Spotify & YouTube Music

Data Cleaning:

Q1. Identify and Handle Missing Values:

- Examine the dataset for any missing values. Which columns contain null values?
- How should missing values in the Views and Likes columns be handled? Should they be filled with a default value, removed, or handled in another way? Justify your approach.

Solution:

Examine the missing values	
Column	**Missing Values**
STREAM	610
key	2
valence	2
liveness	2
speechiness	2
loudness	2
tempo	2
danceability	2
instrumentalness	2
acousticness	2
ENERGY	2
DURATION_MS	2
CHANNEL	491
youtube_info	491
views	2484
LIKES	2685
COMMENTS	593
description	911
OFFICIAL_VIDEO	491
licensed	491

Approach for Handing Missing Values:

1. **Views Column**
   Introduction: represents the number of times a particular video or song has been viewed, typically on a platform like YouTube. This is a key metric that helps gauge the popularity and reach of the content.
   Fill Views with `0` assuming no views yet: Filling with `0` allows you to retain all rows in the dataset, ensuring no data is lost.

Justification: This preserves the row in the dataset and avoids removing potentially valuable contextual data from other columns. Default values are simple to interpret in analysis.

Why Choose 0: If missing values indicate a lack of engagement e.g., videos with no views yet For basic statistical summaries where missing values would skew calculations.

2. Likes

Introduction: On "likes" refer to a feature that allows users to express their approval or enjoyment of a video. Here's a breakdown of what "likes" mean on the platform:

Fill Likes with 0 **assuming no likes yet**: Filling with 0 allows you to retain all rows in the dataset, ensuring no data is lost.

Justification: This preserves the row in the dataset and avoids removing potentially valuable contextual data from other columns. Default values are simple to interpret in analysis.

Why Choose 0: If missing values indicate a lack of engagement e.g., videos with no likes yet For basic statistical summaries where missing values would skew calculations.

Summary: In both cases, filling missing values with 0 helps preserve the dataset's integrity and ensures accurate analysis

---

Q2.Fix Irregularities in Merged Columns:

- The Spotify_Info and Youtube_Info columns contain merged data separated by delimiters. Split these columns back into their original components. What are the original components, and how can you ensure that the split data is clean and accurate?

- After splitting, remove any unnecessary delimiters or prefixes/suffixes that do not belong.

Solution:

Splitting the Columns:

1. Spotify_Info

Introduction : Spotify_Info is a leading digital music streaming service that provides users with access to a vast library of songs, podcasts, and other audio content.

Splitting the Columns by Delimiters: choosing the **pipe (|)** delimiter for splitting merged data is a strategic decision based on several practical and technical factors.

Original Components: These columns may contain:Spotify_Info: A Spotify URL (e.g., artist or track links)

Why Choose Delimiters: Choosing **delimiters** for splitting merged data in the **Spotify_Info** column (or any other column containing combined data) is a key part of the data cleaning and preprocessing process.

2. Youtube_Info:

Introduction : YouTube is one of the largest and most popular video-sharing platforms in the world, enabling users to upload, share, and view videos across a wide range of genres and topics.

Splitting the Columns by Number of Characters: Fixed-Length Data If the first part of the merged data (e.g., a URL) always has the same number of characters, splitting by a fixed character limit can reliably separate components.

Original Components: These columns may contain: Youtube_Info: A YouTube URL (e.g., video links).

Why Choose Fixed-Length : Using a fixed **number of characters** to split the data can be helpful when the data follows a **consistent pattern or format** where components have predictable lengths.

Summary: In both cases, filling **Delimiters and by Number of Characters** helps preserve the dataset's integrity and ensures accurate analysis

---

Q3. Correct Case Sensitivity and Naming Conventions:

- The column names have inconsistent case sensitivity (some are uppercase, others lowercase). Standardize all column names to follow a consistent format (e.g., all lowercase with underscores).
- Fix any data entries where case sensitivity might affect consistency (e.g., artist names or track titles). Ensure that the Artist and Track columns are formatted consistently.

Solution:

Standardizing Column Names: Converted all column names to lowercase to maintain consistency. Replaced spaces and special characters (e.g. : and .)with underscores to follow standard naming conventions for easy referencing.

Standardizing "Artist" and "Track" Data: Converted all entries in the "Artist" and "Track" columns to title case (e.g., gorillaz → Gorillaz). This makes the data consistent and easier to interpret while aligning with typical naming conventions for proper nouns.

- Scanned the dataset for anomalies in the "Artist" and "Track" columns.
- Verified column name changes by cross-referencing before and after outputs.

Method Used:

- Select the column (artist or track).
- Use the **Transform** tab → **Format** → **Capitalize Each Word**.

Why These Methods

Standardization : Lowercase column names with underscores are a widely accepted convention in data analytics and programming, promoting readability and system compatibility.

Capitalize Each Word : Title-cased "Artist" and "Track" entries improve readability and ensure a professional appearance of the dataset.

Q4. Remove or Handle Irrelevant Columns:

- Identify and remove any irrelevant or randomly generated columns that do not provide useful information for analysis. Which columns should be removed, and why?
- If any random data exists in relevant columns, clean or remove those entries.

Solution:

Remove Irrelevant Columns

- Select **random_column_1,** and **random_column_2.**
- Right-click and choose **Remove Columns**.

Why Remove : random_column_1, and random_column_2

These columns do not provide actionable insights and only add noise to the dataset.

Handle Random Data in Relevant Columns

1. For textual columns:

- Use **Filter** to identify entries like “nan”, ”Blank”, ”N/A”, or “UNKNOWN” and remove or replace them.
- Use **Replace Values** to handle placeholder entries.

2. For numeric columns:

- Identify outliers or placeholder values by sorting or applying filters.
- Replace with meaningful values or mark as missing (null).

Why Clean Random Data?

- Random or placeholder entries in relevant columns could lead to incorrect analysis, misleading results, or errors in downstream processes.

Q5.Handle Inconsistent Data Types:

- Some columns that should be numeric (e.g., Danceability, Energy) are stored as text. Convert these columns back to numeric format. What steps would you take to identify and fix any issues that arise during this conversion?
- Ensure that all numeric columns are in the correct format and handle any non-numeric values or anomalies.

Solution:

Steps to Handle Inconsistent Data

Convert Text Columns to Numeric:

- Select the problematic column (e.g., Views Danceability or Energy).
- From the **Transform** tab, choose **Data Type** and select **Decimal Number** as appropriate.

Handle Conversion Errors:

- If the conversion results in errors, click on the small error icon that appears in the column to see a list of problematic rows.
- Use the **Replace Errors** option to handle these issues:
    - Replace invalid values with `0`, or a default value.

Handle Missing or Anomalous Values:

- Replace null values with a meaningful default using .

Q6.Address and Fix Invalid Data Entries:

- Check the Views column for any entries labeled as "invalid_data" or any other incorrect values. Replace these entries and justify your method.
- Ensure that all values in the Album column are correctly labeled and that there are no numeric entries or irrelevant data.

Solution:

Replace invalid entries with: replacing with null is a simpler approach.

Why Use Null to Replace Invalid Entries: Replacing with null ensures we don't treat invalid data as actual views. This prevents skewing the results of analysis or calculations (such as averages or totals). By using null, we signal that the data is missing or erroneous, not just absent

Justification for the Chosen Methods: Using `null` is a valid option if you prefer to track missing data instead of assigning a default label.

Album : Album column is clean and correctly labeled.

---

7. Check for and Remove Duplicate Rows:

- Identify and remove any duplicate rows in the dataset. How can you ensure that the remaining data is unique and accurate?

Solution:

Check for Unique Values Across Key Columns

- **Identify Key Columns for Uniqueness**:
  If your dataset has multiple columns, you need to identify which ones should be unique, such as ID, Album, Track, or any other key identifier.

How to Check:

1. Open the **Power Query Editor**.
2. Select **Transform** > **Group By**.
3. Group by the columns that should have unique combinations (e.g., ID,*Album* and *Track*).
4. After grouping, if there are no aggregations needed, the count of rows should match the number of unique records.

Why These Steps Matter Even When There Are No Duplicates:

- **Ensures Data Quality**: Even with no duplicates, the data might still have inconsistencies, missing values, or other issues that could affect analysis. It's important to ensure data integrity and consistency before proceeding with any analysis.

Q8. Reorder and Rename Columns for Clarity:

- Reorder the columns in a logical sequence to improve the dataset's readability and usability. What order makes the most sense for this dataset?
- Rename columns where necessary to ensure that their names clearly reflect the data they contain.

Solution:

Reordering Columns: (Columns that uniquely identify a record, like unnamed: 0 = ID or names)
Example: Track , Album , Track Name, Album Name, Artist Name.

Renaming Columns:

1. unnamed: 0 = ID
2. youtube_info = youtube_ track_link
3. spotify_info = spotify_song_link
4. Track = Track Name
5. Album = Album Name

THANK YOU