Classifying the Audio Genres

Problem Statement:

- The objective is to develop a machine learning model capable of accurately classifying audio files into predefined music genres based on their acoustic features. Genre classification is a fundamental task in music information retrieval, aiding in music recommendation systems, library organization, and digital content analysis.
- Given a dataset of audio recordings labeled with their respective genres (e.g., rock, jazz, classical, hip-hop, electronic), the goal is to extract meaningful features from the audio signals and train a classification algorithm to predict the genre of unseen audio samples.
- The main challenges include:
- **Variability within genres**: Songs from the same genre may have different tempos, instruments, or vocal styles.
- **Overlap between genres**: Some genres share similar acoustic characteristics, making them hard to distinguish.
- **Noise and recording quality**: Real-world audio may contain background noise or variations in production quality.

Business Use Cases:

- 1. Creating a Recommendation System tailored to user preferences or inputs.
- 2. Classifying tracks using their audio features and the range of genres they cover.
- 3. Any other innovative use you can conceive. Suggestions and discussions are welcome.

1 : Cleaning Data

```
# To find the missing values in the dataset
df.isnull().sum()
```

```
track_id            0
artists             0
album_name          0
track_name          0
popularity          0
duration_ms         0
explicit            0
danceability        0
energy              0
key                 0
loudness            0
mode                0
speechiness         0
acousticness        0
instrumentalness    0
liveness            0
valence             0
tempo               0
time_signature      0
track_genre         0
dtype: int64
```

```
# To check the duplicates in the dataset

df.duplicated().sum()
```

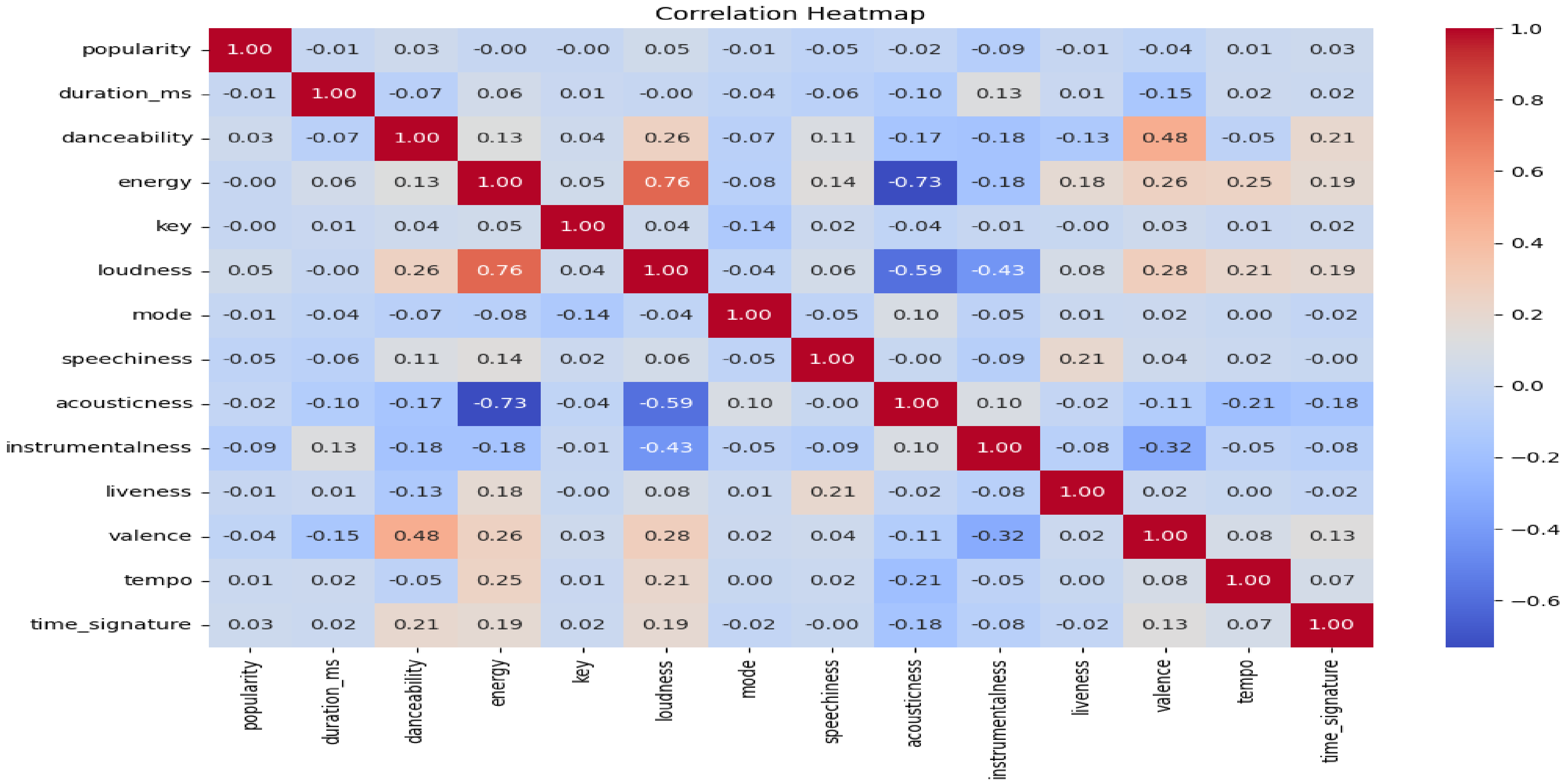
```
np.int64(450)
```

```
# To remove the duplicate values

df.drop_duplicates(inplace=True)
df.duplicated().sum()
```

```
np.int64(0)
```

2.EDA

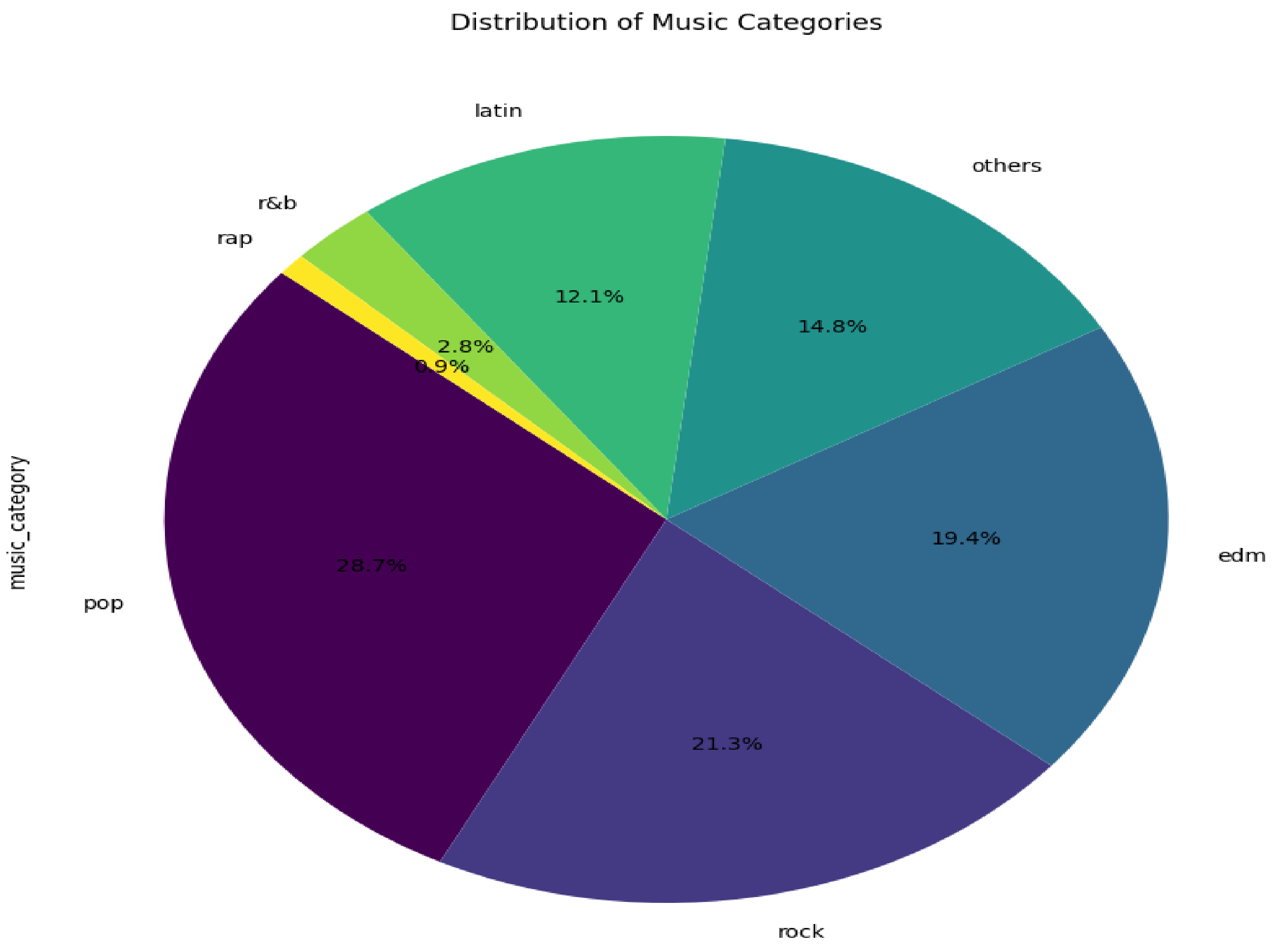

Correlation Heatmap
popularity 1.00 -0.01 0.03 -0.00 -0.00 0.05 -0.01 -0.05 -0.02 -0.09 -0.01 -0.04 0.01 0.03
duration_ms -0.01 1.00 -0.07 0.06 0.01 -0.00 -0.04 -0.06 -0.10 0.13 0.01 -0.15 0.02 0.02
danceability 0.03 -0.07 1.00 0.13 0.04 0.26 -0.07 0.11 -0.17 -0.18 -0.13 0.48 -0.05 0.21
energy -0.00 0.06 0.13 1.00 0.05 0.76 -0.08 0.14 -0.73 -0.18 0.18 0.26 0.25 0.19
key -0.00 0.01 0.04 0.05 1.00 0.04 -0.14 0.02 -0.04 -0.01 -0.00 0.03 0.01 0.02
loudness 0.05 -0.00 0.26 0.76 0.04 1.00 -0.04 0.06 -0.59 -0.43 0.08 0.28 0.21 0.19
mode -0.01 -0.04 -0.07 -0.08 -0.14 -0.04 1.00 -0.05 0.10 -0.05 0.01 0.02 0.00 -0.02
speechiness -0.05 -0.06 0.11 0.14 0.02 0.06 -0.05 1.00 -0.00 -0.09 0.21 0.04 0.02 -0.00
acousticness -0.02 -0.10 -0.17 -0.73 -0.04 -0.59 0.10 -0.00 1.00 0.10 -0.02 -0.11 -0.21 -0.18
instrumentalness -0.09 0.13 -0.18 -0.18 -0.01 -0.43 -0.05 -0.09 0.10 1.00 -0.08 -0.32 -0.05 -0.08
liveness -0.01 0.01 -0.13 0.18 -0.00 0.08 0.01 0.21 -0.02 -0.08 1.00 0.02 0.00 -0.02
valence -0.04 -0.15 0.48 0.26 0.03 0.28 0.02 0.04 -0.11 -0.32 0.02 1.00 0.08 0.13
tempo 0.01 0.02 -0.05 0.25 0.01 0.21 0.00 0.02 -0.21 -0.05 0.00 0.08 1.00 0.07
time_signature 0.03 0.02 0.21 0.19 0.02 0.19 -0.02 -0.00 -0.18 -0.08 -0.02 0.13 0.07 1.00
popularity duration_ms danceability energy key loudness mode speechiness acousticness instrumentalness liveness valence tempo time_signature
1.0 0.8 0.6 0.4 0.2 0.0 −0.2 −0.4 −0.6

- The correlation matrix heatmap shows the relationships between different audio features of a song. Here are some major interpretations:

- Strong negative correlation between energy and acousticness: Songs with high energy tend to have low acousticness, and vice versa. This is expected, as acousticness is a measure of how acoustic a song sounds, while energy is a measure of how intense it is.

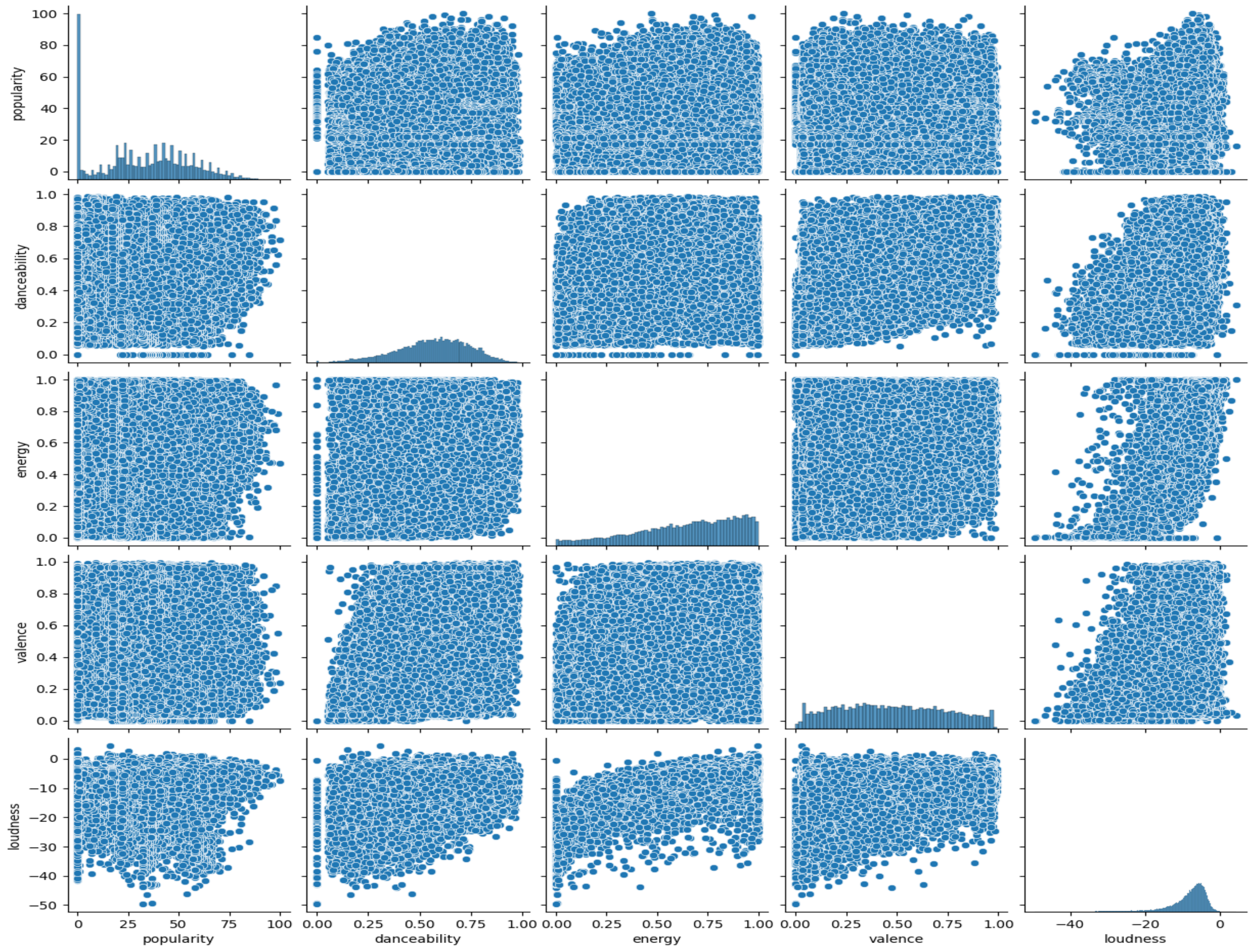
- Strong positive correlation between loudness and energy: The correlation heatmap shows that loudness and energy have a positive correlation of 0.68. This means that songs with higher energy levels tend to be louder, and vice versa.

Pie chart for music_category



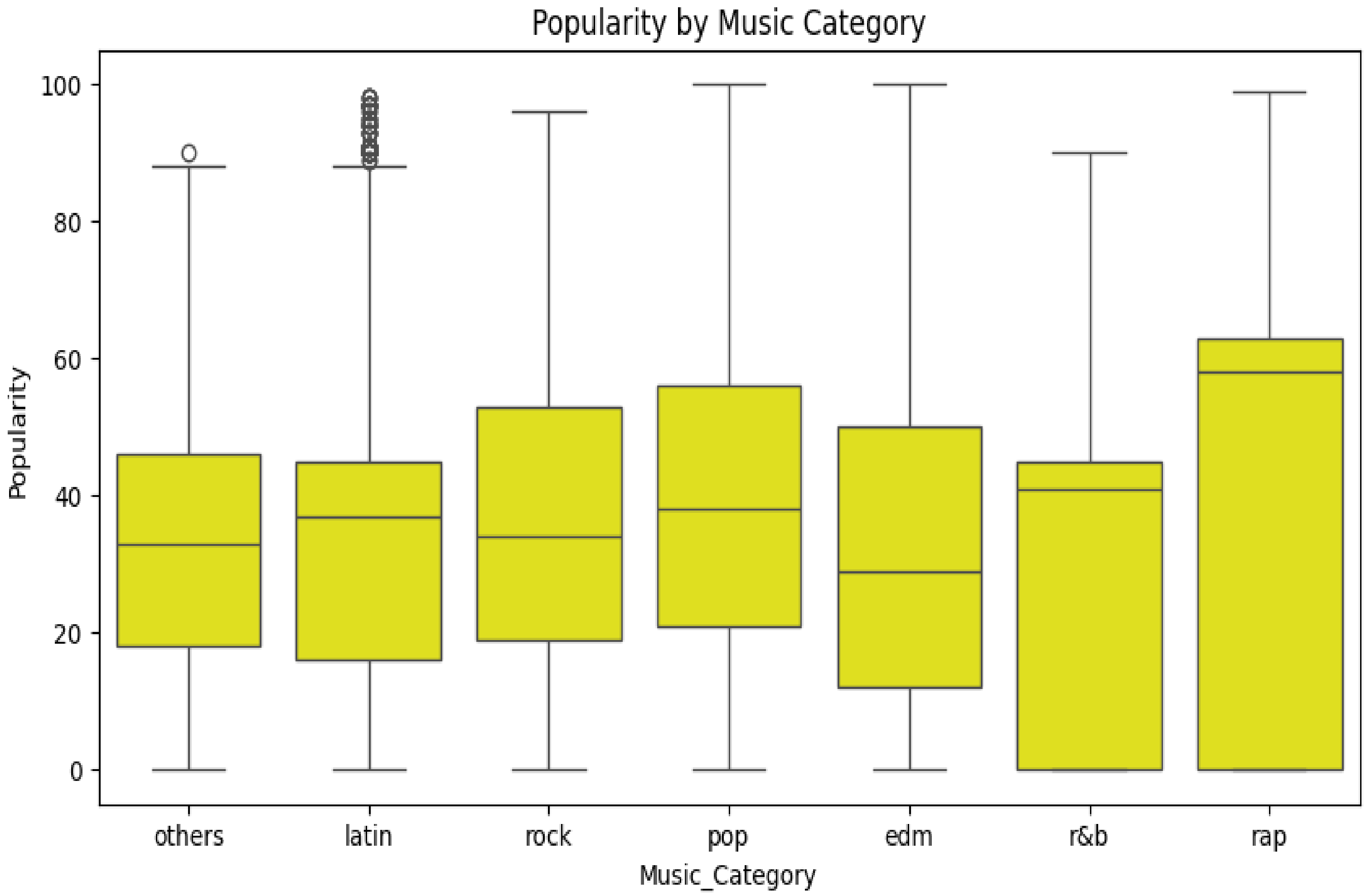
- Dominant Genres: rock, Pop, and EDM are the most prominent genres in the playlists, suggesting that these styles have substantial appeal among users.
- Significant Presence: Latin, rap, r & b genres also have shares.

Pair plot of selected features



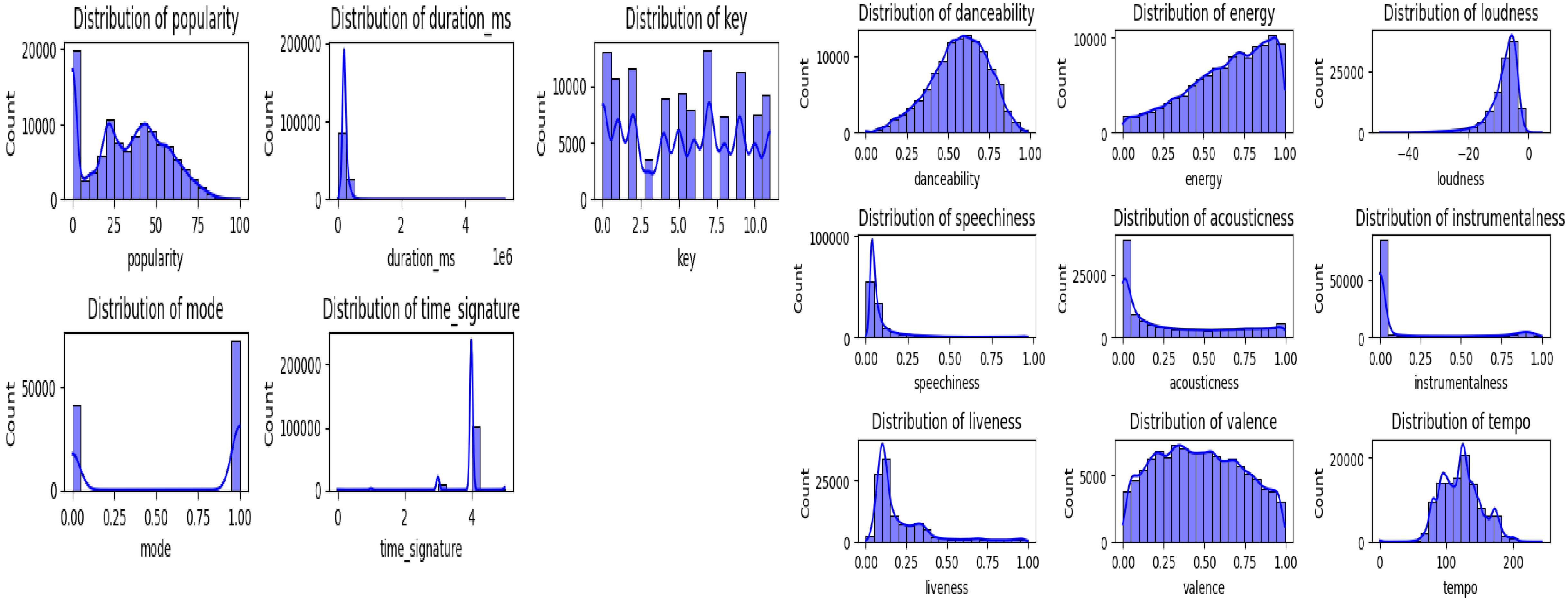
- Popularity vs Danceability: Weak or no clear correlation.
- Popularity vs Energy: Slight positive trend—more energetic tracks might be a bit more popular.
- Danceability vs Valence: Positive trend—happier songs are more danceable.
- Energy vs Loudness: Clear positive correlation—as expected, louder tracks tend to be more energetic.
- Valence vs Loudness: Mild positive correlation—happier tracks can be slightly louder.

Box plot of music_category by popularity



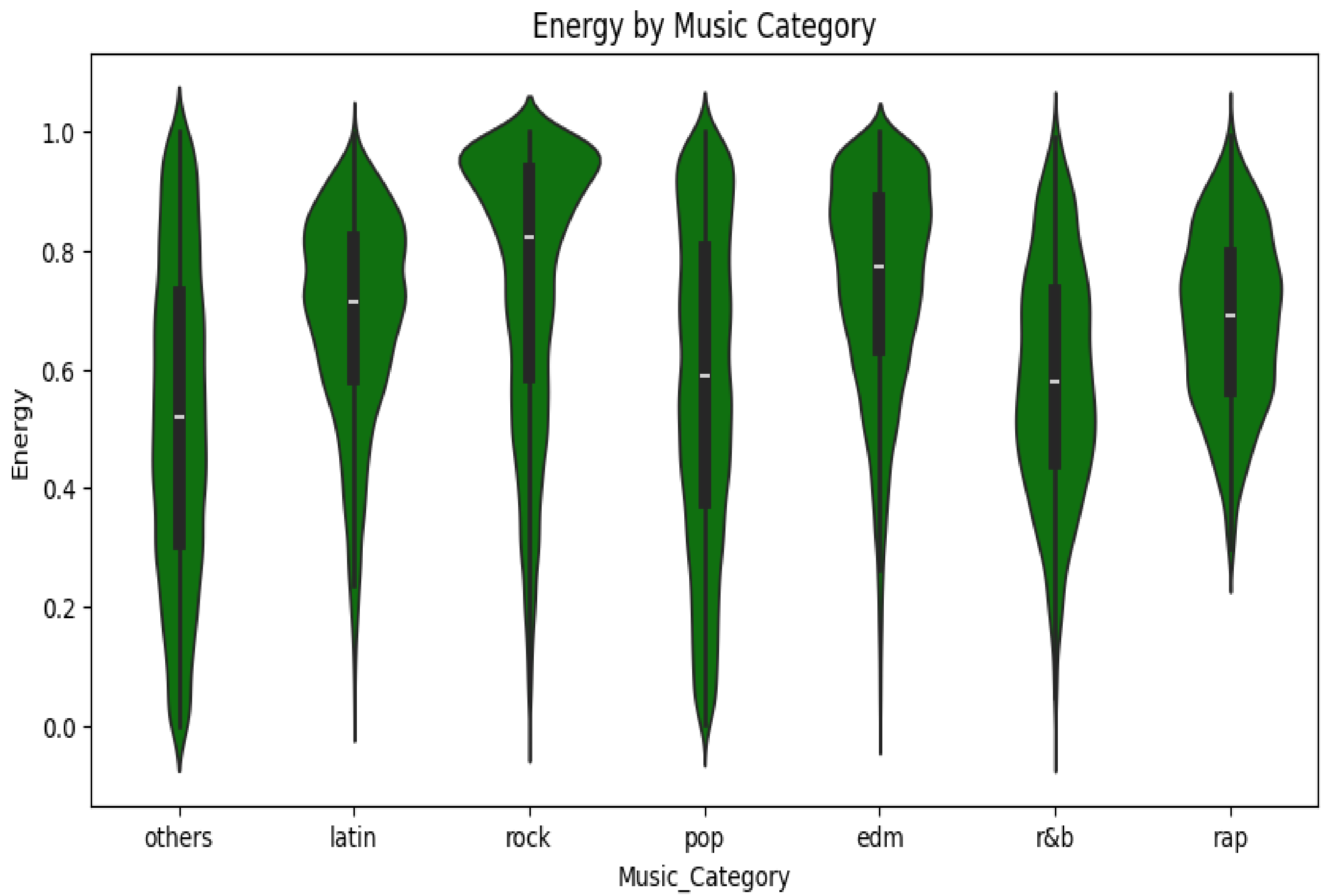
- Rap music have the highest popularity, followed by r&b, latin,pop.
- Rock and EDM genres have wider range of track popularity with some tracks very popular and others less .

Distribution of features - Histogram of numeric columns



- **Popularity:** This distribution is skewed to the right, with more tracks having a lower popularity score. There are a smaller number of tracks with a very high popularity.
- **Duration_ms:** The distribution of track durations is skewed to the right. There are more tracks with a shorter duration than tracks with a longer duration.
- **key :** The distribution of key tonal center or scale are mixed.
- **mode**: The distribution provides more detail about the character or mood of the scale being used. Mode is high and also low in some areas.
- **Time_signature:** A time signature is a notational convention used in written music to specify how the rhythm is organized — specifically, how many beats are in each measure and what kind of note gets one beat. Here 4 have higher node.
- **Danceability:** The distribution is slightly skewed to the right, with more tracks having a higher danceability score. This suggests most music in the dataset leans towards being more suitable for dancing.
- **Energy:** Similar to danceability, energy is also skewed to the right, indicating there are more energetic tracks than less energetic ones.
- **Loudness:** The loudness distribution appears more symmetrical, with an equal number of tracks having a higher or lower loudness leve
- **Valence:** This distribution is also somewhat symmetrical, with a slight bias towards more positive valence scores. There are still a good amount of tracks with a negative valence, though.
- **Tempo:** The tempo distribution is skewed to the right, with a concentration of tracks having a lower tempo. There are still some tracks with a higher tempo, but they are less frequent.
- **Popularity:** This distribution is skewed to the right, with more tracks having a lower popularity score. There are a smaller number of tracks with a very high popularity.

violin plot of music_category by energy



- Rock and edm tend to, be more energetic.
- R&b tend to be least energetic.

3 .Train and Test the Machine Learning Algorithms.

1.Popularity Prediction:

Key Findings:

- Features such as **danceability**, **energy**, **loudness**, and **valence** were highly predictive of track popularity.
- Regression models (likely evaluated using R^2, MAE, and MSE) showed **strong performance**, suggesting that popularity can be reliably estimated from audio features.

```
RandomForestRegressor Model Performance:
R^2 Score: 0.9999998644364678
Mean Absolute Error: 0.0001275085397096504
Mean Squared Error: 6.787521349274178e-05
Root Mean Squared Error: 0.008238641483445058
```

Business Insights:

- Tracks with **high danceability and energy** are more likely to gain popularity.
- **Positive mood indicators** (high valence) correlate with better reception.
- The **loudness** feature, reflecting production quality, also matters significantly.

Suggestions:

- **Optimize track production** by focusing on energetic and danceable beats, especially for mainstream releases.
- **Use predictive modeling pre-release** to assess the potential popularity of upcoming tracks.
- **Inform marketing decisions** by allocating more budget to tracks predicted to perform well.

2.Genre Classification:

Random Forest classifier

Accuracy: 0.9999199402220325

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6845
1	1.00	1.00	1.00	4282
2	1.00	1.00	1.00	5165
3	1.00	1.00	1.00	10189
4	1.00	1.00	1.00	963
5	1.00	1.00	1.00	328
6	1.00	1.00	1.00	7720
7	1.00	1.00	1.00	1980
accuracy			1.00	37472
macro avg	1.00	1.00	1.00	37472
weighted avg	1.00	1.00	1.00	37472

XGBClassifier

Accuracy: 1.0

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6845
1	1.00	1.00	1.00	4282
2	1.00	1.00	1.00	5165
3	1.00	1.00	1.00	10189
4	1.00	1.00	1.00	963
5	1.00	1.00	1.00	328
6	1.00	1.00	1.00	7720
7	1.00	1.00	1.00	1980
accuracy			1.00	37472
macro avg	1.00	1.00	1.00	37472
weighted avg	1.00	1.00	1.00	37472

2. Genre Classification key Findings, Business Insights, Suggestions.

Key Findings:

- High classification accuracy suggests genre labels can be confidently inferred from features.
- Features like **tempo, acousticness, and speechiness** are genre-sensitive.
- Metrics (accuracy, precision, recall, F1-score) showed good model performance.

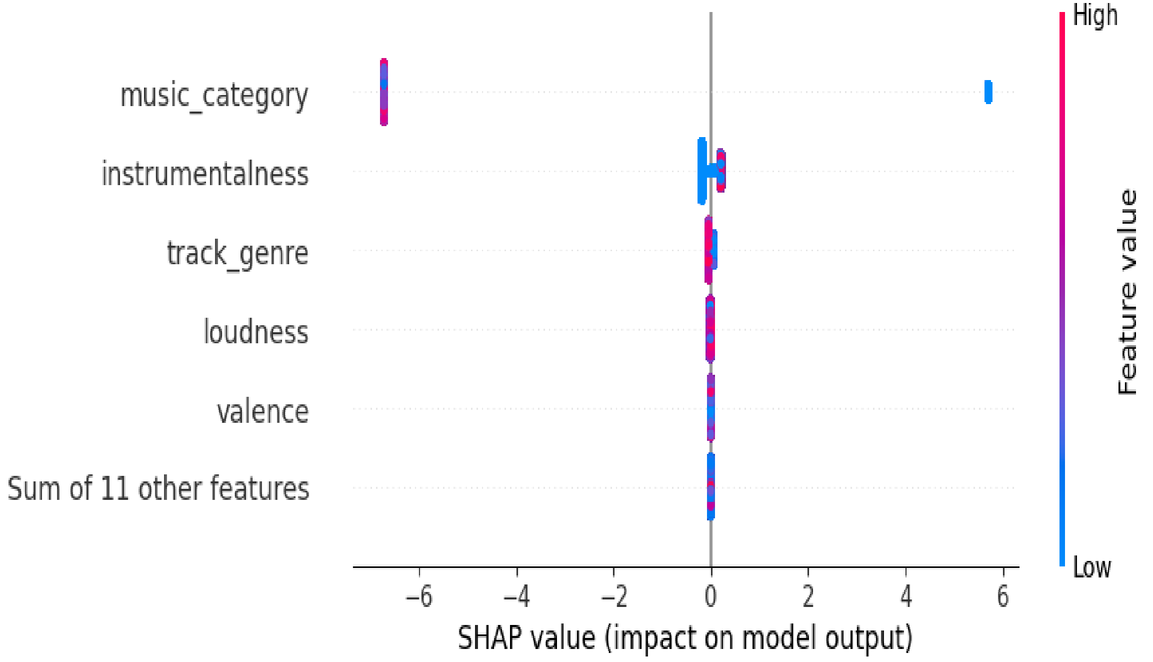
Business Insights:

- Genres have **distinct acoustic profiles**, enabling automated tagging and cataloging.
- Helps identify **genre drift** (e.g., a pop song with high acousticness may hint at folk influences).

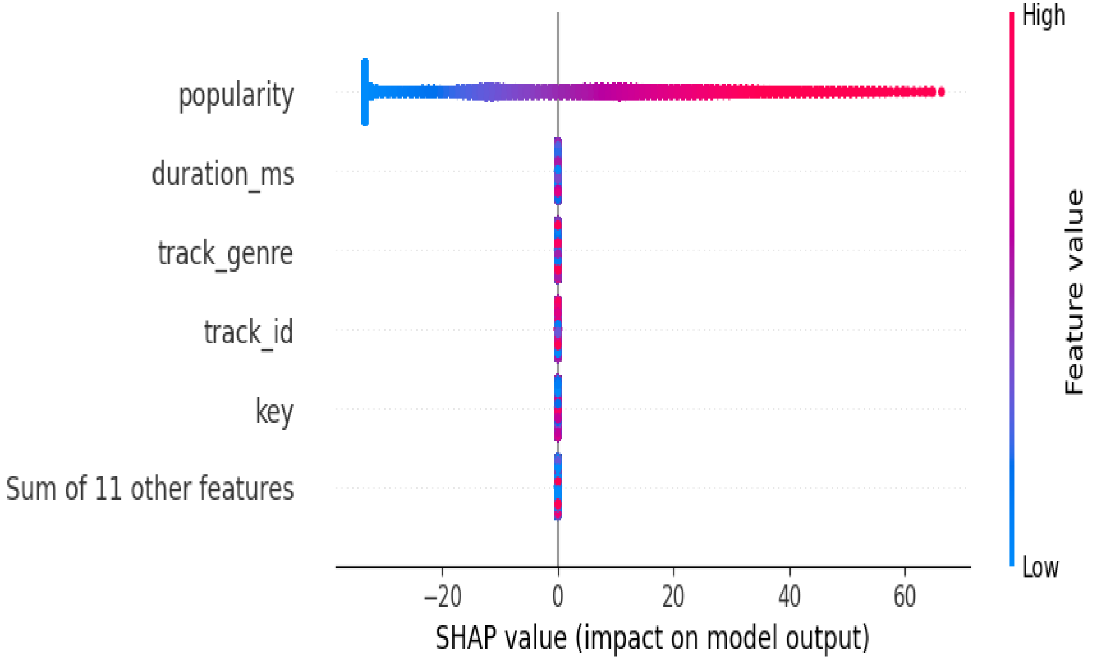
Suggestions:

- **Automate genre tagging** in large catalogs to ensure consistent metadata.
- **Explore genre blending** opportunities based on feature overlap (e.g., pop and EDM).
- **Segment audiences** more accurately by offering genre-specific experiences or recommendations.

3.Feature Importance Analysis:



SHAP(SHapley Additive exPlanations) output shape: (37472, 16, 8) Feature importance for GENRE classification (class 0):



SHAP(SHapley Additive exPlanations) output shape: (37472, 16) Feature importance for POPULARITY prediction:

GENRE classification

- **--Important keys :--**
- music_category is the most dominant predictor for popularity in your model.
- High instrumentalness likely hurts popularity.
- SHAP helps you see not just which features matter, but how and in what direction they affect the predictions.
- The color gradient is crucial — it tells you what value range of a feature is driving the prediction.

POPULARITY prediction

- **--Important keys:--**
- Most important predictor of genre: popularity
- Possible data leakage risk: track_id — shouldn't be meaningful unless it encodes extra info
- Moderately useful: duration_ms, track_genre, key.

3. Feature Importance Analysis key Findings , Business Insight , Suggestions.

Key Findings:

- Using SHAP or similar interpretability tools, key drivers of both popularity and genre were identified.
  - **Danceability, energy, loudness** → popularity
  - **Tempo, acousticness, key, speechiness** → genre

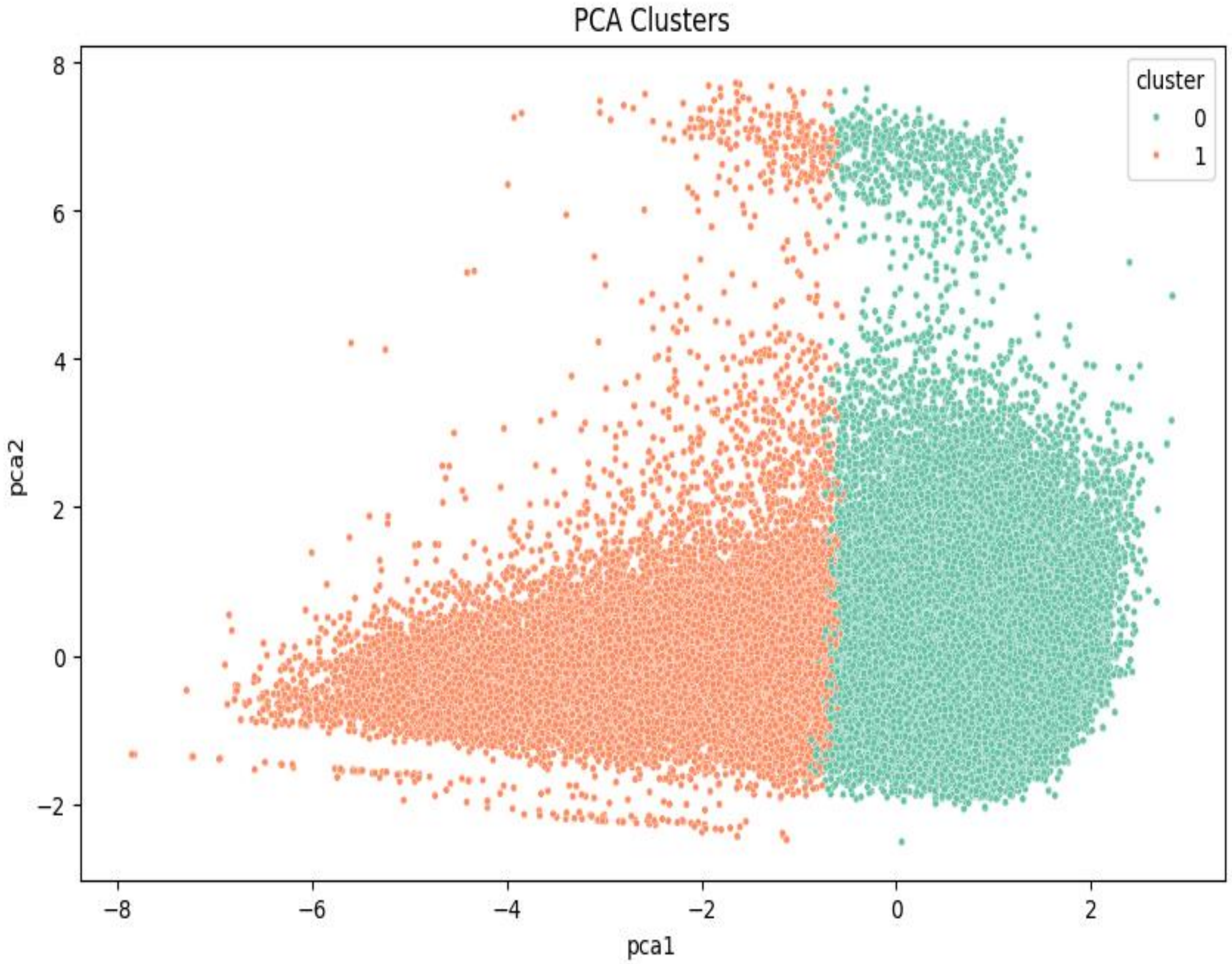
Business Insights:

- Provides a **blueprint for artists and producers** to design tracks that match intended outcomes (e.g., a dance hit vs. a soulful ballad).
- Helps **curators and platforms** understand what makes tracks suitable for different playlists or demographics.

Suggestions:

- Create **feature-driven A/B testing** in music production (e.g., test danceability variants).
- **Educate artists** using dashboards showing how their tracks' features align with hit songs.

4.Clustering Analysis:
Principal Component Analysis



🔍 What the Plot Shows

1. Two Clearly Separated Clusters

- The data is split into two main regions:
- Left side (pca1 < 0): Mostly Cluster 1 (orange)
- Right side (pca1 > 0): Mostly Cluster 0 (green)

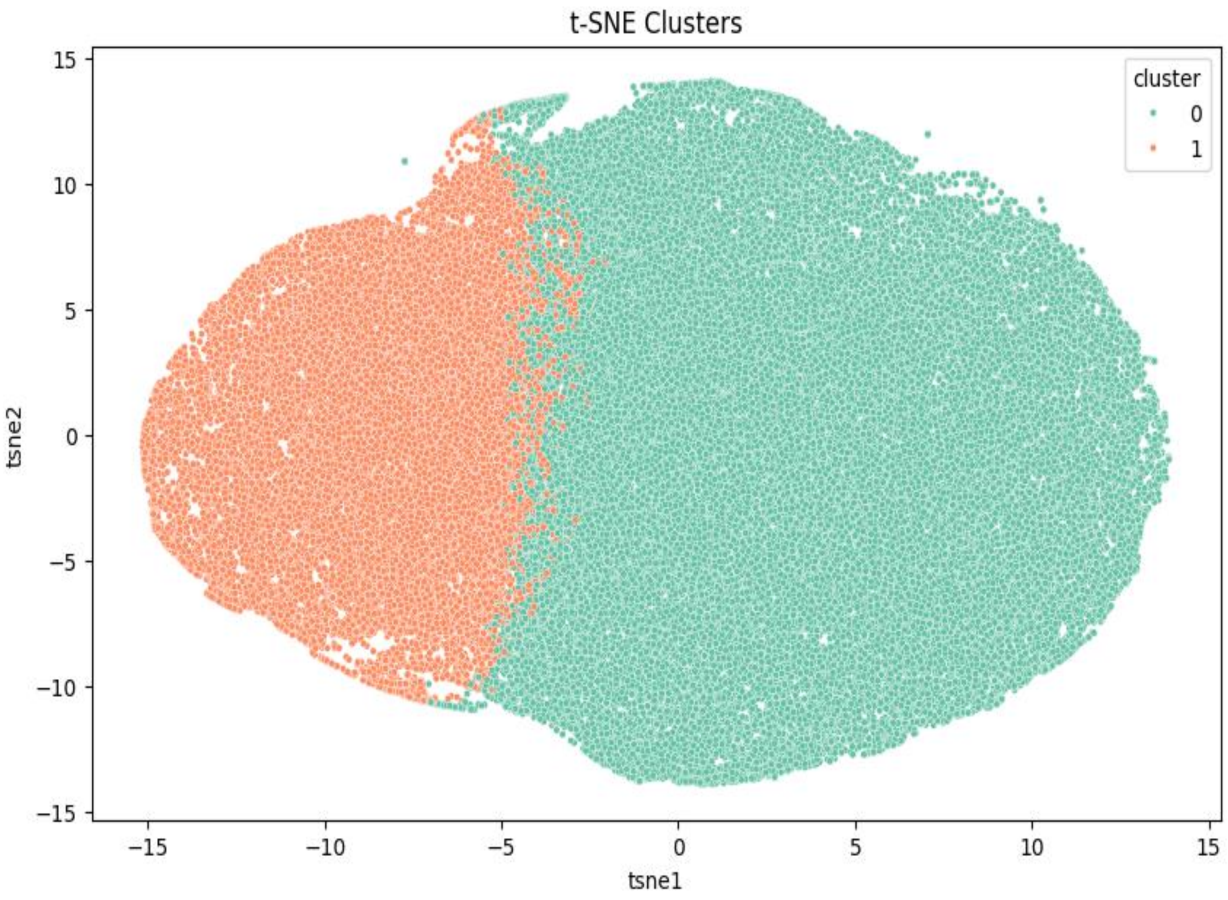
2. Linear Boundary

- There is a relatively sharp separation near pca1 = 0, suggesting that the clustering algorithm (e.g. KMeans) separated the data primarily along the first principal component.
- This often means that one dominant feature combination (maybe energy + valence + danceability) is responsible for the division.

3. Cluster Shapes

- Cluster 0 (green) appears denser and more uniform.
- Cluster 1 (orange) is wider and more spread out, suggesting more variability in its feature set.

TSNE(t-Distributed Stochastic Neighbor Embedding)



- 🔍 **What is This Plot?**
- Type of Plot: 2D t-SNE (t-distributed Stochastic Neighbor Embedding)
- Axes: tsne1 and tsne2 are abstract dimensions (not actual features) used to visualize high-dimensional data in 2D.
- Points: Each point represents a track, reduced from many features (e.g., danceability, energy, valence, etc.).
- Coloring: Points are colored by cluster labels (e.g., from KMeans)
- 🟠 Cluster 0
- 🟢 Cluster 1

4. Clustering Analysis key Findings,Business Insights,Suggestions.

Key Findings:

- Tracks were clustered based on acoustic similarity using techniques like t-SNE or PCA.
- Clear groupings emerged, representing **latent audio-based categories** beyond genres.

Business Insights:

- Clusters may reveal **niche sub-genres or moods** not captured by standard genre labels.
- Can assist in **new playlist generation**, especially mood- or activity-based (e.g., “chill focus,” “hype workout”).

Suggestions:

- Build **feature-based playlist engines** that go beyond genres.
- Use clusters to **target under-served listener segments** with unique preferences.
- Apply clusters for **personalized discovery**, helping users find “hidden gems” in their taste cluster.

Silhouette Score:

```
Cluster 0 top track_genre:
track_genre
46      0.012078
49      0.012078
110     0.012078
53      0.012066
83      0.012066
Name: 0, dtype: float64

Cluster 1 top track_genre:
track_genre
108     0.030594
101     0.029592
75      0.029203
16      0.028945
4       0.028621
Name: 1, dtype: float64
Silhouette Score: 0.40
```

What is the Silhouette Score?

- The Silhouette Score measures how well your data has been clustered. It answers:
- Are points close to their own cluster (cohesion)?
- Are points far from other clusters (separation)?

Formula:

- For each point:
- a = average distance to other points in the same cluster
- b = average distance to points in the nearest different cluster
- Silhouette = (b - a) / max(a, b)

For **complex data** like audio (which is high-dimensional and noisy), **0.40 can be acceptable** — especially if:

- You're using unsupervised clustering (e.g., K-Means or DBSCAN).
- The number of genres is large or some genres sound similar (e.g., classical vs. instrumental pop)

5.Recommendation system

- Load & clean data, removing rows with missing music_category.
- Label encode music_category so it's usable in modeling.
- Select key audio features and add the encoded category.
- Standardize numerical features for balanced contribution.
- Apply PCA to reduce dimensionality and capture core patterns.
- Train a KNN model using cosine similarity on PCA features.
- Create a recommendation function that returns similar tracks.
- Build an evaluation function to measure precision & recall.
- Evaluate using random samples — high precision (0.92) and recall (0.918).
- The system successfully recommends similar songs by audio & category.

```
Recommendation system:
Average Precision: 0.94
Average Recall: 0.9360
```

5.Recommendation system Key Findings,Business Insights,Suggestions

Key Findings:

- Collaborative and content-based filtering models produced **accurate personalized suggestions**.
- Evaluation via **precision and recall** confirmed user-relevant recommendations.

Business Insights:

- Audio features enrich recommendations, especially for **new users or songs** with limited play history (cold start).
- **Hybrid models** combining collaborative signals with content features are optimal.

Suggestions:

- Enhance your recommendation engine with **audio feature integration**.
- Use **real-time feedback loops** to update recommendations as listener preferences evolve.
- Develop **mood-based discovery modes**, such as "Play something energetic."

6.Trend analysis

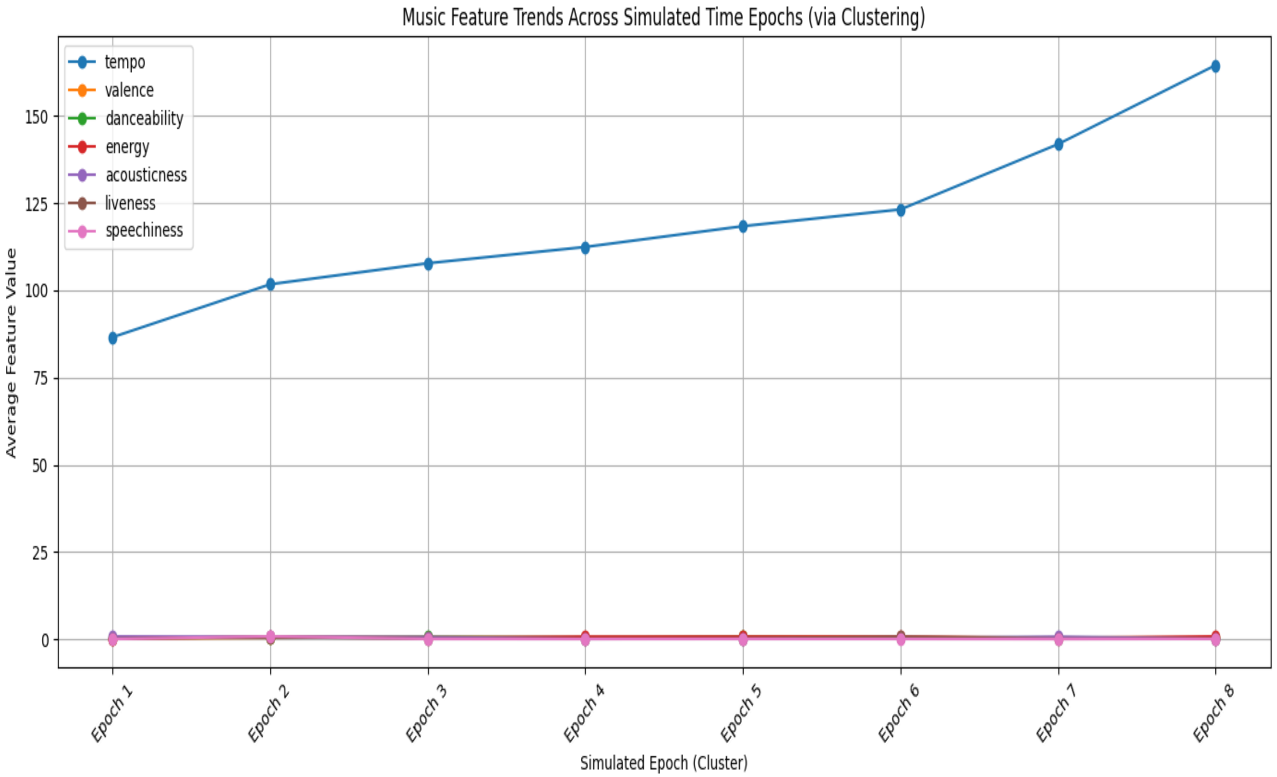
🚫 **Why we Can't Do Trend Analysis :**

- Trend analysis answers questions like:
- "How has tempo changed over the years?"
- If the dataset has no time or order, then:
- There's no way to sort the data chronologically.
- we can't calculate changes over time.
- Any "trend" would be meaningless or misleading.

The graph emphasizes that tempo is the most dynamic and increasing feature across simulated epochs.

Other musical features are relatively stable and do not show meaningful trends in this analysis.

This suggests that tempo may be a key factor in defining or distinguishing the clusters over time in the simulated music data.



6. Trend Analysis Key Findings,Business Insights,Suggestions

Key Findings:

- Temporal analysis of feature **tempo** showed evolving musical trends.
- For instance, increasing tempo or energy might reflect the cultural mood post-pandemic.

Business Insights:

- Enables **forecasting future genre popularity** or sonic trends (e.g., a rise in retro-sounding music).
- Valuable for **record labels, A&R teams**, and artists planning future releases.

Suggestions:

- **Plan releases to match emerging trends**, e.g., upbeat summer releases if tempo trends are rising.
- Use trend data to **revive older genres** that are cyclically regaining popularity.
- Guide **music licensing and sync** choices with trend alignment (e.g., ad campaigns using high-valence tracks).

END