contain visualizations such as:

- **Nominated Songs per Genre:** A bar chart showing the number of nominated songs across different genres like Electronic/Dance, Rock/Metal, and Jazz and Soul.
- **Nominations per Artist:** A chart that displays how many nominations various artists, such as BTS, Adele, and Glee Cast, have received.
- **Energy Average per Track Genre:** A bar graph showing the average energy per track for different genres, indicating the relative energy levels of songs in each genre.
- **Danceability Average per Track Mood:** A chart showing the average danceability based on the mood of the track (e.g., Happy, Neutral, Sad).

Data flow



Process

Loading raw Grammys dataset

Files used → `db_operations.py` / `00-grammy_raw_load.ipynb`

Before we can run the EDA corresponding to the Grammy Awards dataset, we must upload it to a PostgreSQL database. **This process is mandatory if you wish to continue with the execution of the project.**

The process is described step by step in notebook 00, therefore, we are going to dwell on the details of the module in charge of the operations performed on the database.

`db_operations.py` - The details behind the DB connection and the load of the raw data

To run the connection engine we must specify a URL with the credentials of our database. These credentials are collected in the `.env` file that specifies the environment variables for our project.

Once we have these credentials, the engine is created (*as well as the database, in case it does not exist*); when we no longer need the engine, we have a function to terminate it and, thus, free memory.

```
# Reading the environment variables
load_dotenv("../env/.env")

user = os.getenv("PG_USER")
password = os.getenv("PG_PASSWORD")

host = os.getenv("PG_HOST")
port = os.getenv("PG_PORT")

database = os.getenv("PG_DATABASE")

# Creating the connection engine from the URL made up of the environment variables
def creating_engine():
    url = f"postgresql://{user}:{password}@{host}:{port}/{database}"
    engine = create_engine(url)

    if not database_exists(url):
        create_database(url)
        logging.info("Database created")

    logging.info("Engine created. You can now connect to the database.")

    return engine

def disposing_engine(engine):
    engine.dispose()
    logging.info("Engine disposed.")
```

The loading of the raw data is specified in a function that receives the engine, the Pandas dataframe and the table name. The data types are inferred by Pandas through its `to_sql` function, which is in charge of transferring the data to the database.

```
# Creating table and loading the raw data
def load_raw_data(engine, df, table_name):

    logging.info(f"Creating table {table_name} from Pandas DataFrame.")

    try:

        df.to_sql(table_name, con=engine, if_exists="replace", index=False)

        logging.info(f"Table {table_name} created successfully.")

    except Exception as e:

        logging.error(f"Error creating table {table_name}: {e}")
```

Query Query History

1

SELECT * FROM public.grammy_awards_raw

2

Data Output Messages Notifications

	year bigint	title text	published_at text	updated_at text	category text	nominee text	artist text	workers text
1	2019	62nd Annual GRAMMY Awards (201...	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	Bad Guy	Billie Eilish	Finneas O'Connell, producer; Rob Kinski &
2	2019	62nd Annual GRAMMY Awards (201...	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	Hey, Ma	Bon Iver	BJ Burton, Brad Cook, Chris Messina & Just
3	2019	62nd Annual GRAMMY Awards (201...	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	7 rings	Ariana Grande	Charles Anderson, Tommy Brown, Michael I
4	2019	62nd Annual GRAMMY Awards (201...	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	Hard Place	H.E.R.	Rodney "Darkchild" Jerkins, producer; Jose
5	2019	62nd Annual GRAMMY Awards (201...	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	Talk	Khalid	Disclosure & Denis Kosiak, producers; Ing
6	2019	62nd Annual GRAMMY Awards (201...	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	Andrew "VoxGod" Bolooki, Jocelyn "Jozzy" I
7	2019	62nd Annual GRAMMY Awards (201...	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	Truth Hurts	Lizzo	Ricky Reed & Tele, producers; Chris Galland
8	2019	62nd Annual GRAMMY Awards (201...	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	Sunflower	Post Malone & Swae Lee	Louis Bell & Carter Lang, producers; Louis E
9	2019	62nd Annual GRAMMY Awards (201...	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	Finneas O'Connell, producer; Rob Kinski &
10	2019	62nd Annual GRAMMY Awards (201...	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Album Of The Year	IJ	Bon Iver	Brad Cook, Chris Messina & Justin Vernon,
11	2019	62nd Annual GRAMMY Awards (201...	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	Jack Antonoff & Lana Del Rey, producers; J
12	2019	62nd Annual GRAMMY Awards (201...	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Album Of The Year	thank u, next	Ariana Grande	Tommy Brown, Ilya, Max Martin & Victoria K
13	2019	62nd Annual GRAMMY Awards (201...	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.	David "Swagg R'Cellous" Harris, H.E.R., Wall
14	2019	62nd Annual GRAMMY Awards (201...	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Album Of The Year	7	Lil Nas X	Joe Grasso, engineer/mixer; Montero Lama
15	2019	62nd Annual GRAMMY Awards (201...	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Album Of The Year	Cuz I Love You (Deluxe)	Lizzo	Ricky Reed, producer; Manny Marroquin & E
16	2019	62nd Annual GRAMMY Awards (201...	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Album Of The Year	Father Of The Bride	Vampire Weekend	Ezra Koenig & Ariel Rechtshaid, producers; .
17	2019	62nd Annual GRAMMY Awards (201...	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Song Of The Year	Bad Guy	[null]	Billie Eilish O'Connell & Finneas O'Connell, s
18	2019	62nd Annual GRAMMY Awards (201...	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Song Of The Year	Always Remember Us This Way	[null]	Natalie Hemby, Lady Gaga, Hillary Lindsey &
19	2019	62nd Annual GRAMMY Awards (201...	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Song Of The Year	Bring My Flowers Now	[null]	Brandi Carlile, Phil Hanseroth, Tim Hanseroth
20	2019	62nd Annual GRAMMY Awards (201...	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Song Of The Year	Hard Place	[null]	Ruby Amanfu, Sam Ashworth, D. Arcelious I
21	2019	62nd Annual GRAMMY Awards (201...	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Song Of The Year	Lover	[null]	Taylor Swift, songwriter (Taylor Swift)

Total rows: 1000 of 4810 Query complete 00:00:01.626 Ln 1, Col 1

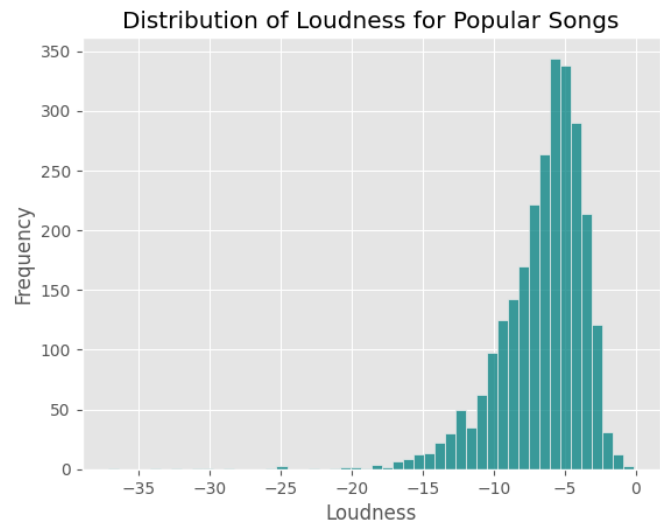
Exploring the Data - Spotify dataset

Files used → 01-EDA_Spotify.ipynb

In order not to make the analysis of certain variables more difficult, we chose to select a sample of the dataset: *the most popular songs* were selected from it. Some of the findings were:

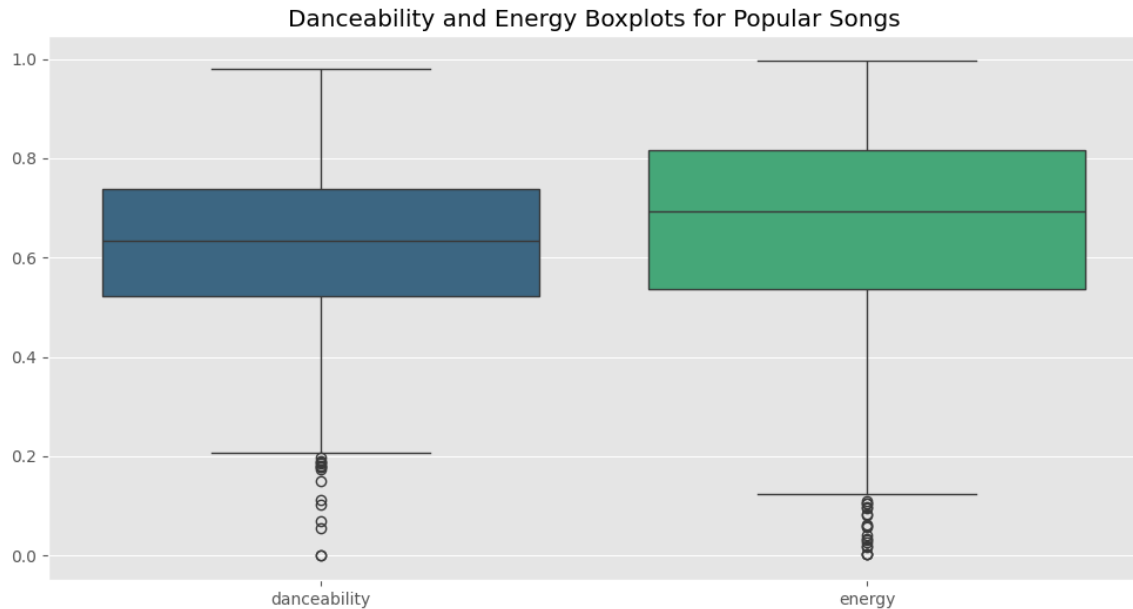
The most popular songs tend to have high volume levels

These high volume levels range from -10 dB to -2.5 dB, suggesting that the trend in popular music is to **maintain high volume levels**.



Popular songs show high variability in terms of **danceability** and **energy**

- For **energy**, the median of 0.7 implies that popular songs are typically energetic. This aligns with the idea that high-energy tracks are more likely to engage listeners and perform well on charts.
- Regarding **danceability**, the median of 0.65 suggests that popular songs generally have a moderate to high danceability. This indicates that the music industry tends to favor tracks that are easy to dance to, potentially to appeal to a wider audience.

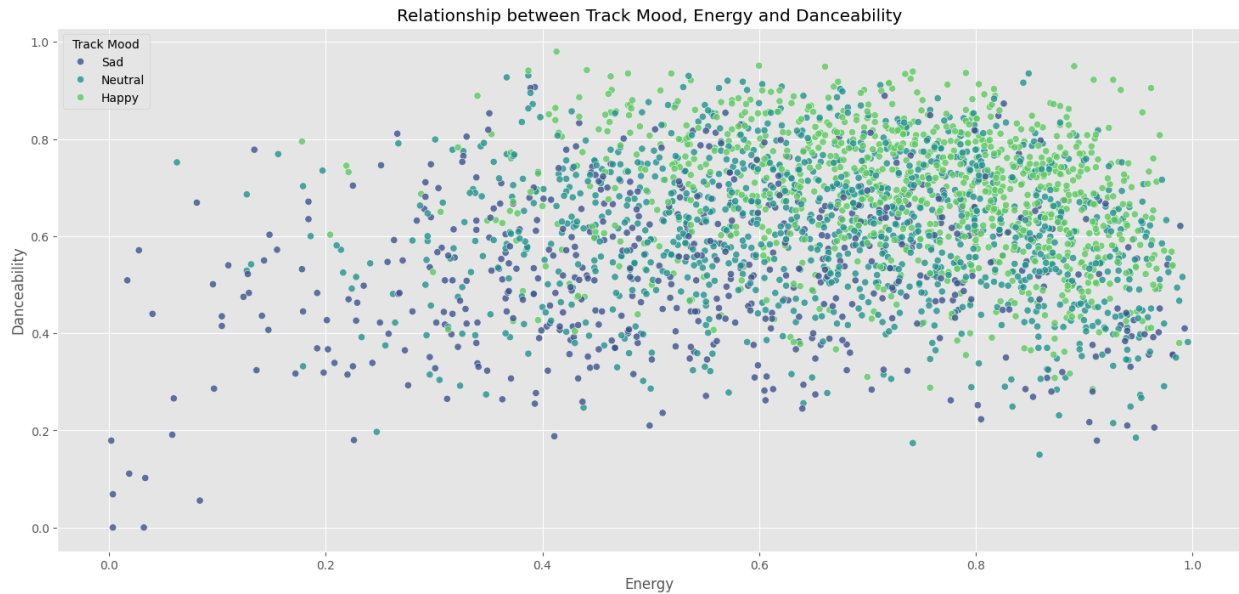


Happy songs tend to be more energetic and danceable

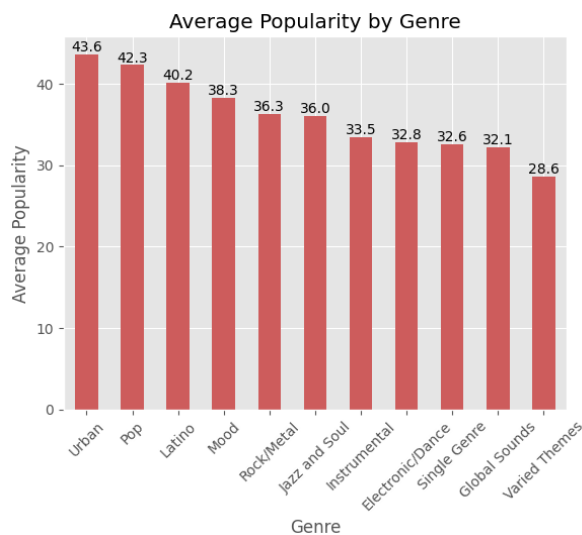
A scatter plot analysis reveals interesting patterns in the relationship between song moods and their energy and danceability characteristics:

- *Sad songs* tend to have **lower energy and danceability**.
- *Happy songs* generally cluster at **high levels of energy and danceability**.
- *Neutral songs* show a **more varied distribution**.

Regardless of mood, many popular songs tend to have moderate to high levels of energy and danceability, suggesting a general preference for these characteristics in mainstream music.



Urban, Pop and Latin have the highest average popularity



This suggests that the most produced genres **are not always the most popular**, while categories with fewer songs, such as *Urban*, have **a greater impact** on the audience.

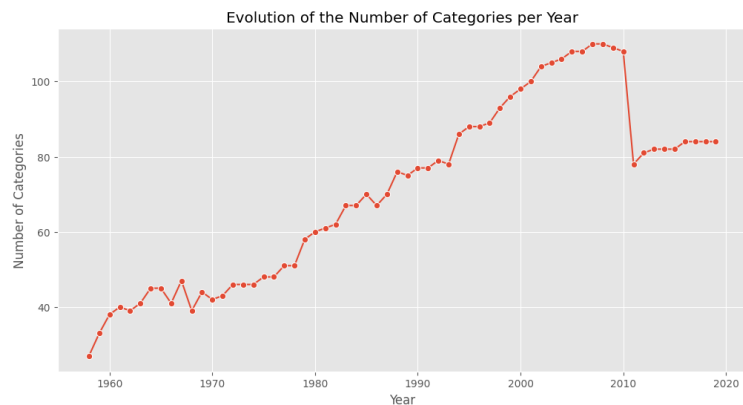
Exploring the Data - Grammys dataset

Files used → → `db_operations.py` / `02-EDA_Grammys.ipynb`

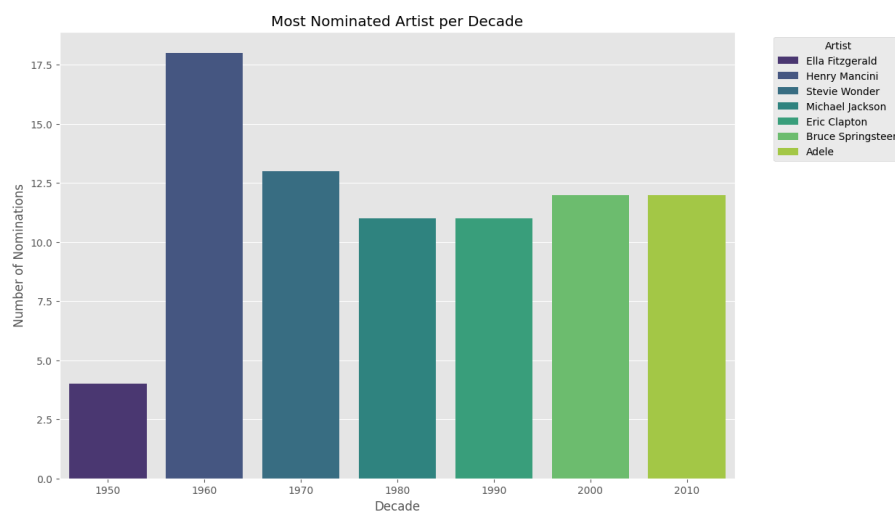
In order not to make the analysis of certain variables more difficult, we chose to select a sample of the dataset: *the most popular songs* were selected from it. Some of the findings were:

The number of Grammy nominations was steadily increased over the decades, but...

The number of Grammy nominations were at its peak in 2010, but a significant reorganization in 2011-2012 to streamline the awards caused a massive reduction in the number of nominations.



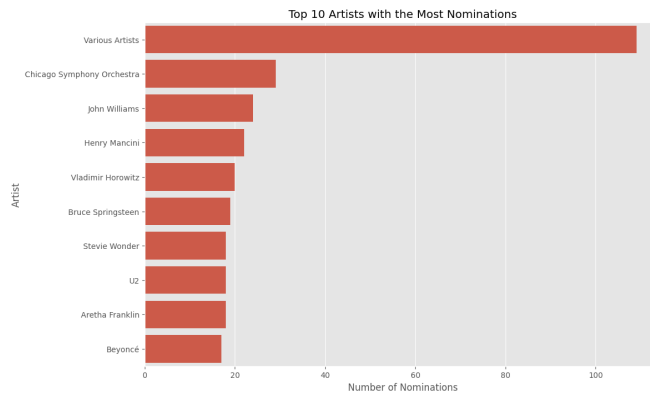
Certain artists dominated the nominations in specific decades



These trends correspond to changes in musical trends and popularity.

Collaborative or compilation albums received the highest number of nominations

These kind of nominations are represented by the title of *"Various Artists"*: they accumulate almost 100 nominations.



The structure of the Data Pipeline

Files used → every module on **src** / 03-data_pipeline.ipynb

This notebook goes through the process performed in Apache Airflow applying the functions of the different modules that are part of the **src** directory. However, special emphasis is placed on **the merge between the two datasets**.

Some of the remarks that can be highlighted are:

Extraction of the data

Here it's used both modules of the `src.extract` package. While the Spotify dataset is extracted locally, the Grammy Awards dataset is extracted by connecting to the database.

```
In [3]: spotify_data = extracting_spotify_data("../data/spotify_dataset.csv")
```

23/09/2024 09:11:49 PM Data extracted from ../data/spotify_dataset.csv.

```
In [4]: spotify_data.head()
```

```
Out[4]:
```

	Unnamed: 0	track_id	artists	album_name	track_name	popularity	duration_ms	explicit	danceability	ene
0	0	5SuOikwiRyPMVolQDJUgSV	Gen Hoshino	Comedy	Comedy	73	230666	False	0.676	0.4
1	1	4qPND8W1i3p13qLct0Ki3A	Ben Woodward	Ghost (Acoustic)	Ghost - Acoustic	55	149610	False	0.420	0.1
2	2	1iJBSr7s7jYXzM8EGcbK5b	Ingrid Michaelson:ZAYN	To Begin Again	To Begin Again	57	210826	False	0.438	0.3
3	3	6lfxq3CG4xtTiEg7opyCyx	Kina Grannis	Crazy Rich Asians (Original Motion Picture Sou...	Can't Help Falling In Love	71	201933	False	0.266	0.0
4	4	5vjLSffimilP26QG5WcN2K	Chord Overstreet	Hold On	Hold On	82	198853	False	0.618	0.4

5 rows × 21 columns

Press **SHIFT** + **F** for Shortcuts

DAG: workshop2_dag / Run: 2024-09-23, 00:00:00 UTC / Task: spotify_extraction

Clear task | Mark state as... | Filter DAG by task

Details | Graph | Gantt | Code | Event Log | Logs | XCom | Task Duration

All Levels | All File Sources | Wrap | Download | See More

Large log file. Some lines have been truncated. Download logs in order to see everything.

```
DESKTOP-HW474JE.
*** Found local files:
*** + /home/nitigar34/ETL/etl-workshop-2/airflow/logs/dag_id=workshop2_dag/run_id=scheduled_2024-09-23T00-00-00-00/task_id=spotify_extraction/attempt=1.log
[2024-09-24, 02:26:17 UTC] [local_task_job_runner.py:123] ▶ Pre task execution logs
[2024-09-24, 02:26:17 UTC] [spotify_extract.py:18] INFO - Data extracted from ../data/spotify_dataset.csv.
[2024-09-24, 02:26:18 UTC] [python.py:240] INFO - Done. Returned value was: [{"Unnamed: 0":0,"track_id":"5SuOikwiRyPMVolQDJUgSV","artists":"Gen Hoshino","album_name":"Comedy","track_name":"Comedy","popularity":73,"duration_ms":230666,"
[2024-09-24, 02:26:19 UTC] [taskinstance.py:340] ▶ Post task execution logs
```

```
In [5]: grammys_data = extracting_grammys_data()
```

```
23/09/2024 09:11:52 PM Engine created. You can now connect to the database.
23/09/2024 09:11:52 PM Extracting data from the Grammy Awards table.
23/09/2024 09:11:56 PM Data extracted from the Grammy Awards table.
23/09/2024 09:11:56 PM Engine disposed.
```

```
In [6]: grammys_data.head()
```

Out[6]:	year	title	published_at	updated_at	category	nominee	artist	workers	img	wi
0	2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	Bad Guy	Billie Eilish	Finneas O'Connell, producer; Rob Kinelski & Fi...	https://www.grammy.com/sites/com/files/styles/...	
1	2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	Hey, Ma	Bon Iver	BJ Burton, Brad Cook, Chris Messina & Justin V...	https://www.grammy.com/sites/com/files/styles/...	
2	2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	7 rings	Ariana Grande	Charles Anderson, Tommy Brown, Michael Foster ...	https://www.grammy.com/sites/com/files/styles/...	
3	2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	Hard Place	H.E.R.	Rodney "Darkchild" Jerkins, producer; Joseph H...	https://www.grammy.com/sites/com/files/styles/...	
4	2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year	Talk	Khalid	Disclosure & Denis Kosiak, producers; Ingmar C...	https://www.grammy.com/sites/com/files/styles/...	

Press (SHIFT) + ⌘ for Shortcuts

workshop2_dag / 2024-09-23, 00:00:00 UTC / grammys_extraction

Clear task Mark state as... Filter DAG by task

Details Graph Gantt Code Event Log Logs XCom Task Duration

All Levels All File Sources

Large log file. Some lines have been truncated. Download logs in order to see everything.

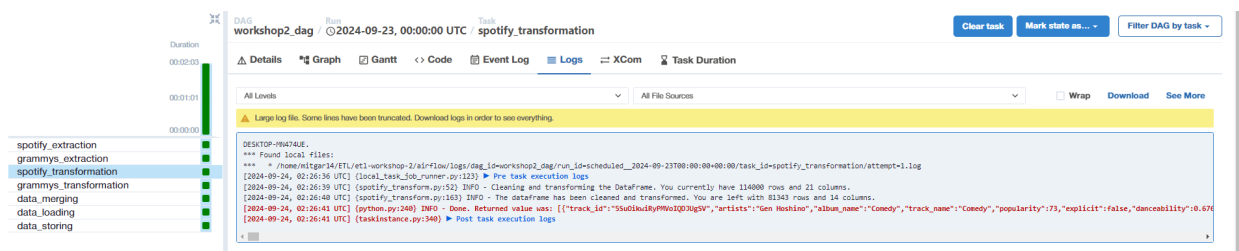
```
desktop-macache:
*** Found local files:
*** * /home/etiger14/ETL/etl-workshop-2/airflow/logs/dag_id=workshop2_dag/run_id=scheduled_2024-09-23T00:00:00-00:00/task_id=grammys_extraction/attempt=1.log
[2024-09-24, 02:26:22 UTC] [local_task_job_runner.py:125] ▶ Pre task execution logs
[2024-09-24, 02:26:25 UTC] [db_operations.py:32] INFO - Engine created. You can now connect to the database.
[2024-09-24, 02:26:25 UTC] [grammys_extractor.py:18] INFO - Extracting data from the Grammy Awards table.
[2024-09-24, 02:26:30 UTC] [grammys_extractor.py:18] INFO - Data extracted from the Grammy Awards table.
[2024-09-24, 02:26:30 UTC] [db_operations.py:38] INFO - Engine disposed.
[2024-09-24, 02:26:30 UTC] [python.py:240] INFO - Done. Returned value was: [{"year": 2019, "title": "62nd Annual GRAMMY Awards (2019)", "published_at": "2020-05-19T05:10:28-07:00", "updated_at": "2020-05-19T05:10:28-07:00", "category": "Recor
[2024-09-24, 02:26:30 UTC] [taskinstance.py:340] ▶ Post task execution logs
```

Transformations in the Spotify dataset

module used was → `transform.spotify_transform`

- Removing unnecessary columns (e.g., "Unnamed: 0").
- Eliminating null values and resetting the DataFrame index.

- Removing duplicates through several steps:
 - Dropped exact duplicate rows.
 - Removed duplicates based on the `"track_id"` column.
 - Mapped detailed genres to broader categories using a predefined genre mapping dictionary.
 - Dropped duplicates based on song names and artists, keeping the most popular entries.
- Generated new columns for enhanced data analysis:
 - `duration_min` : Converted song duration from milliseconds to minutes.
 - `duration_category` : Categorized songs based on their duration.
 - `popularity_category` : Categorized songs based on their popularity scores.
 - `track_mood` : Identified the mood of songs using valence scores.
 - `live_performance` : Flagged songs with a high likelihood of being live performances.
- Dropped irrelevant columns to streamline the dataset (e.g., `"loudness"` , `"mode"` , `"tempo"`).

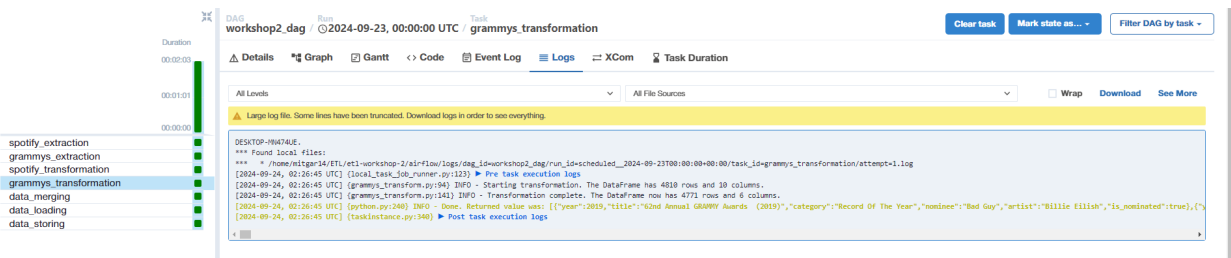


Transformations in the *Grammys* dataset

module used was → `transform.grammys_transform`

- Renaming the column `winner` to `is_nominated` .
- Dropping unnecessary columns (e.g., `published_at` , `updated_at` , `img`).

- Removing rows with null values in `nominee`.
- Handling cases where both `artist` and `workers` are null:
 - Filtered out specific categories listed in the `categories` list.
 - For the remaining rows, filled `artist` with the value from `nominee`.
- Populating the `artist` column by applying several functions:
 - `extract_artist`: Extracted artist names within parentheses from the `workers` column.
 - `move_workers_to_artist`: Moved data from `workers` to `artist` if `artist` is null and `workers` doesn't contain semicolons or commas.
 - `extract_artists_before_semicolon`: Extracted artist names before semicolons in `workers`, excluding any roles of interest.
 - `extract_roles_based_on_interest`: Extracted names associated with specific roles defined in the `roles_of_interest` list from `workers`.
- Dropped rows with null values in `artist`.
- Replaced certain values in the `artist` column (e.g., changing `(Various Artists)` to `Various Artists`).
- Dropped the `workers` column as it was no longer needed.



Merging the datasets

module used was → `transform.merge`

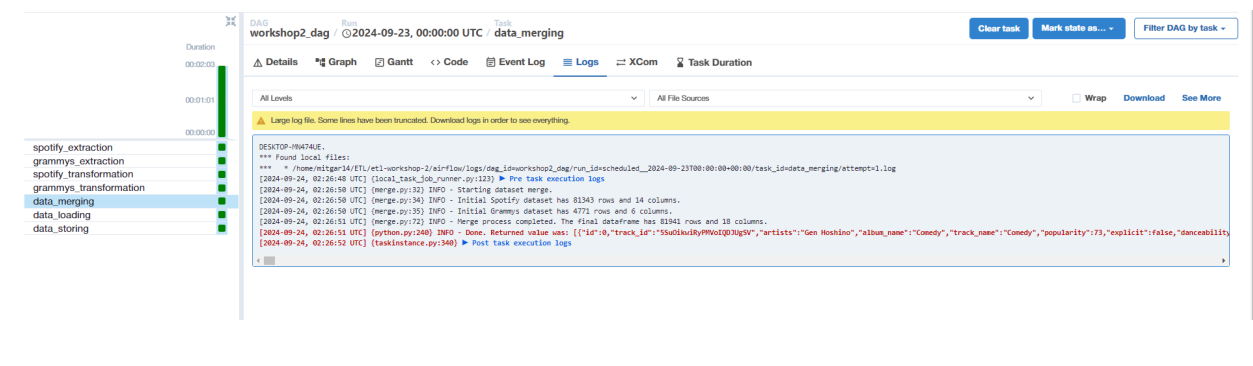
Before listing the processes that occurred within the module it should be noted that the criteria chosen for the merge, although effective for most cases, yields

quite a few false positives.

In order to fix these errors we must include the artist's name in the merge criteria. However, it would be necessary to analyze artist one by one if we want a successful merge.

The cleaning process executed was:

- **Cleaning key columns for accurate merging:**
 - Converted the `track_name` column in the Spotify DataFrame to lowercase and stripped whitespace, creating a new column `track_name_clean`.
 - Converted the `nominee` column in the Grammys DataFrame to lowercase and stripped whitespace, creating a new column `nominee_clean`.
- **Merging the datasets:**
 - Performed a left join on the cleaned columns `track_name_clean` and `nominee_clean` to merge the DataFrames.
 - Used suffixes to differentiate overlapping columns, appending `_grammys` to columns from the Grammys DataFrame when necessary.
- **Handling missing values:**
 - Filled null values in the `title` and `category` columns with `"Not applicable"`.
 - Filled null values in the `is_nominated` column with `False`.
- **Dropping unnecessary columns:**
 - Removed columns that were no longer needed after the merge, such as `"year"`, `"artist"`, `"nominee"`, `"nominee_clean"`, and `"track_name_clean"`.



The screenshot displays the Airflow web interface for a task named 'data_merging' within a DAG called 'workshop2_dag'. The task is shown as completed with a green status icon. The logs for this task are visible, showing the execution of a Python script that performs data merging. The logs include the following information:

- Task ID: data_merging
- Task Name: data_merging
- Task Duration: 00:00:00
- Task State: Success
- Task Log: The log content shows the execution of a Python script that performs data merging. It includes the following information:
 - Task ID: data_merging
 - Task Name: data_merging
 - Task Duration: 00:00:00
 - Task State: Success
 - Task Log: The log content shows the execution of a Python script that performs data merging. It includes the following information:
 - Task ID: data_merging
 - Task Name: data_merging
 - Task Duration: 00:00:00
 - Task State: Success
 - Task Log: The log content shows the execution of a Python script that performs data merging. It includes the following information:

Loading the data

Although it is not necessary for the notebook, it is recommended that the data be uploaded to a cloud database due to the complexity of handling PostgreSQL in WSL, especially when handling Apache Airflow.

```
In [18]: loading_merged_data(merged_df, "merged_data")
```

```
23/09/2024 09:11:58 PM Loading clean data to the database.
23/09/2024 09:12:01 PM Engine created. You can now connect to the database.
23/09/2024 09:12:01 PM Creating table merged_data from Pandas DataFrame.
23/09/2024 09:12:03 PM Adjusting column artists to Text due to length 513.
23/09/2024 09:12:03 PM Adjusting column track_name to Text due to length 511.
23/09/2024 09:12:04 PM Table merged_data created successfully.
23/09/2024 09:12:55 PM Data loaded to table merged_data.
23/09/2024 09:12:55 PM Engine disposed.
```

workshop2_dag / 2024-09-23, 00:00:00 UTC / data_loading

Clear task Mark state as... Filter DAG by task

Details Graph Gantt Code Event Log Logs XCom Task Duration

All Levels All File Sources Wrap Download See More

Large log file. Some lines have been truncated. Download logs in order to see everything.

```
DESKTOP-HM24LE.
*** Found local files:
*** /home/mrizer14/ETL/etl-workshop-2/airflow/logs/dag_id=workshop2_dag/run_id=scheduled_2024-09-23T00:00:00/task_id=data_loading/attempt=1.log
[2024-09-24, 02:26:55 UTC] (local_task_job_runner.py:122) ▶ Pre task execution logs
[2024-09-24, 02:26:57 UTC] (load.py:25) INFO - Loading clean data to the database.
[2024-09-24, 02:26:59 UTC] (db_operations.py:32) INFO - Engine created. You can now connect to the database.
[2024-09-24, 02:26:59 UTC] (db_operations.py:77) INFO - Creating table merged_data from Pandas DataFrame.
[2024-09-24, 02:27:01 UTC] (db_operations.py:49) INFO - Adjusting column artists to Text due to length 513.
[2024-09-24, 02:27:02 UTC] (db_operations.py:49) INFO - Adjusting column track_name to Text due to length 511.
[2024-09-24, 02:27:02 UTC] (db_operations.py:90) INFO - Table merged_data created successfully.
[2024-09-24, 02:27:48 UTC] (db_operations.py:94) INFO - Data loaded to table merged_data.
[2024-09-24, 02:27:48 UTC] (db_operations.py:38) INFO - Engine disposed.
[2024-09-24, 02:27:49 UTC] (python.py:240) INFO - Done. Returned value was: [{"id":0,"track_id":"5Su0ikwiRyPMVoQJUGSV","artists":"Gen Hoshino","album_name":"Comedy","track_name":"Comedy","popularity":73,"explicit":false,"danceability":0.461,"energy":0.461,"instrumental":false,"type":"audio/mp4"}]
[2024-09-24, 02:27:49 UTC] (taskinstance.py:340) ▶ Post task execution logs
```

Query Query History

```
1 SELECT * FROM public.merged_data
2 ORDER BY id ASC
```

Data Output Messages Notifications

	id	track_id	artists	album_name	track_name	popularity	explicit	danceability	energy	track_genre
	[PK] integer	character varying (255)	text	character varying (255)	text	integer	boolean	double precision	double precision	character varying (255)
1	0	5Su0ikwiRyPMVoQJUGSV	Gen Hoshino	Comedy	Comedy	73	false	0.676	0.461	Instrumental
2	1	4qPNDBW13p13qLCOKI3A	Ben Woodward	Ghost (Acoustic)	Ghost - Acoustic	55	false	0.42	0.166	Instrumental
3	2	1UBS97x7JYXzME8Gcck5b	Ingrid Michaelson,ZAYN	To Begin Again	To Begin Again	57	false	0.438	0.359	Instrumental
4	3	6ftx3CG4xtIEq7opyGyx	Kina Grannis	Crazy Rich Asians (Original Motion Picture Soundtrack)	Can't Help Falling In Love	71	false	0.266	0.0596	Instrumental
5	4	5vLSffimlIP26QGSvNc2K	Chord Overstreet	Hold On	Hold On	82	false	0.618	0.443	Instrumental
6	5	01MVOi9K1VTNFBU9I7dc	Tyrone Wells	Days I Will Remember	Days I Will Remember	58	false	0.688	0.481	Instrumental
7	6	6Vc5wAMmXgdKIAM7WUeL...	A Great Big World,Christ...	Is There Anybody Out There?	Say Something	74	false	0.407	0.147	Instrumental
8	7	1EzEOXmMH3G43AXT1y7...	Jason Mraz	We Sing. We Dance. We Steal Things.	I'm Yours	80	false	0.703	0.444	Instrumental
9	8	0lktbUcnAGrvD03AWn23Q8	Jason Mraz,Colibie Calliat	We Sing. We Dance. We Steal Things.	Lucky	74	false	0.625	0.414	Instrumental
10	9	7k9GuJYlp2AaqokEdwEw2	Ross Copperman	Hunger	Hunger	56	false	0.442	0.632	Instrumental
11	10	4mzP5mHRvGxhdgdAH7...	Zack Tabudlo	Episode	Give Me Your Forever	74	false	0.627	0.363	Instrumental
12	11	5wF4eQ8QjVL5IAE9RyI	Jason Mraz	Love Is a Four Letter Word	I Wont Give Up	69	false	0.483	0.303	Instrumental
13	12	4ptDJBj35d7pQfNteBwp	Dan Berk	Solo	Solo	52	false	0.489	0.314	Instrumental
14	13	0X9MxHR1rTKEHdjp9F200	Anna Hamilton	Bad Liar	Bad Liar	62	false	0.691	0.234	Instrumental
15	14	4LbWBNK8Z2Rh29jzqgD3	Chord Overstreet,Deepend	Hold On (Remix)	Hold On - Remix	56	false	0.755	0.78	Instrumental
16	15	1KH0q8Nk90xmGjdX55NIG	Landon Pigg	The Boy Who Never	Falling in Love at a Coffee Shop	58	false	0.489	0.561	Instrumental
17	16	6xkGqzTjxUldI14qUezm	Andrew Foy,Renee Foy	ily (i love you baby)	ily (i love you baby)	56	false	0.706	0.112	Instrumental
18	17	4YoIgmcoNyatIsecaH0D	Andrew Foy,Renee Foy	At My Worst	At My Worst	54	false	0.795	0.0841	Instrumental
19	18	6CqNoABFJ4Q4id4EtbXIC	Boyce Avenue,Bea Miller	Cover Sessions, Vol. 4	Photograph	67	false	0.717	0.32	Instrumental
20	19	21QJcwLbYD4Yd16i8Gz9IP	Boyce Avenue,Jennel Gar...	Cover Sessions, Vol. 3	Demons	63	false	0.678	0.351	Instrumental
21	20	6D33wCKzWNEgOovgeVJ7r	Jason Mraz	Mellow Adult Pop	Bella Luna	1	false	0.755	0.454	Instrumental

Total rows: 1000 of 81941 Query complete 00:00:03.238 Ln 1, Col 1

duration_category character varying (255)	popularity_category character varying (255)	track_mood character varying (255)	live_performance boolean	title character varying (255)	category character varying (255)	is_nominated boolean
Average	High Popularity	Happy	false	Not applicable	Not applicable	false
Short	Average Popularity	Sad	false	Not applicable	Not applicable	false
Average	Average Popularity	Sad	false	Not applicable	Not applicable	false
Average	High Popularity	Sad	false	Not applicable	Not applicable	false
Average	High Popularity	Sad	false	Not applicable	Not applicable	false
Average	Average Popularity	Happy	false	Not applicable	Not applicable	false
Average	High Popularity	Sad	false	57th Annual GRAMMY Awards (2014)	Best Pop Duo/Group Performance	true
Average	High Popularity	Happy	false	Not applicable	Not applicable	false
Average	High Popularity	Happy	false	52nd Annual GRAMMY Awards (200...	Best Pop Collaboration With Vocals	true
Average	Average Popularity	Sad	false	Not applicable	Not applicable	false
Average	High Popularity	Happy	false	Not applicable	Not applicable	false
Average	Average Popularity	Sad	false	Not applicable	Not applicable	false
Average	Average Popularity	Happy	false	44th Annual GRAMMY Awards (2001)	Best Rock Gospel Album	true
Average	Average Popularity	Sad	false	Not applicable	Not applicable	false
Average	Average Popularity	Neutral	false	Not applicable	Not applicable	false
Average	Average Popularity	Sad	false	Not applicable	Not applicable	false
Short	Average Popularity	Neutral	false	Not applicable	Not applicable	false
Average	Average Popularity	Happy	false	Not applicable	Not applicable	false
Average	Average Popularity	Neutral	false	Not applicable	Not applicable	false
Average	Average Popularity	Neutral	false	Not applicable	Not applicable	false
Long	Low Popularity	Neutral	false	Not applicable	Not applicable	false

Storing the data

It's important to note the configuration files and credentials required by this module before starting the Airflow process:

- *credentials/client_secrets.json*
- *credentials/saved_credentials.json*
- *env/settings.yaml*

In addition, you must specify in the .env file the locations of these files with their respective absolute paths and the Google Drive ID of the folder in which you want to insert the file.

```
# Google Drive variables

CLIENT_SECRETS_PATH = "/home/mitgar14/ETL/etl-workshop-2/credentials/client_secrets.json"
SETTINGS_PATH = "/home/mitgar14/ETL/etl-workshop-2/env/settings.yaml"
SAVED_CREDENTIALS_PATH = "/home/mitgar14/ETL/etl-workshop-2/credentials/saved_credentials.json"

FOLDER_ID = 1x3tS43kSxC2oKhq7xCiJFzXqGerGvcy7
```

```
In [19]: storing_merged_data("merged_data", merged_df)
```

```
23/09/2024 09:12:55 PM Starting Google Drive authentication process.
23/09/2024 09:12:55 PM access_token is expired. Now: 2024-09-24 02:12:55.993523, token_expiry: 2024-09-20 20:52:25
23/09/2024 09:12:55 PM Access token expired, refreshing token.
23/09/2024 09:12:56 PM Refreshing access_token
23/09/2024 09:12:56 PM Google Drive authentication completed successfully.
23/09/2024 09:12:56 PM Storing merged_data on Google Drive.
23/09/2024 09:13:10 PM File merged_data uploaded successfully.
```

workshop2_dag Run 2024-09-23, 00:00:00 UTC / data_storing

Clear task Mark state as... Filter DAG by task

Details Graph Gantt Code Event Log Logs XCom Task Duration

All Levels All File Sources

DESKTOP-HW474E...

*** Found local files:

*** /home/milgr14/ETL/etl-workshop-2/airflow/logs/dag_id=workshop2_dag/run_id=scheduled_2024-09-23T00:00:00/task_id=data_storing/attempt=1.log

[2024-09-24, 02:28:00 UTC] (local_task_id=run_id.py:123) ▶ Pre task execution logs

[2024-09-24, 02:28:00 UTC] (store.py:37) INFO - Starting Google Drive authentication process.

[2024-09-24, 02:28:00 UTC] (client.py:649) INFO - access_token is expired. Now: 2024-09-24 02:28:00.777931, token_expiry: 2024-09-22 21:41:31

[2024-09-24, 02:28:00 UTC] (store.py:145) INFO - Access token expired, refreshing token.

[2024-09-24, 02:28:00 UTC] (client.py:777) INFO - Refreshing access_token

[2024-09-24, 02:28:01 UTC] (store.py:58) INFO - Google Drive authentication completed successfully.

[2024-09-24, 02:28:01 UTC] (store.py:162) INFO - Storing merged_data on Google Drive.

[2024-09-24, 02:28:15 UTC] (store.py:86) INFO - File merged_data uploaded successfully.

[2024-09-24, 02:28:15 UTC] (python.py:240) INFO - Done. Returned value was: None

[2024-09-24, 02:28:15 UTC] (taskinstance.py:340) ▶ Post task execution logs

https://drive.google.com/drive/u/1/folders/1x3tS43kSxC2oKhq7xCUFzXqGerGvcy7

Buscar en Drive

Mi unidad > ETL - Workshop #2

Tipo Personas Modificado

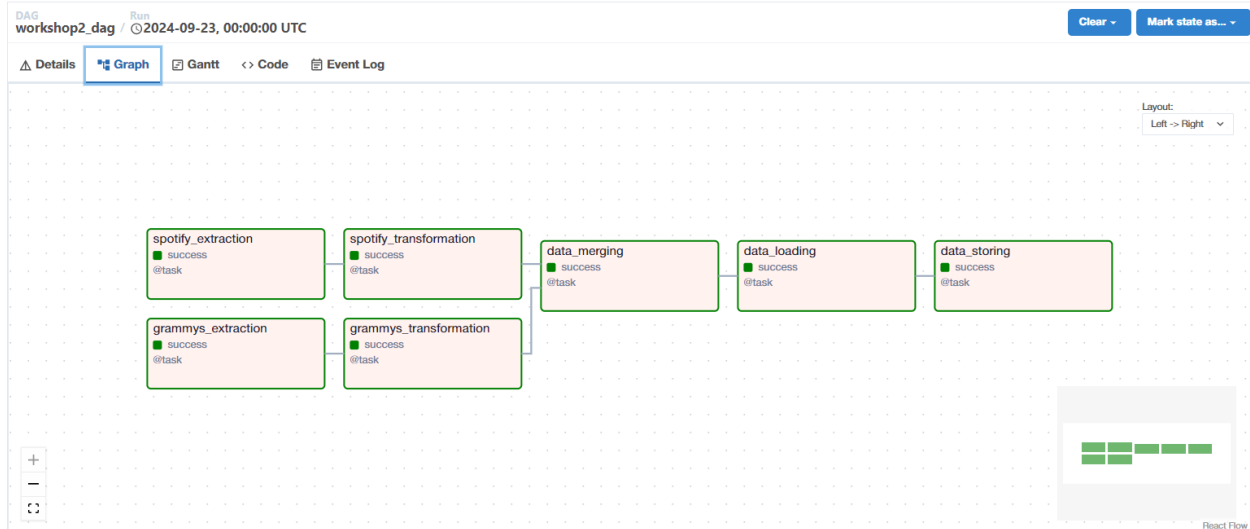
¡Nuevo! Combinaciones de teclas Las combinaciones de teclas de Drive se han actualizado para que puedas navegar escribiendo las primeras letras

Nombre	Propietario	Última modificación	Tamaño de
merged_data	yo	21:13 yo	15,3 MB

merged_data																	Abrir con	Compartir		
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q			
1	id	track_id	artists	album_name	track_name	popularity	explicit	danceability	energy	track_genre	duration_min	duration_category	popularity_category	track_mood	live_performance	title	category			
2	0	55U0nWfRyPMY0iQ	Gen Hoshino	Comedy	Comedy	73	FALSE	0.676	0.461	Instrumental	3	Average	High Popularity	Happy	FALSE	Not applicable	Not applicable			
3	1	4qPNDWfW3p193qU	Ben Woodward	Ghost (Acoustic)	Ghost - Acoustic	55	FALSE	0.42	0.166	Instrumental	2	Short	Average Popularity	Sad	FALSE	Not applicable	Not applicable			
4	2	1UB5r0f7YXzMEEC	Ingrid Michaelson	Zi To Begin Again	Zi To Begin Again	57	FALSE	0.438	0.359	Instrumental	3	Average	Average Popularity	Sad	FALSE	Not applicable	Not applicable			
5	3	6Hq3OG4HT	Big Top Kina Gramis	Crazy Rich Asians	Can't Help Falling In	71	FALSE	0.266	0.0596	Instrumental	3	Average	High Popularity	Sad	FALSE	Not applicable	Not applicable			
6	4	5yLS8mIP26G5V	Chord Overstreet	Hold On	Hold On	62	FALSE	0.618	0.443	Instrumental	3	Average	High Popularity	Sad	FALSE	Not applicable	Not applicable			
7	5	01MVO9Kv7NFB8	Tyrene Wells	Days 1 Will Remember	Days 1 Will Remember	58	FALSE	0.688	0.481	Instrumental	3	Average	Average Popularity	Happy	FALSE	Not applicable	Not applicable			
8	6	6V5saAMnXG8AM	A Great Big World	Cl Is There Anybody Out	Say Something	74	FALSE	0.407	0.147	Instrumental	3	Average	High Popularity	Sad	FALSE	Not applicable	Not applicable			
9	7	1EeEOXmH3G43J	Jason Mraz	We Sing, We Dance, I'm Yours	We Sing, We Dance, I'm Yours	80	FALSE	0.703	0.444	Instrumental	4	Average	High Popularity	Happy	FALSE	Not applicable	Not applicable			
10	8	0NBKcAqG0G3AJ	Jason Mraz; Colbie C	We Sing, We Dance, Lucky	We Sing, We Dance, Lucky	74	FALSE	0.625	0.414	Instrumental	3	Average	High Popularity	Happy	FALSE	Not applicable	Not applicable			
11	9	7NBGU7Lj2AZpYfE	Ross Coppleman	Hunger	Hunger	56	FALSE	0.442	0.632	Instrumental	3	Average	Average Popularity	Sad	FALSE	Not applicable	Not applicable			
12	10	4ndP5eIRfWdabD	Zack Tabudlo	Epicade	Give Me Your Power	74	FALSE	0.627	0.363	Instrumental	4	Average	High Popularity	Happy	FALSE	Not applicable	Not applicable			
13	11	5uf4e3BqJVL5AE	Jason Mraz	Love Is a Four Letter	I Won't Give Up	69	FALSE	0.483	0.303	Instrumental	4	Average	Average Popularity	Sad	FALSE	Not applicable	Not applicable			
14	12	4yCJcX557yJdWd	Dan Berk	Solo	Solo	52	FALSE	0.489	0.314	Instrumental	3	Average	Average Popularity	Happy	FALSE	Not applicable	Not applicable			
15	13	0Q9M4WV7T6E5G	Anne Hathorn	Bad Liar	Bad Liar	62	FALSE	0.691	0.234	Instrumental	4	Average	Average Popularity	Sad	FALSE	Not applicable	Not applicable			
16	14	4LW8BmL2RvdjG	Chord Overstreet	Hold On (Remix)	Hold On - Remix	56	FALSE	0.755	0.79	Instrumental	3	Average	Average Popularity	Neutral	FALSE	Not applicable	Not applicable			
17	15	1H4q5mK29GdGd	Landon Pigg	The Boy Who Never	Falling in Love at a	56	FALSE	0.489	0.561	Instrumental	4	Average	Average Popularity	Sad	FALSE	Not applicable	Not applicable			
18	16	68xGpDj3SLU14	Andrew Foy	Renée F Ily (I love you baby)	Ily (I love you baby)	56	FALSE	0.706	0.112	Instrumental	2	Short	Average Popularity	Neutral	FALSE	Not applicable	Not applicable			
19	17	4Y0TgmcoNfYt2	Andrew Foy	Renée F At My Worst	At My Worst	54	FALSE	0.795	0.0841	Instrumental	2	Average	Average Popularity	Happy	FALSE	Not applicable	Not applicable			
20	18	6C9tA6fJ4Q4H6E	Boyce Avenue	Bee F Cover Sessions, Vol. Photograph	Photograph	67	FALSE	0.717	0.32	Instrumental	4	Average	Average Popularity	Neutral	FALSE	Not applicable	Not applicable			
21	19	21U3cXJL7DYH6I	Boyce Avenue	Jenni Cover Sessions, Vol. Photograph	Photograph	63	FALSE	0.676	0.351	Instrumental	2	Average	Average Popularity	Neutral	FALSE	Not applicable	Not applicable			
22	20	6D3cXKwMNGJoc	Jason Mraz	Mellow Adult Pop	Bella Luna	1	FALSE	0.755	0.054	Instrumental	5	Long	Low Popularity	Neutral	FALSE	Not applicable	Not applicable			
23	21	5RCZ0uZ2q5m5A	Jason Mraz	Holly Jolly Christmas	Winter Wonderland	0	FALSE	0.62	0.309	Instrumental	2	Short	Low Popularity	Happy	FALSE	Not applicable	Not applicable			
24	22	0uK8y8tP5f5aM	Jason Mraz	Feeling Good - Adult	If It Kills Me	0	FALSE	0.633	0.429	Instrumental	4	Average	Low Popularity	Neutral	FALSE	Not applicable	Not applicable			
25	23	1mSjC2R5E528y7I	Chord Overstreet	Christmas Country	St All I Want For Christmas	0	FALSE	0.593	0.455	Instrumental	3	Average	Low Popularity	Neutral	FALSE	Not applicable	Not applicable			
26	24	3H56v7u2GD468I	Brandi Carlile	Sam & Human - Best Adult	I Party of One	0	FALSE	0.296	0.206	Instrumental	4	Average	Low Popularity	Sad	FALSE	Not applicable	Not applicable			
27	25	62EwQLD6p4DzU	Brandi Carlile	Chill Christmas	Don't Lonely This Christmas	0	FALSE	0.409	0.153	Instrumental	4	Average	Low Popularity	Sad	FALSE	Not applicable	Not applicable			
28	26	3v0t8c2q8JDLZw	Brandi Carlile	rainy day indie	Throwing Good After	0	FALSE	0.501	0.0952	Instrumental	4	Average	Low Popularity	Neutral	FALSE	Not applicable	Not applicable			
29	27	6g8vX3a2f5pPbC	Eddie Vedder	Mega Hits	Autumn	0	FALSE	0.474	0.519	Instrumental	5	Long	Low Popularity	Neutral	FALSE	Not applicable	Not applicable			
30	28	1WVYV0bm5v73I	Brandi Carlile	Mellow Adult Pop	When You're Wrong	0	FALSE	0.565	0.301	Instrumental	4	Average	Low Popularity	Sad	FALSE	Not applicable	Not applicable			
31	29	64P4yJy5D9WwW	Brandi Carlile	Lucid Country	Car Hits	0	FALSE	0.565	0.466	Instrumental	3	Average	Low Popularity	Happy	FALSE	Not applicable	Not applicable			
32	30	6J0feyFyJy7Y	Brandi Carlile	Pinet Country	Speak Your Mind	0	FALSE	0.476	0.666	Instrumental	3	Average	Low Popularity	Neutral	FALSE	Not applicable	Not applicable			
33	31	7xT4Tou2Dn8a5E	Highland Peak	Trampoline (Acoustic)	Trampoline - Acoustic	46	FALSE	0.596	0.2	Instrumental	3	Average	Average Popularity	Happy	FALSE	Not applicable	Not applicable			
34	32	3m4M4fY2u2G5D	Moonshine Hala	Documentary	漫遊時光	61	FALSE	0.373	0.914	Instrumental	3	Average	Average Popularity	Neutral	FALSE	Not applicable	Not applicable			
35	33	2D3CvAD4uW6Wd	Andrew Belle	Black Bear	Pieces	60	FALSE	0.444	0.652	Instrumental	4	Average	Average Popularity	Sad	FALSE	Not applicable	Not applicable			
36	34	5XcU4mUd7T4W6Z	Ron Pope	The Bedroom Demo	A Drop in the Ocean	68	FALSE	0.331	0.393	Instrumental	3	Average	Average Popularity	Sad	FALSE	Not applicable	Not applicable			
37	35	24YF3V5V525W8K5	Adam Christopher	So Far Away	Acoustic	52	FALSE	0.331	0.331	Instrumental	2	Average	Average Popularity	Neutral	FALSE	Not applicable	Not applicable			

Managing the data pipeline with Airflow

In this workshop it's used the DAG workshop2_dag. The DAG code was realized using the Taskflow API, therefore, the tasks and context statement of our DAG is composed of decorators.



Analyzing the composition of the DAG

The image in this section belongs to the DAG declaration of this workshop. As can be seen, we use decorators to declare that the following functions are our tasks.

```
32 def workshop2_dag():
33     @task
34     def spotify_transformation(raw_df):
35         return transform_spotify(raw_df)
36
37     spotify_data = spotify_transformation(spotify_raw_data)
38
39     @task
40     def grammys_transformation(raw_df):
41         return transform_grammys(raw_df)
42
43     grammys_data = grammys_transformation(grammys_raw_data)
44
45     @task
46     def data_merging(spotify_df, grammys_df):
47         return merge_data(spotify_df, grammys_df)
48
49     df = data_merging(spotify_data, grammys_data)
50
51     @task
52     def data_loading(df):
53         return load_data(df)
54
55     data_load = data_loading(df)
56
57     @task
58     def data_storing(df):
59         return store_data(df)
60
61     data_store = data_storing(data_load)
62
63     workshop2_dag = workshop2_dag()
```

```
# Creating tasks functions
# -----

def extract_spotify():
    try:
        df = extracting_spotify_data("./data/spotify_dataset.csv")
        return df.to_json(orient="records")
    except Exception as e:
        logging.error(f"Error extracting data: {e}")

def extract_grammys():
    try:
        df = extracting_grammys_data()
        return df.to_json(orient="records")
    except Exception as e:
        logging.error(f"Error extracting data: {e}")

def transform_spotify(df):
    try:
        json_df = json.loads(df)

        raw_df = pd.DataFrame(json_df)
        df = transforming_spotify_data(raw_df)

        return df.to_json(orient="records")
    except Exception as e:
        logging.error(f"Error transforming data: {e}")

def transform_grammys(df):
    try:
        json_df = json.loads(df)

        raw_df = pd.DataFrame(json_df)
        df = transforming_grammys_data(raw_df)

        return df.to_json(orient="records")
    except Exception as e:
        logging.error(f"Error transforming data: {e}")
```

However, our DAG functions call functions that are not even part of that file.

The functions that are called in that file belong to the `etl` module, which is part of *tasks*.

One important thing to note is that in order to receive and send data between tasks, the functions will receive and convert the data to JSON, one of the few data formats allowed in Airflow.

Another thing to note is that the call chain does not stop here: these functions call more functions. These other functions are part of the various packages and modules that make up

`src`.

Visualizing the Data

The Power BI dashboard provides a comprehensive visual representation of data related to songs, genres, and artists on Spotify. It enables users to quickly identify patterns and gain deeper insights into the music trends on Spotify.

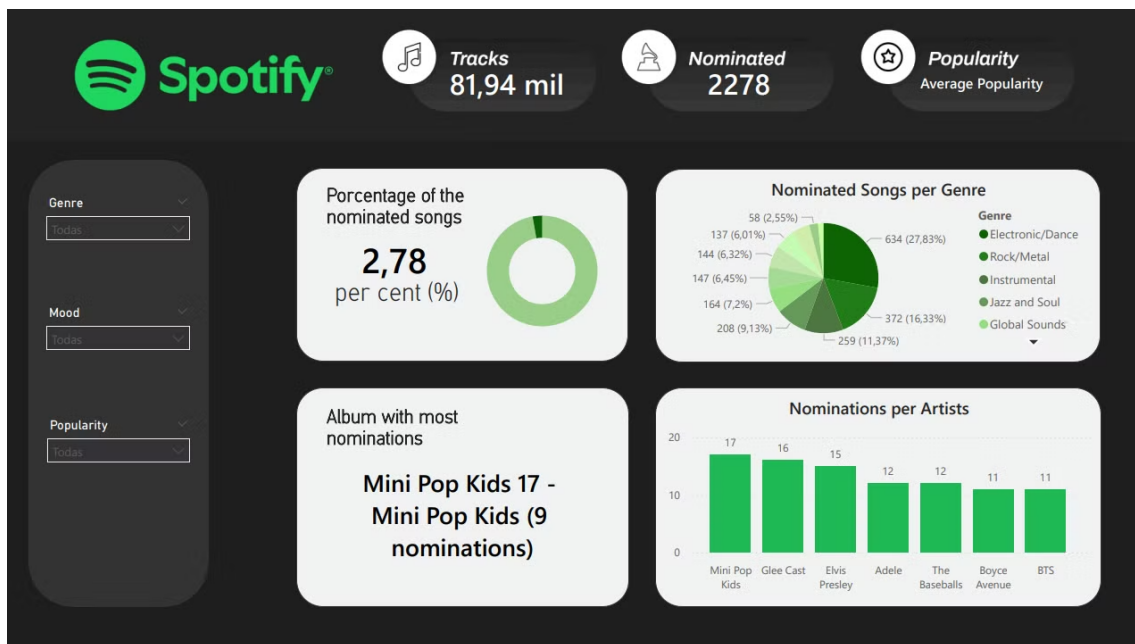
Below is a detailed breakdown of each page of the dashboard and its key elements.

Page 1: Nominations

This first page of the dashboard gives an overview of song popularity and the number of nominations received by artists and albums. Key visualizations include:

- **Average Popularity:** A KPI visual showing the average popularity score across all songs in the dataset (e.g., 81.94). This provides a quick snapshot of how well songs are performing overall.

- **Nominations per Artist:** A bar chart that displays the number of nominations received by different artists. For example, artists like BTS, Adele, and Elvis Presley have notable mentions, allowing users to see which artists are leading in nominations.
- **Nominated Songs per Genre:** A bar chart that shows the number of nominated songs in each genre. Genres like Electronic/Dance, Rock/Metal, and Jazz and Soul are prominently featured, giving insight into which genres dominate nominations.
- **Album with Most Nominations:** A card visual showing the album with the highest number of nominations, such as *Mini Pop Kids 17*, which has 9 nominations, making it the top-nominated album in this dataset.

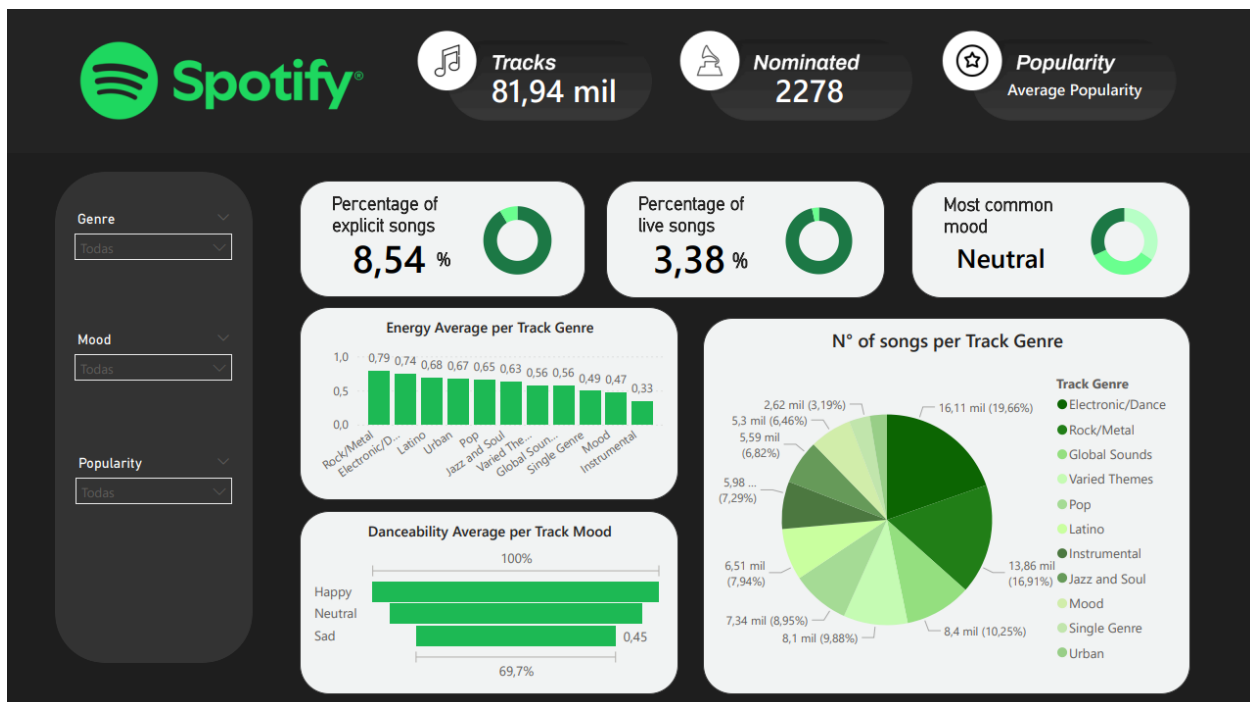


Page 2: Song Characteristics

The second page dives deeper into the specific characteristics of songs, such as energy levels, explicit content, and live performances. Key metrics and visualizations include:

- **Percentage of Explicit Songs:** A KPI that displays the percentage of songs marked as explicit, providing a clear view of how much explicit content is present in the dataset (e.g., 8.54%).

- **Percentage of Live Songs:** Another KPI showing the percentage of live recordings within the dataset (e.g., 3.38%). This helps differentiate between studio recordings and live performances.
- **Most Common Mood:** A card visual showing the most common mood found in the tracks. For instance, "Neutral" may be highlighted as the predominant mood, giving an idea of the emotional tone of the majority of songs.
- **Energy Average per Track Genre:** A bar chart that compares the average energy level of tracks across different genres. Genres like Rock/Metal, Electronic/Dance, and Jazz and Soul are shown with varying energy averages, giving insights into the intensity of songs in each category.
- **Danceability Average per Track Mood:** A bar chart that displays the average danceability score based on the mood of the song. Moods such as Happy, Neutral, and Sad are analyzed, allowing users to understand how different emotional tones impact the danceability of tracks.
- **Number of Songs per Track Genre:** A chart that shows the total number of songs within each genre, providing an overview of the distribution of genres in the dataset. Genres like Pop, Electronic/Dance, and Rock/Metal may have the highest representation.



Conclusions

1. **Successful Data Integration:** The workshop demonstrated effective merging of Spotify and Grammy datasets, highlighting the importance of data cleaning and transformation in preparing datasets for analysis.
2. **Airflow Pipeline Efficiency:** The implementation of Apache Airflow showcased the power of automated, scalable data pipelines, essential for handling complex ETL processes in real-world scenarios.
3. **Insightful Visualizations:** The Power BI dashboard provided valuable insights into music trends, artist popularity, and genre distribution, emphasizing the importance of data visualization in extracting meaningful information.
4. **Cross-Platform Data Management:** The workshop illustrated the seamless integration of various tools (Python, PostgreSQL, Airflow, Power BI) in a comprehensive data analysis workflow, reflecting real-world data engineering practices.
6. **Industry Relevance:** By analyzing music industry data, the workshop provided practical experience in handling and interpreting real-world datasets, preparing participants for similar challenges in professional settings.