Spotify & Youtube Data Cleaning Project

1. Identify and Handle Missing Values:

- Examine the dataset for any missing values. Which contains 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.

Examine the dataset for any missing values. Which columns contains null values?

The dataset contains missing values in the following columns, along with the count of null values in each:

Columns	**Number of missing values**
- Key:	2 missing values
- Licensed:	491 missing values
- Valence:	2 missing values
- Liveness:	2 missing values
- Speechiness:	2 missing values
- Loudness:	2 missing values
- Official_Video:	491 missing values
- Tempo:	2 missing values
- Views:	2484 missing values
- Danceability:	2 missing values
- Duration_MS:	2 missing values
- Instrumentalness:	2 missing values
- Channel:	491 missing values
- Youtube_info:	491 missing values
- Comments:	593 missing values
- Description:	911 missing values
- Likes:	2685 missing values
- Stream:	610 missing values
- Acousticness:	2 missing values
- Energy:	2 missing 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.

1. Fill with Default Values (e.g., 0 or Mean/Median)

When to Use:

- If the missing values indicate no views or likes, filling with 0 can be appropriate.
- If you assume missing values are random and unintentional, using the mean or median preserves the overall distribution of the data.

Pros:

- Retains all rows, ensuring no loss of other potentially valuable data.
- Using median (less sensitive to outliers) or mean maintains statistical properties of the dataset.

Cons:

- Introducing a default value may distort analysis if the reason for missing values is non-random (e.g., new or untracked videos).

2. Remove Rows with Missing Values

When to Use:

- If a substantial portion of the analysis relies on views and likes, and missing data represents a small fraction.
- If there's no reliable method to impute values, removing them avoids introducing noise.

Pros:

- Ensures clean and reliable data.
- Avoids assumptions about the missing values.

Cons:

- Loss of data can reduce the robustness of insights, especially if missing values are widespread (e.g., 2685 missing likes).

3. Predict Missing Values Using Other Columns (Advanced Imputation)

When to Use:

- If there are strong correlations between views/likes and other features (e.g., STREAM, DURATION_MS, or description).
- For high-value datasets where retaining accuracy is critical.

Pros:
- Preserves data and may provide reasonable approximations.
- Minimizes distortion caused by arbitrary filling.

Cons:
- Requires effort to model the relationships.
- Imputation may still introduce error.

Suggested Approach:

Initial Analysis: Check the distribution of views and likes. Are the missing values skewed toward certain types of entries (e.g., missing channels)? Look for correlations with other columns.

Practical Recommendation:
- Use median imputation for large datasets with a random pattern of missingness.
- Remove rows only if missing values are extensive and imputation isn't reliable.

Justification: Median is robust to outliers (common in views/likes), while retaining data maximizes analytical value. However, removal ensures data integrity for critical analyses.

2. 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.

Answer:

Data Cleaning and Transformation Summary:

- **Split youtube_info by Fixed Width:** The *youtube_info* column was split based on a fixed width, as the links maintained a consistent length throughout the column. This ensured that the link and track ID were properly separated.
- **Column Type Conversion:** The split components were converted into appropriate text data types to maintain uniformity and consistency.
- **Right-Most Split Refinement:** A right-most split operation was performed on the second part of the *youtube_info* data to clean and refine it further.
- **Column Merging:** After cleaning, relevant components were merged back to create the *Youtube_track_id* column. This ensured that the title and any additional information were correctly combined.
- **Renaming and Cleanup:** Columns were renamed for clarity, and unwanted details were removed from the *Youtube_track_id* column using a "text before delimiter" approach.
- **Clean and Accurate Data:** The *Youtube_track_id* column was refined to contain only essential track information, ensuring it was clean and readable by eliminating extra delimiters, prefixes, and suffixes.
- **Duplicate Removal:** Duplicates in the *Spotify_Info* column were removed to ensure the uniqueness of data entries.
- **Column Reordering:** Columns were reorganized for better clarity and logical structure within the dataset.
- **Delimiter-Based Splitting:** The *Spotify_Info* column was split using the correct delimiter (|), separating it into two distinct columns: *Spotify_Album_Link* and *Spotify_Track_Id.*
- **Data Type Conversion:** The newly split columns were converted into text data types, which is essential for fields containing URLs or track IDs.
- **Column Renaming:** The split columns were renamed to *Spotify_Album_Link* and *Spotify_Track_Id* to make them clear and self-explanatory.

3. 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.

Answer:

Data Cleaning and Transformation Summary:

- **TRACK Column:** Converted all text to uppercase to ensure uniformity, as the column had a mix of uppercase and lowercase values. **The ARTIST column** was already in uppercase, so no changes were needed.
- **YOUTUBE_TRACK_ID Column:** Applied title casing (capitalizing each word) to ensure consistency, as the column had a mix of uppercase and title-case values.
- **DESCRIPTION Column:** Left unchanged, as the mix of uppercase and title-case values seemed intentional. Uppercase phrases appeared to emphasize specific points, while most other values were in title case.
- **LICENSED and OFFICIAL_VIDEO Columns:** Standardized by converting all text to uppercase for consistency, as these columns contained single-word string values.

This ensured a clean and consistent format across all relevant columns while respecting intentional formatting choices.

4. 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.

Answer:

Data Cleaning and Transformation Summary:

Likely Irrelevant Columns:

- **Random or Placeholder Columns:** random_column_1 and random_column_2 were observed in an earlier preview and seemed to contain placeholder values like "RANDOM" or unrelated numerical data.
- **Empty or Low-Value Columns:** Any column with mostly null or missing values that do not contribute meaningfully to the analysis.
- **Duplicate or Redundant Columns:** Columns that repeat data already present in a more structured or useful form.

Reasons for Removal:

- Such columns do not provide meaningful insights or support the primary objectives of the dataset.
- Removing them helps streamline the dataset and improves clarity.

5. 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.

Answer:

Data Cleaning and Transformation Summary:

- Updated **DANCEABILITY** by replacing NaN with null to resolve the error.
- Fixed **ENERGY** column by substituting NaN with null.
- **Resolved column conversion** issue by addressing non-numeric values, enabling successful numeric transformation.

6. 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.

Answer:

Data Cleaning and Transformation Summary:

- Replaced invalid_data with null in the **VIEWS** column to resolve the error.
- Converted data type of **VIEWS** from Text to Whole Number.
- Trimmed leading and trailing spaces in **COMMENTS**, **LIKES**, and **VIEWS** columns.
- Substituted **NaN** with **null** in the **LIKES** column.
- Substituted **NaN** with null in the **VIEWS** column.
- Changed the data type of **LIKES** and **VIEWS** from **Text** to **Decimal** to accommodate both whole numbers and decimals while preserving data accuracy.

The **ALBUM** column seems generally well-labeled, but further inspection is necessary to identify any numeric entries or irrelevant data. Let's proceed to:

- Check for invalid data (e.g., numeric or irrelevant entries) in the ALBUM column.
- Fix any identified issues.

There are no invalid entries in the ALBUM column; all values are correctly labeled as strings. No numeric or irrelevant data was found.

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?

Answer:

Data Cleaning and Transformation Summary:

- Eliminated duplicates based on unique values in the **INDEX** column.
- Removed duplicates in specific columns, such as **SPOTIFY_INFO,** where applicable. For dependent and correlated columns, duplicates were retained to prevent data loss.

8. 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.

Answer:

Data Cleaning and Transformation Summary:

- Renamed the unnamed column to INDEX.
- Reorganized columns into a logical sequence to enhance dataset readability and usability, ensuring related and dependent columns are placed adjacent to each other.