1. 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:

In light of the contextual framework of the dataset, which pertains to music-related analytics and integrates insights from both Spotify and YouTube, it is important to address any instances of missing values to prevent the distortion of feedback that may influence corporate decision-making. The following presents a refined approach.

Approach to handle Missing Values:

1) "Views" Column:

    - Views represent how many times a track was played, making it one of the major metrics for engagement and popularity.

    - Approach: Replace missing values with the median of the "Views" column.

    - Steps taken:- In views column, first replace blank values with invalid_data because there is a already value named invalid_data in the column. After that I replaced invalid_data with null and calculated median of "Views" column which came out as 14529746 and replaced null values with median.

    - Why? It is replaced by median because the median is less sensitive to extreme values (e.g., viral videos with extremely high views). Making it more reliable choice for filling blanks in this column.

2) "Likes" Column:

    - Likes reflect user's positive feedback, a key indicator of how much listeners enjoy particular track.

- Approach: Replace missing values with the median of the “Likes” column.
- Why? Likes often exhibit bias, with some tracks receiving higher number of likes. Replacing it with median keeps balance and avoid distorting user sentiments analysis.

3) “Licensed” Column:

- License column represent if track is licensed or not.
- Approach: In licensed column replace null with True.
- Why? By analyzing null values and comparing it with rest of the table, it is found that some tracks are exclusive to Spotify only. As Spotify tracks are also licensed songs, so null values can be replaced with True.

4) “Official_Video” Column:

- The term "official video" typically refers to a video that is officially released or endorsed by the creator, organization, or rights holder, often as the primary or authorized representation of content, such as a music video, promotional clip, or product launch.
- Approach: In this column replace null value with False.
- Why? In “Official_Video” column blank values represent video is not available on Youtube and it is only available on spotify, as it is exclusive to Spotify, so replace null the value with False.

5) “Channel” Column:

- A YouTube channel is a user's profile for sharing and organizing videos, where viewers can subscribe for updates.
- Approach: In this column replace blank with “Not available on YouTube”.

- Why? Blank Values in “Channel” column are for those tracks which are exclusive to Spotify, replace blanks with “Not available on YouTube”.

6) Other columns with missing values:

   - “Streams” and “Comments” columns also contains missing values and they are also replaced with median values of those columns, because using the median maintains balance and prevents skewing user sentiment analysis.

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.

Solution:

Fixing irregularities in merged columns is crucial for data integrity and accurate analysis, as it ensures that each column contains only the relevant data and maintains the integrity of the dataset, making it easier to analyze and interpret.

1) “Spotify_Info” Column:

   - Spotify_Info refers to data related to a song's details on Spotify, such as the track title, artist name and album.

   - Approach: Sliptted Spotify_Info by using delimiter “|” which split it into Artist Page URL and Track ID for Spotify.

   - Why? Spliting this information helps in directly browsing the artist album and track. It create ease in access to desired outcome.

2) “YouTube_Info” Column:
    - YouTube Info refers to metadata about YouTube content, such as url of video, video titles and creators.

    - Approach: In YouTube_Info, each video url have specificaly 43 characters, we can extract the url of video from here using extract function by giving length of characters.

    - Why? URL is enough to search track video, removing excess info will benefit in the optimization.

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.

Solution:

Standardizing case sensitivity ensures consistency, prevents errors, avoids duplicate entries, and simplifies data processing and analysis.

1) Column Names:

    - Approach: Columns names can be renamed easily by double tapping the title of columns, use same pattern for all columns and instead of space use underscore.

    - Why? Standardizing case sensitivity is important for consistency, error reduction, efficiency, ease of integration and professionalism.

2) “Artist” Column:

- Artist Column gives the name of the artist of the song.
- Approach: Extract TEXT after delimiter of artist column because every entry starts with ARTIST_.
- Why? It will give the only name of artist which is more aesthetic and reduce data load for better optimization.

3) “Track” Column:

- Track column gives the name of the track.
- Approach: Extract TEXT before delimiter of artist column because every entry ends with _TRACK.
- Why? It will only provide the “Track” column more attractive appearance and reduce the data load for enhanced optimization.

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.

Solution:

1) Irrelevant Columns:

- By eliminating unnecessary and irrelevant columns, we reduce the size of the dataset, which can lead to reduced errors, cleaner data, faster load time and better performance

- Approach: Remove random_column_1 and random_column_2 from the dataset.

- Why? random_column_1 and random_column_2 are irrelevant to the dataset, unable the set any relation with other columns. Removing it will reduce the risk of errors in rest of dataset.

2) Relevant Columns:

- There can be some undisclosed columns which can be relevant to the given dataset, we have to look for them and try to establish relationship.

- Approach: Closely observe the data to look for relevance and “unnamed: 0” column is as such column.

- Why? “unnamed: 0” column has all unique values starting from one, they can be track ID which are unique for every row.

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.

Solution:

Handling inconsistent data types in Power Query is essential for ensuring data accuracy, error-free operations and efficient performance throughout data transformation and data analysis process.

- Approach: In “danceability” column and “energy” column replace nan value with null and fill null value with “0.5”. Change the format for remaining columns like acousticness, tempo, liveness, loundness to decimal number.
- Why? Replacing the nan value with null value help in converting both columns to change type to decimal number. Replacing null values with 0.5 which ensures filling gaps without overly inflating or underestimating the characteristics of particular track. Also changing to proper format will help in better data representation and analyzing.

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.

Solution:

1) "invalid_data":

    - In “Views” column, replace invalid_data to null, to remove error.
        - Why? Replacing invalid_data with null ensures that the column contains valid entries or blanks that can be handled without errors in calculations or transformations. This avoids errors during data processing.
    - Change data type from Text to Whole Number.
        - Why? If "Views" represents numerical data, changing its type to a whole number allows you to perform mathematical operations,

aggregations, or comparisons. It aligns the data type with its intended use.

- Trim “Comments”, “Likes” and “Views”.
    - Why? Trimming removes leading and trailing spaces, which can cause issues when performing operations like joins, filtering, or type conversion. It ensures that the data is clean and consistent.
- Replace nan to null in “Views”.
    - Why? Replacing nan (not-a-number) with null standardizes missing or invalid numerical values, making it easier to handle them during calculations or visualizations. null is the recognized placeholder for missing data in Power Query.
- Change data type of “Likes” and “Views” from text to decimal number as per existing values.
    - Why? Most values in both columns are whole numbers but some decimal also found hence decimal is the correct data type to retain all data accurately.

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:

Approach: As from earlier analysis it is found that column named “unnamed: 0” can act as unique value.

Why? Column “unnamed: 0” has numerical values starting from 1, if we remove the duplicates for this column it is clear that all values in this column are unique and in ascending order starting from 1.

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.

Solution:

1) Reorder:

   - Approach: Reordering columns in a dataset is about organizing the information in a way that improves readability, usability, and logical flow. The best approach depends on the purpose of the dataset and the relationships between the columns. Like rearranging column “unnamed:0” to front, and arranging columns like key, valence, liveness, loudness, speechiness, tempo, dancebility etc together.

   - Why? Arrangement of these columns together helps in understanding of music better because most of these explain characteristics of musics.

2) Rename:

   - Approach: Rename column “unnamed: 0” as track_unique_id, also rename the custom column which is created by giving condition , if “official_video” = null then “Spotify” else “both”, to ‘exclusive_to_platform’.

   - Why? “unnamed: 0” has unique value for each row. For custom column, it is created because some tracks are exclusive to Spotify and are not available to YouTube while other tracks are available on both.