## **Project Overview**

The **Clustering Music Genres** project aims to group music tracks based on the similarities in their audio features using machine learning. Streaming platforms like Spotify rely heavily on identifying patterns in users’ listening preferences to provide relevant music recommendations. By clustering songs with similar characteristics, we can understand how different music genres overlap, enhancing recommendation systems. For this task, we used a dataset containing the audio properties of 2,000 popular Spotify tracks, including metrics such as beats per minute (BPM), energy, danceability, and loudness.

## **Method**

The primary goal was to group music tracks by their audio similarities using the **K-means clustering algorithm**, a widely used unsupervised machine learning technique. The steps followed in the analysis were:

1. **Exploratory Data Analysis (EDA)**: We inspected the dataset to understand the features and identify correlations between attributes such as **energy, valence, acousticness, and popularity**.
2. **Feature Selection**: We selected specific features that capture the essence of music tracks:
   * **Beats Per Minute (BPM)**
   * **Loudness (dB)**
   * **Liveness**
   * **Valence**
   * **Acousticness**
   * **Speechiness**
3. **Clustering with K-means**:
   * We applied **K-means clustering** to group songs into **10 clusters** based on these features.
   * Each song was assigned to a cluster representing a segment of music with similar audio properties.
4. **Visualization**: We used **3D scatter plots** to visualize clusters based on **BPM, energy, and danceability**, allowing us to observe trends and overlaps between segments.

## **Data Processing**

1. **Loading and Cleaning the Data**:
   * The dataset was loaded into a pandas DataFrame, and unnecessary columns (like the Index column) were dropped.
2. **Handling Missing Data**:
   * As all columns were complete, no further imputation was needed.
3. **Feature Scaling**:
   * We applied **Min-Max scaling** to normalize the numerical features, ensuring that all variables contribute equally to the clustering process.
4. **Clustering**:
   * We used the **fit\_predict()** method from the KMeans class to perform clustering and added the resulting cluster labels to the dataset.
   * The clusters were labeled as **Cluster 1 to Cluster 10** to represent different music segments.

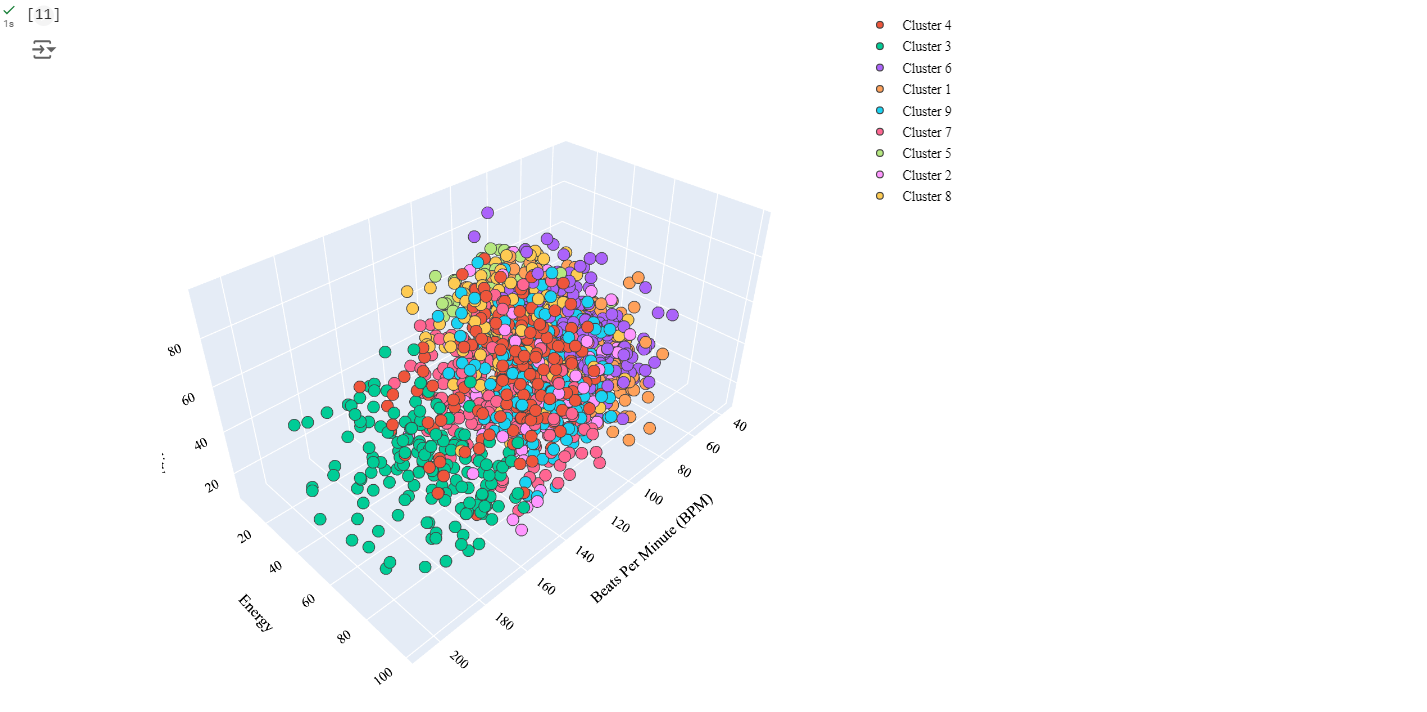
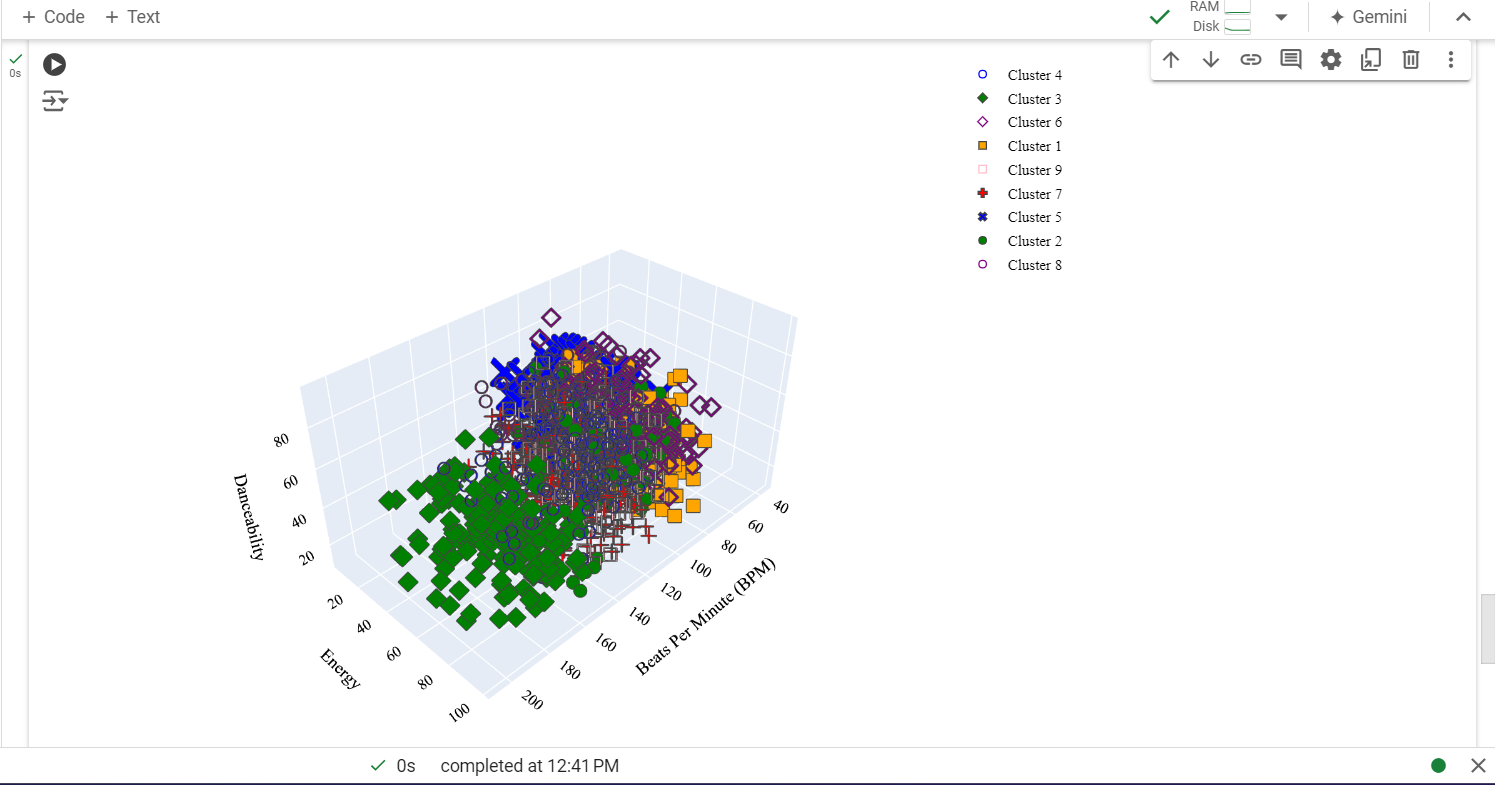
## **Results**

The K-means clustering algorithm grouped the music tracks into 10 distinct clusters. Key observations include:

1. **Energetic vs. Acoustic Tracks**:
   * Songs with high **energy** and low **acousticness** tended to form distinct clusters.
2. **Valence and Danceability**:
   * Tracks with high **danceability** and positive **valence** were more likely to appear in clusters representing upbeat genres.
3. **Popularity Insights**:
   * Popular songs were spread across different clusters, indicating that music's popularity is not determined by a single feature but a combination of several factors.

### **Visualization**

The 3D scatter plot displayed how **BPM, energy, and danceability** relate across clusters. Clusters with similar BPM ranges but differing energy levels suggest that tempo alone does not determine energy, and multiple features contribute to genre overlaps.



## **Conclusion**

This project demonstrated how **K-means clustering** can group music tracks based on their audio properties, providing insights into how genres overlap and differ. This clustering analysis has practical applications in developing more personalized music recommendation systems by identifying songs with similar characteristics.

References:

Clustering Music Genres with Machine Learning <https://thecleverprogrammer.com/2022/04/05/clustering-music-genres-with-machine-learning/>