## **Song Recommendation System Presentation**

### **Introduction**

* **Context**: In the age of digital music, users face the challenge of discovering new songs among vast libraries. Recommendation systems address this by suggesting songs a user might enjoy, enhancing user experience and engagement.
* **Objective**: This project aims to develop a song recommendation system using machine learning techniques.

### **Methodology**

1. **Data Acquisition**:
   * The dataset, "Top Spotify Songs 2023," was obtained from Kaggle.
   * The dataset contains a variety of song attributes, including track name, artist, and audio features.
2. **Data Preprocessing**:
   * Missing values in the key column were imputed with the mode (most frequent value).
   * Missing values in the in\_shazam\_charts column were filled with 0.
   * The streams column was converted to a numeric data type; rows with invalid values were removed.
3. **Feature Selection**:
   * The following numerical features were selected for the recommendation model: danceability\_%, valence\_%, and energy\_%. These features represent song characteristics relevant to user preference.
4. **Data Scaling**:
   * StandardScaler was applied to scale the selected features. This ensures that all features contribute equally to the distance calculations in the k-nearest neighbors algorithm.
5. **Model Training**:
   * A k-Nearest Neighbors (k-NN) model was trained on the scaled features.
   * k-NN is a simple yet effective algorithm for recommendation systems. For a given song, it finds the k most similar songs in the dataset based on their features.
6. **Recommendation Generation**:
   * For a given song, the trained k-NN model identifies its k nearest neighbors.
   * The track names of these neighbors are then presented as song recommendations.

### **Algorithm**

1. **Data Loading and Preprocessing**
   * Load the dataset containing song information.
   * Handle missing values in the relevant columns.
   * Convert the streams column to a numeric format, removing any non-numeric entries.
   * Select the features danceability\_%, valence\_%, and energy\_%.
   * Scale the selected features using StandardScaler.
2. **K-Nearest Neighbors (k-NN) Model**
   * Initialize a k-NN model.
   * Fit the model to the scaled feature data.
3. **Recommendation Function**
   * Define a function that takes a song name as input.
   * Find the index of the input song in the dataset.
   * Use the k-NN model to query the k-nearest neighbors to the input song.
   * Return the names of the top k nearest neighbor songs as recommendations.

### **Results**

* The system successfully generates song recommendations based on the similarity of danceability\_%, valence\_%, and energy\_% features.
* For example, when the song "LALA" is input, the system recommends songs like "Party", "Unholy," and " বৃষ্টি নেমে যায়".
* The recommendations are tailored to the characteristics of the input song, providing users with potentially relevant and enjoyable music suggestions.

### **References**

* Dataset: [Top Spotify Songs 2023](https://www.kaggle.com/datasets/nelgiriyewithana/top-spotify-songs-2023)
* KaggleHub: [KaggleHub Documentation](https://github.com/kaggle/kagglehub)
* Pandas: [Pandas Documentation](https://pandas.pydata.org/docs/)
* Scikit-learn: [Scikit-learn Documentation](https://scikit-learn.org/stable/documentation.html)

### **Additional Notes**

* The performance of the recommendation system can be further improved by:
  + Incorporating more audio features.
  + Tuning the value of k in the k-NN model.
  + Exploring other recommendation algorithms.
* The encoding='latin1' was used to handle special characters in the dataset.