

Music Genre Classification

(Phân loại thể loại âm nhạc)

Thành viên nhóm:

Lê Văn Thông - 3123410362

Phan Thanh Thịnh - 3123410360

Võ Hoàng Thông - 3123410363

Giáo viên hướng dẫn: TS.Đỗ Như Tài

Giới thiệu:

Music Genre Classification là quá trình sử dụng các kỹ thuật xử lý âm thanh và học máy để tự động xác định thể loại của một bản nhạc dựa trên các đặc trưng như tiết tấu, giai điệu và phổ tần số. Công nghệ này giúp hỗ trợ gợi ý nhạc, quản lý dữ liệu và nâng cao trải nghiệm người dùng trên các nền tảng âm nhạc số.



Mục lục

1. Định nghĩa Vấn đề
2. Chuẩn bị Dữ liệu
3. Phân tích khám phá dữ liệu - EDA
4. Xử lý dữ liệu
5. Lựa chọn và huấn luyện mô hình
6. Đánh giá & Kết luận

1. Định nghĩa Vấn đề

Mục tiêu

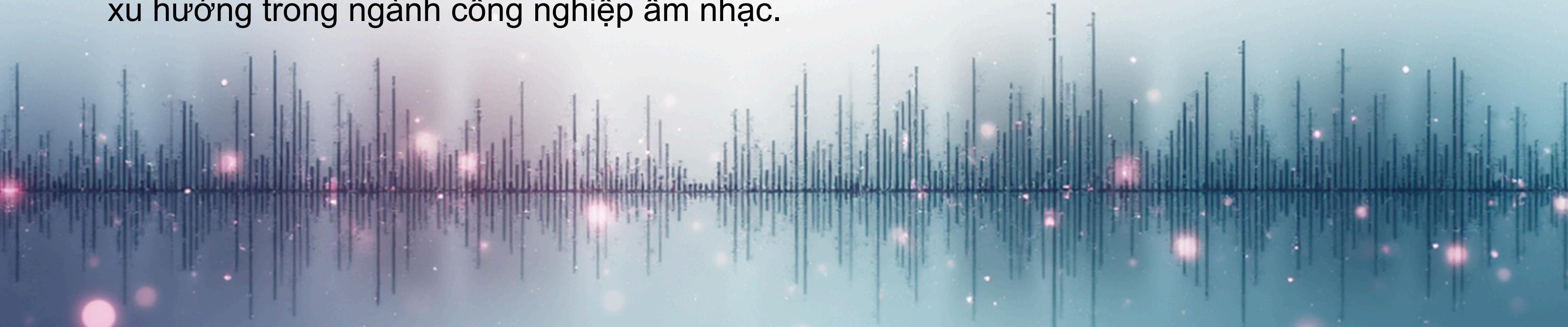
- Xây dựng một hệ thống có khả năng tự động phân loại các bản nhạc vào từng thể loại dựa trên đặc trưng âm thanh.
- Ứng dụng các phương pháp học máy/học sâu để nâng cao độ chính xác của việc nhận dạng thể loại.
- Phân tích và rút trích các đặc trưng quan trọng như spectrogram, MFCC nhằm phục vụ quá trình huấn luyện mô hình.
- Kiểm thử và đánh giá mô hình trên bộ dữ liệu chuẩn để xác định mức độ hiệu quả.
- Hỗ trợ các ứng dụng thực tế như gợi ý nhạc, tìm kiếm thông minh và tổ chức thư viện âm nhạc.



1. Định nghĩa Vấn đề

Ứng dụng

- Hệ thống gợi ý nhạc (Music Recommendation): Giúp các nền tảng như Spotify, Apple Music đề xuất bài hát phù hợp với sở thích người dùng.
- Quản lý và tổ chức thư viện âm nhạc: Tự động phân loại bài hát theo thể loại, hỗ trợ sắp xếp, tìm kiếm nhanh chóng.
- Nhận dạng nội dung (Content Identification): Hỗ trợ các ứng dụng phân tích và hiểu nội dung âm thanh như Shazam hoặc Youtube Content ID.
- Phân tích hành vi người dùng: Giúp các dịch vụ âm nhạc hiểu xu hướng thưởng thức và tối ưu hóa trải nghiệm nghe nhạc.
- Hỗ trợ sản xuất và nghiên cứu âm nhạc: Phục vụ nhu cầu phân tích, thống kê và phát hiện xu hướng trong ngành công nghiệp âm nhạc.



2. Chuẩn bị Dữ liệu

Mô tả các file trong bộ dữ liệu từ Kaggle

train.csv

→ Bộ dữ liệu huấn luyện, bao gồm tất cả các đặc trưng âm nhạc (danceability, energy, tempo, acousticness, ...) cùng với nhãn Class (thể loại nhạc).

Được dùng để huấn luyện và tối ưu mô hình phân loại thể loại âm nhạc.

test.csv

→ Bộ dữ liệu kiểm tra, chỉ chứa đặc trưng, không có cột Class.

Nhiệm vụ là dự đoán thể loại nhạc cho từng bài hát dựa trên mô hình đã học từ tập train.

data_description.txt

→ File mô tả ý nghĩa đầy đủ của từng cột trong dữ liệu như artist, energy, speechiness, tempo...

Giúp người dùng hiểu rõ từng thuộc tính âm thanh và cách chúng được sử dụng cho tác vụ phân loại.

sample_submission.csv

→ File mẫu để nộp kết quả, thể hiện định dạng dự đoán cuối cùng mà mô hình phải tạo ra (example_id và genre dự đoán).

Được dùng để tham khảo khi tạo file submission cho bộ chấm điểm.

2. Chuẩn bị Dữ liệu

Giải thích chi tiết các trường dữ liệu

1. Thông tin mô tả bài hát

- artist: Tên nghệ sĩ.
- song: Tên bài hát.
- popularity: Mức độ phổ biến của bài hát (giá trị càng cao → càng nổi tiếng).



2. Chuẩn bị Dữ liệu

Giải thích chi tiết các trường dữ liệu

2. Các đặc trưng âm nhạc (Audio Features – dạng số)

- danceability: Mức độ phù hợp để nhảy, dựa trên nhịp điệu và tempo.
- energy: Mức năng lượng hoặc cường độ của bài hát (0–1).
- key: Tông của bài hát (0 = C, 1 = C#/D \flat , ...).
- loudness: Độ lớn âm thanh trung bình (dB).
- mode: Thang âm (1 = major, 0 = minor).
- speechiness: Mức độ chứa lời nói trong bản thu.
- acousticness: Độ "acoustic" của bài hát (0–1).
- instrumentalness: Khả năng không chứa giọng hát (0–1).
- liveness: Mức độ biểu diễn live (0–1).
- valence: Mức độ “tươi vui” của giai điệu (0–1).
- tempo: Tốc độ nhạc tính bằng BPM.
- duration_ms: Thời lượng bài hát (ms).
- time_signature: Chỉ số nhịp (ví dụ 3/4, 4/4,...).

2. Chuẩn bị Dữ liệu

Giải thích chi tiết các trường dữ liệu

3. Nhãn phân loại

- Class: Thể loại nhạc (genre) của bài hát — mục tiêu cần dự đoán.

4. Cấu trúc file

- train.csv: Dữ liệu huấn luyện gồm đầy đủ các đặc trưng + nhãn Class.
- test.csv: Dữ liệu kiểm tra chỉ chứa các đặc trưng, không có nhãn.
- example_id, feature_1, feature_2,...: Cách đặt tên cột của tập test được tiêu chuẩn hóa như format cuộc thi (ví dụ Kaggle).



3. Phân tích khám phá dữ liệu - EDA

Tổng quan dữ liệu

Bộ dữ liệu gồm các bài hát với nhiều đặc trưng âm thanh (audio features) theo chuẩn Spotify API. Các đặc trưng đều là dạng số (numeric), phù hợp cho mô hình hóa bằng các thuật toán học máy.

Một số nhóm dữ liệu chính:

- Thông tin mô tả: artist, song
 - Đặc trưng âm nhạc: danceability, energy, loudness, tempo, valence...
 - Thời lượng & nhịp: duration_ms, time_signature
 - Nhãn phân loại: Class (genre)
- Đây là bài toán classification đa lớp (multi-class), không phải regression.



3. Phân tích khám phá dữ liệu - EDA

Kiểm tra dữ liệu thiếu (Missing Values)

Cột	Số giá trị thiếu
Id	0
Artist Name	0
Track Name	0
Popularity	333
danceability	0
energy	0
key	1609
loudness	0
mode	0
speechiness	0
acousticness	0
instrumentalness	3541
liveness	0
valence	0
tempo	0
duration_in min/ms	0
time_signature	0
Class	0

3. Phân tích khám phá dữ liệu - EDA

Kiểm tra dữ liệu thiếu (Missing Values)

Các cột có Missing Values:

1. **instrumentalness** — 3541 missing

→ Đây là cột có lượng thiếu nhiều nhất.

→ Có thể cân nhắc impute bằng median hoặc bỏ cột nếu độ quan trọng thấp.

2. **key** — 1609 missing

→ Key là dạng categorical numeric (0–11).

→ Có thể fill bằng mode (giá trị xuất hiện nhiều nhất).

3. **popularity** — 333 missing

→ Mức độ thiếu không nhiều; có thể fill bằng mean/median.

Các cột còn lại: KHÔNG CÓ giá trị thiếu.

3. Phân tích khám phá dữ liệu - EDA

Phân bố của các đặc trưng quan trọng

Energy

- Dao động trong khoảng 0.4 – 0.9 đối với đa số bài hát.
- Các thể loại như EDM, Rock có energy cao; Classical và Acoustic có energy thấp.

Danceability

- Nằm nhiều ở mức 0.5 – 0.8.
- Pop, EDM, Hip-hop thường có danceability cao.

Loudness

- Thường nằm trong khoảng -30 dB → 0 dB
- Các thể loại nhẹ như Acoustic hoặc Classical có loudness thấp (âm lượng nhỏ hơn).

Acousticness

- Thường phân bố lệch, nhiều bài có acousticness thấp (nhạc điện tử).
- Một số thể loại như Folk, Indie Acoustic có acousticness cao.

Speechiness

- Rap có speechiness cao.
- Nhạc pop/EDM có speechiness thấp.

3. Phân tích khám phá dữ liệu - EDA

Quan hệ giữa các đặc trưng (Correlation)

Một số mối tương quan đáng chú ý:

Energy Loudness → tương quan dương mạnh (nhạc mạnh thường to).

Acousticness Energy → càng acoustic thì energy càng thấp.

Valence Danceability → bài hát vui (valence cao) thường dễ nhảy.

Tempo ít tương quan với các biến khác (rất phân tán).

→ Đây là thông tin hữu ích để chọn feature khi xây dựng mô hình.

3. Phân tích khám phá dữ liệu - EDA

Phân tích theo từng thể loại (Class)

Đặc trưng giúp phân biệt tốt nhất các thể loại:

speechiness (phân biệt Rap, Spoken Word)

acousticness (nhạc Acoustic/Indie Folk)

energy + loudness (Rock/EDM vs Classical/Jazz)

danceability (Pop/EDM vs classical)

valence (Pop thường vui hơn Hip-hop hoặc Blues)

Nhận xét quan trọng:

Một số thể loại có ranh giới rất rõ ràng (Rap, Acoustic).

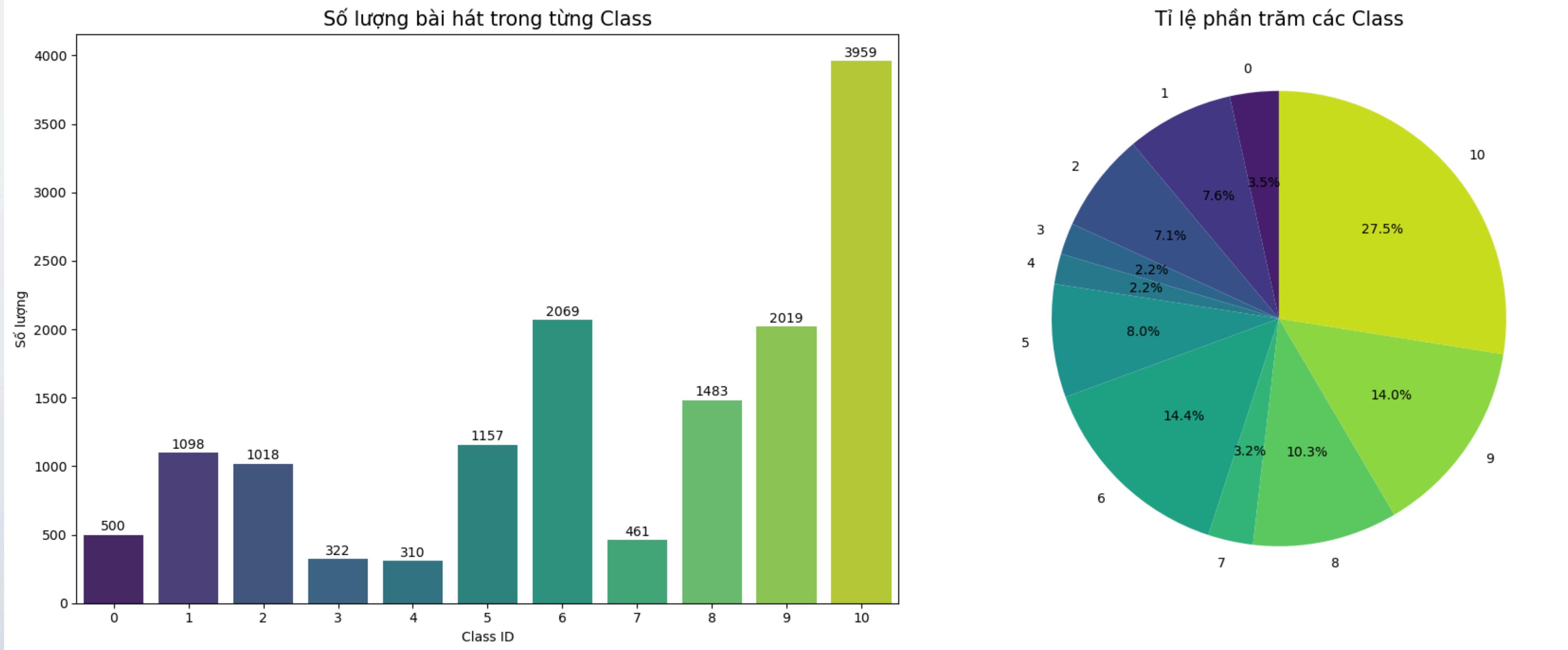
Một số thể loại dễ nhầm lẫn (Pop vs Dance, Rock vs Alternative).

Điều này ảnh hưởng đến độ chính xác của mô hình và gợi ý nên dùng mô hình mạnh như

Random Forest, XGBoost hoặc Neural Network.

3. Phân tích khám phá dữ liệu - EDA

Phân tích theo từng thể loại (Class)

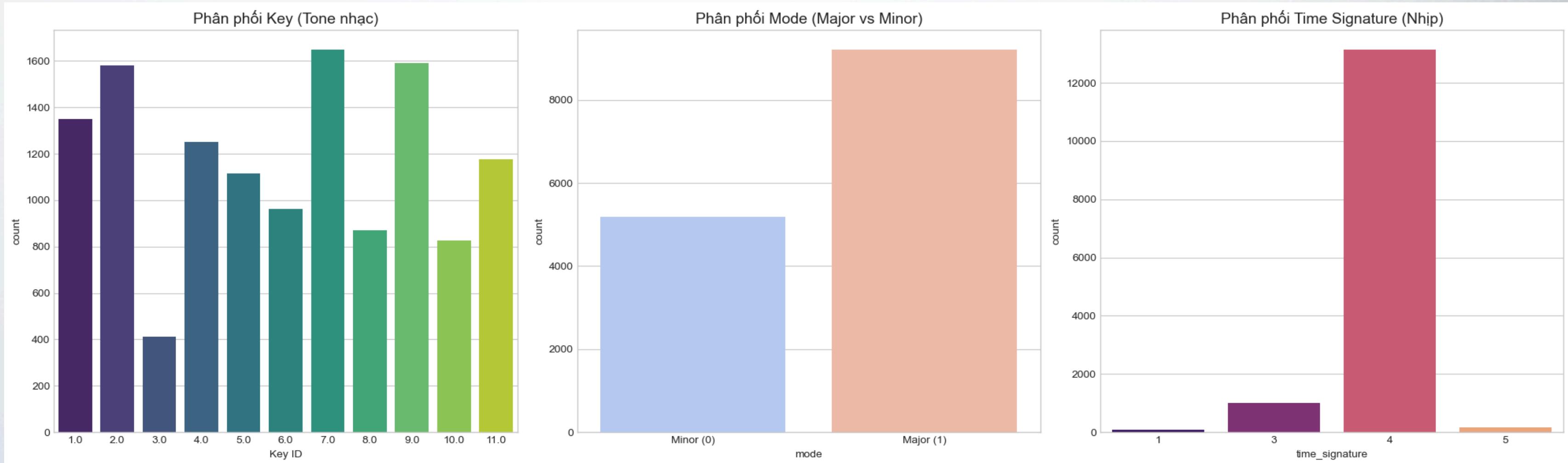


Nhận xét:

Dữ liệu bị mất cân bằng Class 10 chiếm tới 27.5% (3959 mẫu), áp đảo hoàn toàn so với các class khác. Các Class 3, 4, 7 có số lượng cực thấp (chỉ khoảng 2.2% - 3.2%, tương đương hơn 300 mẫu)

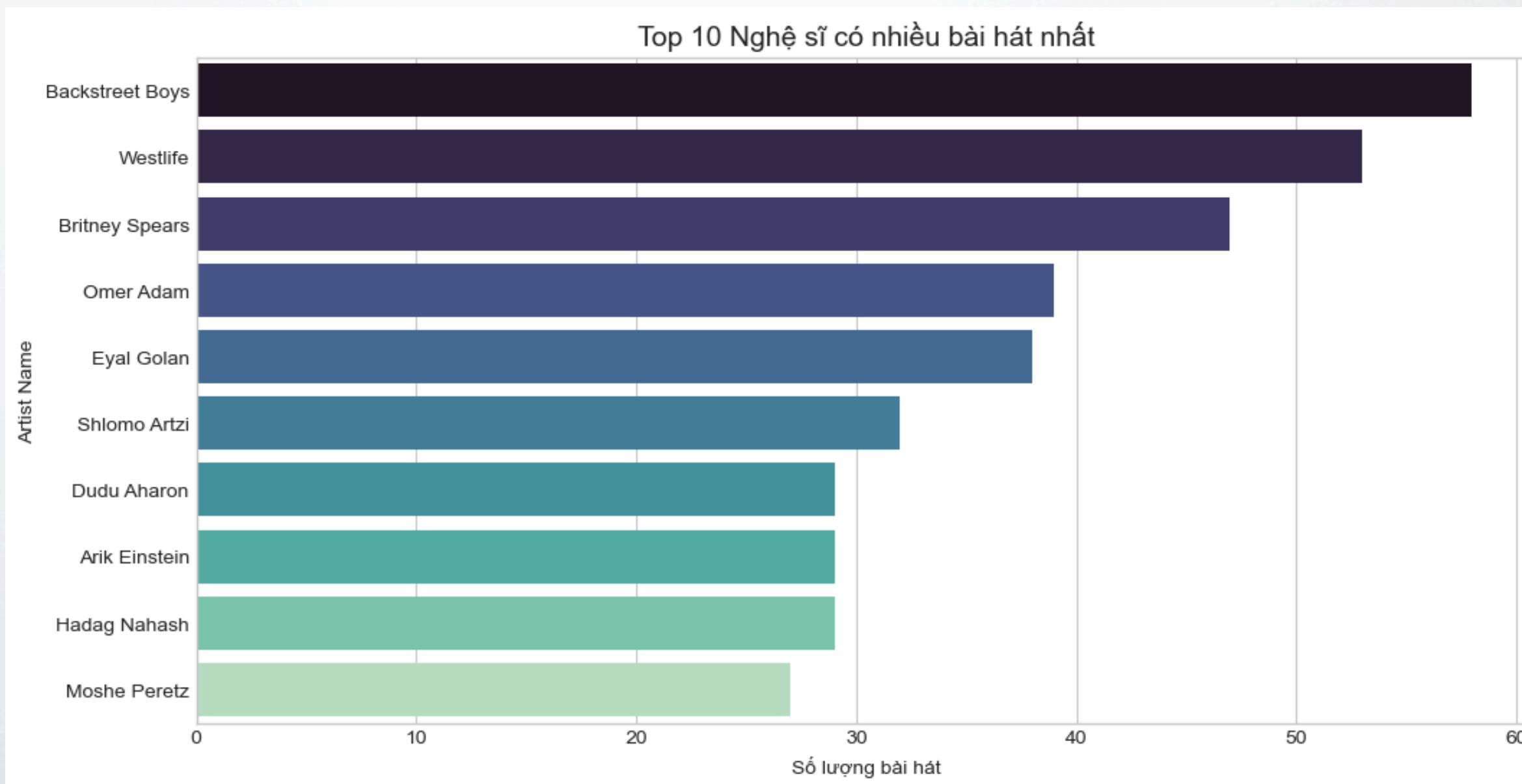
3. Phân tích khám phá dữ liệu - EDA

EDA cho các biến phân loại (Categorical)



3. Phân tích khám phá dữ liệu - EDA

EDA cho các biến phân loại (Categorical)

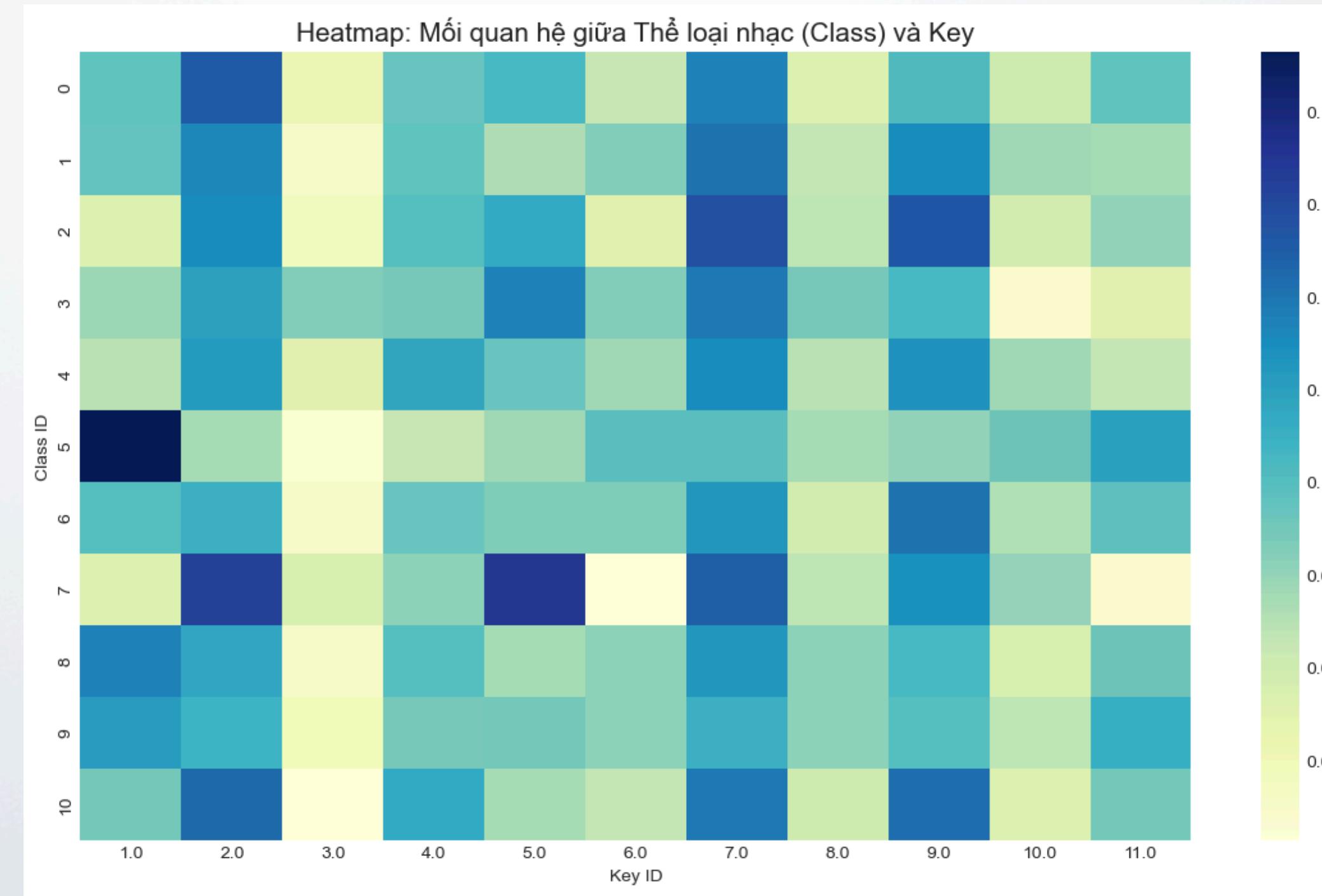


Tổng số nghệ sĩ (Unique Artists): 7913

Tổng số bài hát (Unique Tracks): 12455

3. Phân tích khám phá dữ liệu - EDA

EDA cho các biến phân loại (Categorical)

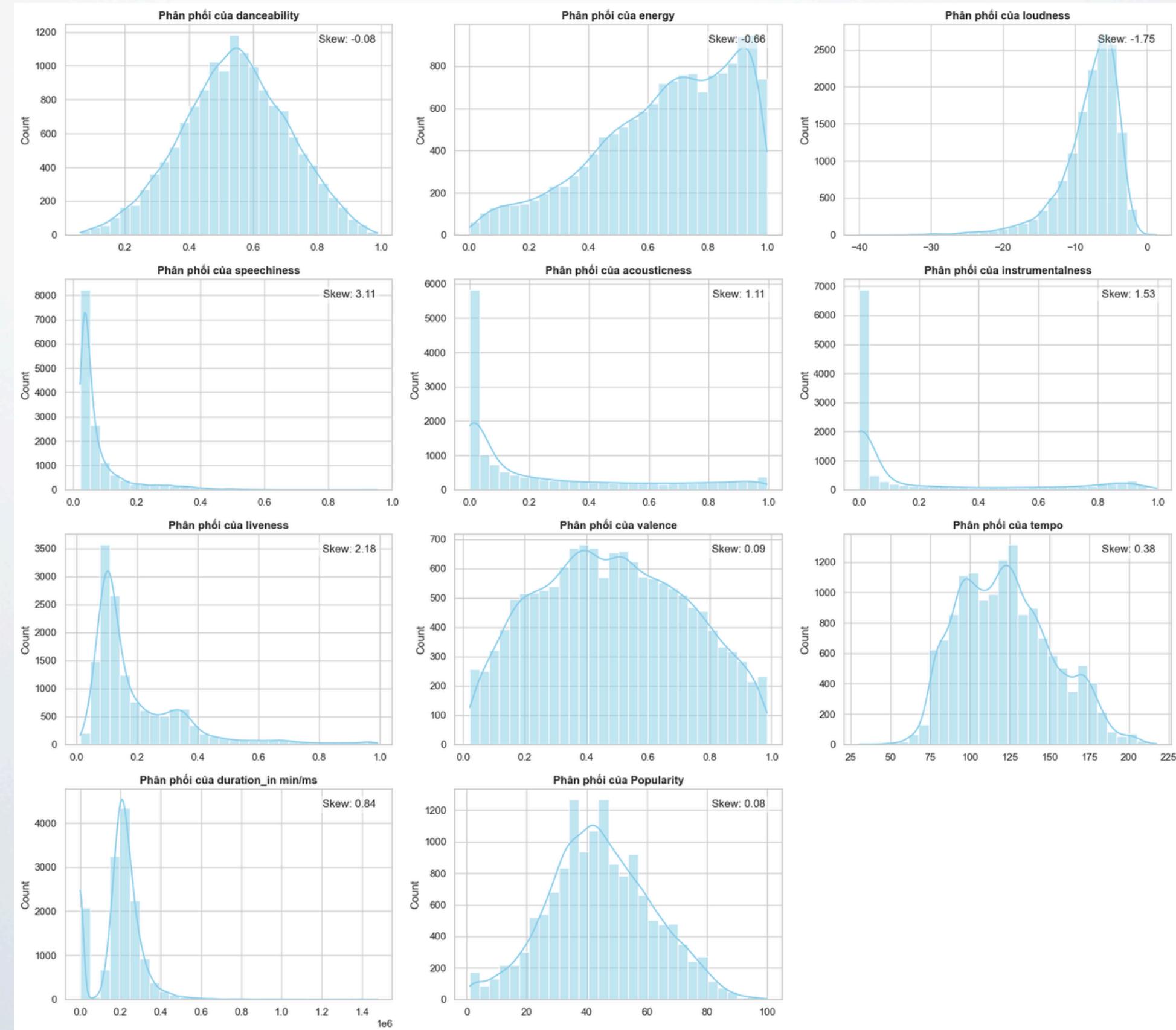


Nhận xét:

Class 5 có xu hướng sử dụng Key 1.0 một cách áp đảo so với các Key khác. Class 7 rất "kén chọn" tone nhạc. Đây là dấu hiệu tốt để phân loại. Các Key 2, 7, 9 có thể là các Tone phổ biến nhất trong âm nhạc đại chúng (ví dụ: Đô trưởng, Sol trưởng...). Cột Key 3.0 và Key 6.0 có màu khá nhạt trên toàn bộ bản đồ (trừ một chút ở Class 3).

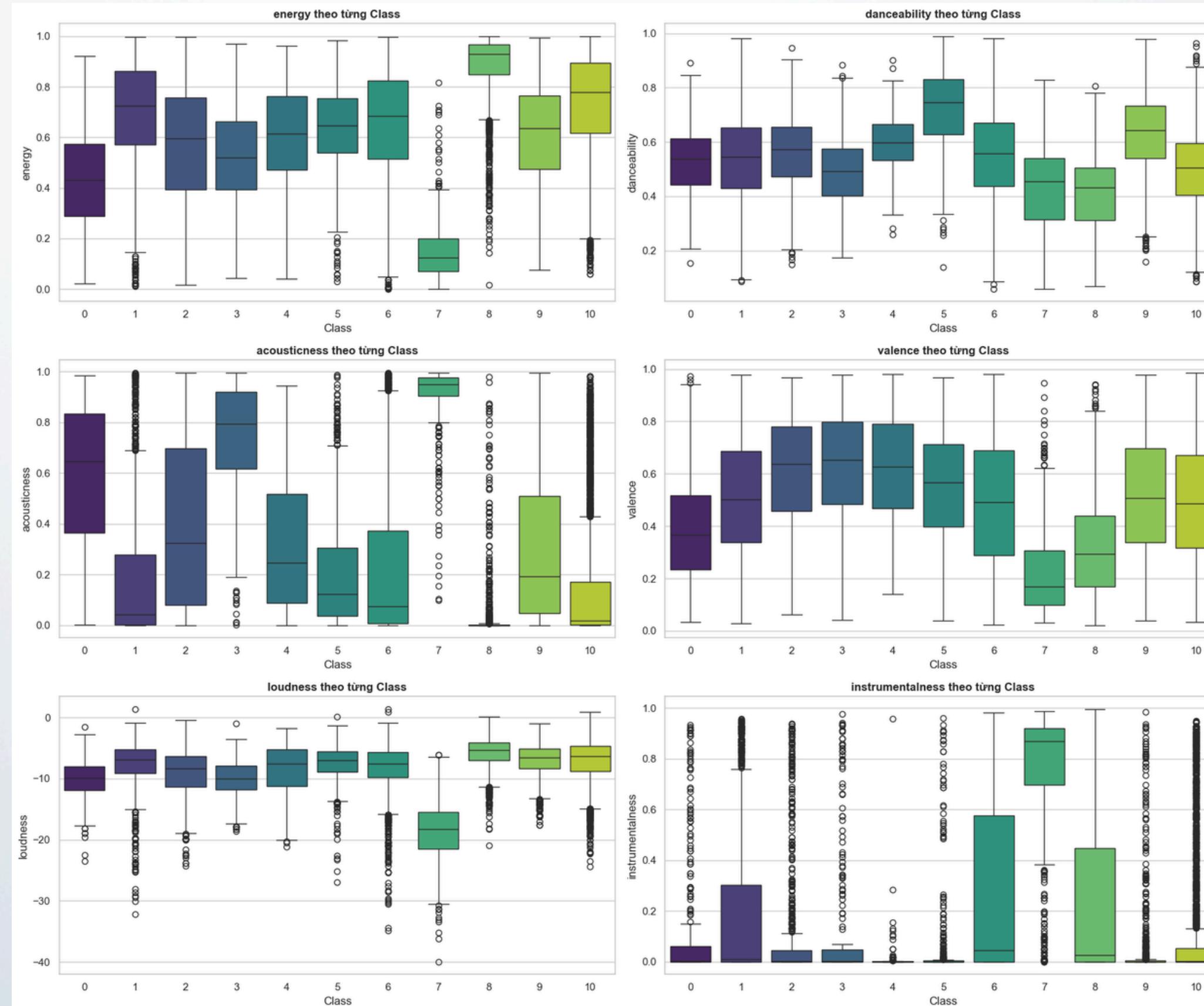
3. Phân tích khám phá dữ liệu - EDA

EDA cho biến số (Numerical)



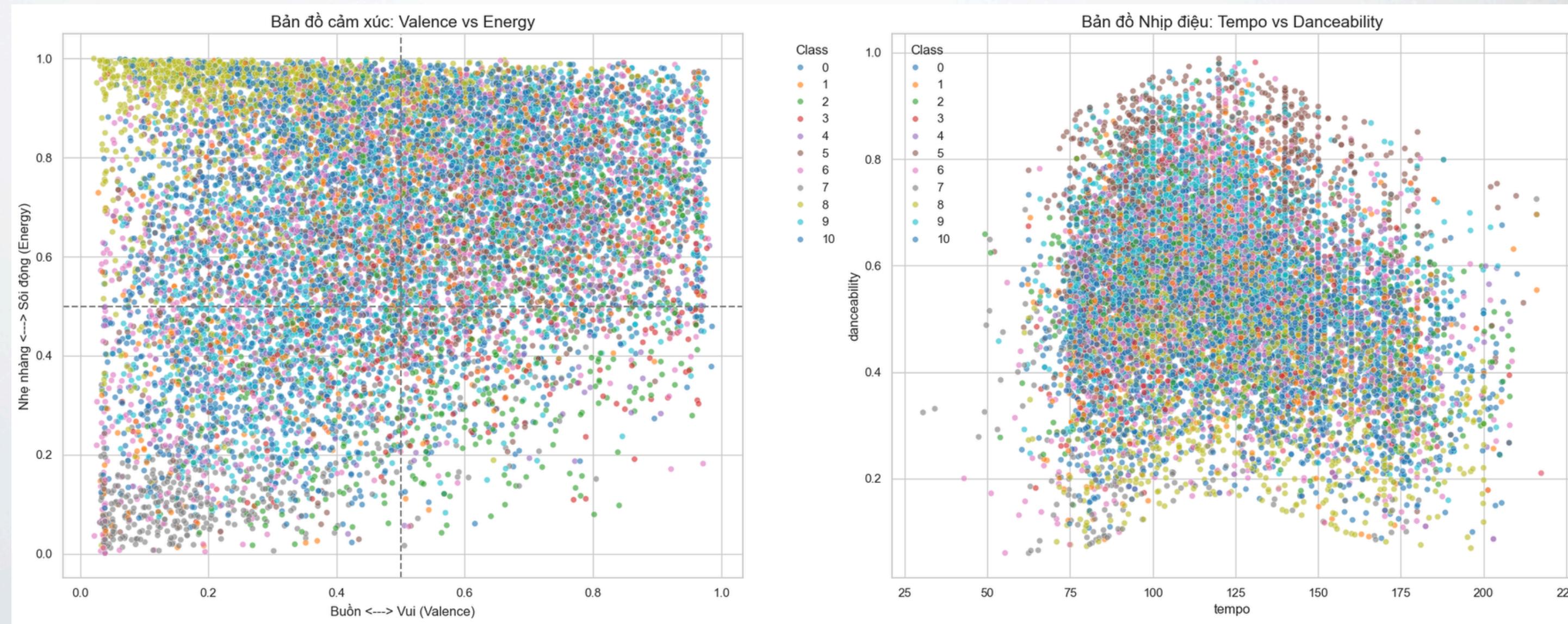
3. Phân tích khám phá dữ liệu - EDA

EDA cho biến số (Numerical)



3. Phân tích khám phá dữ liệu - EDA

EDA cho biến số (Numerical)

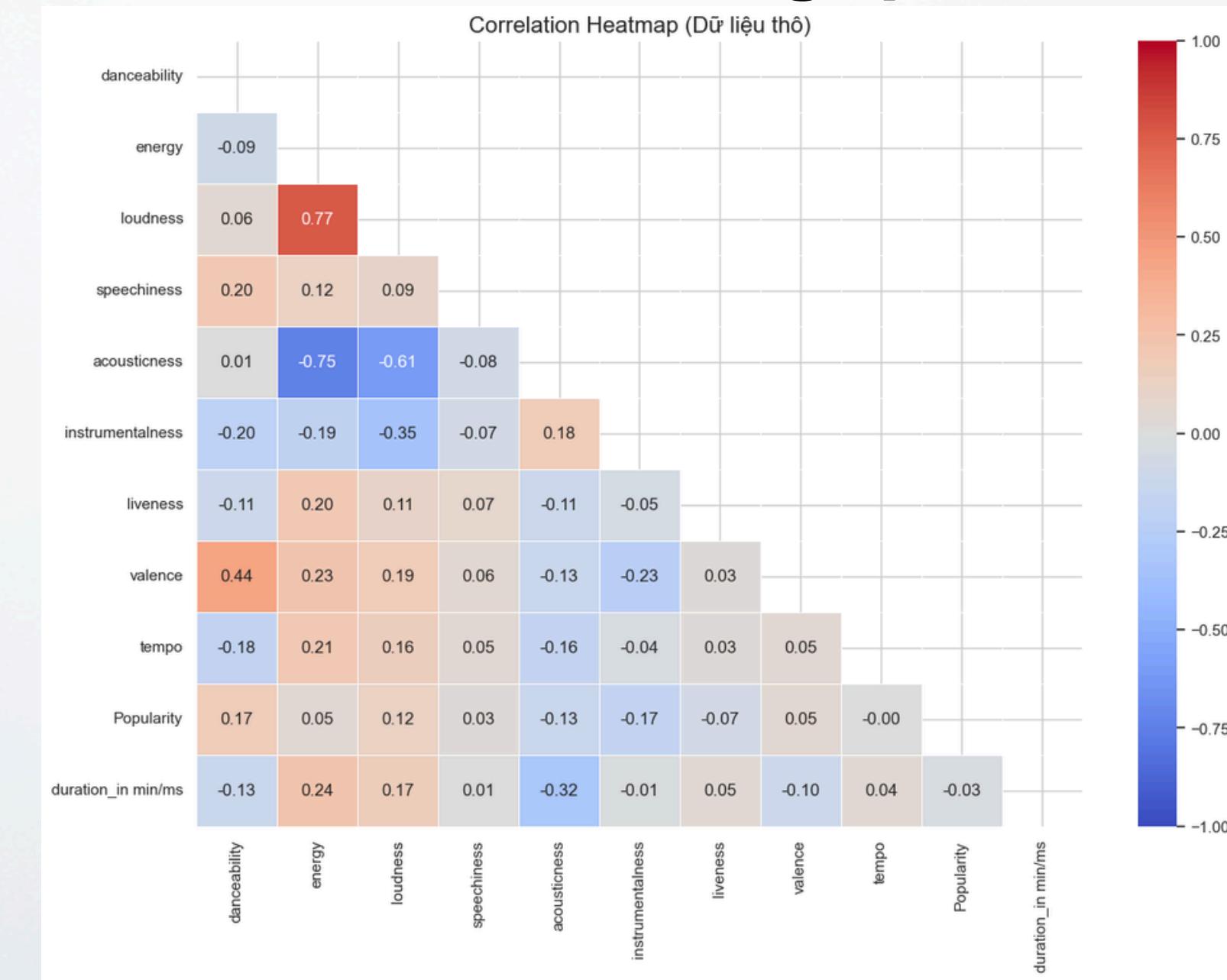


Nhận xét:

Biểu đồ phân tán phản ánh rõ dày đặc giữa đa số các Class (đặc biệt là 0, 1, 2, 8, 9, 10) tại khu vực năng lượng cao và cảm xúc tích cực (góc trên bên phải biểu đồ Valence-Energy), điều này báo hiệu các mô hình tuyến tính đơn giản sẽ gặp khó khăn lớn và cần đến các thuật toán phi tuyến tính mạnh mẽ như XGBoost hay Random Forest để phân tách ranh giới; tuy nhiên điểm sáng là Class 7 lại một lần nữa tách biệt hoàn toàn thành một "ốc đảo" riêng ở góc dưới bên trái (Low Energy, Low Valence).

3. Phân tích khám phá dữ liệu - EDA

Kiểm tra tương quan

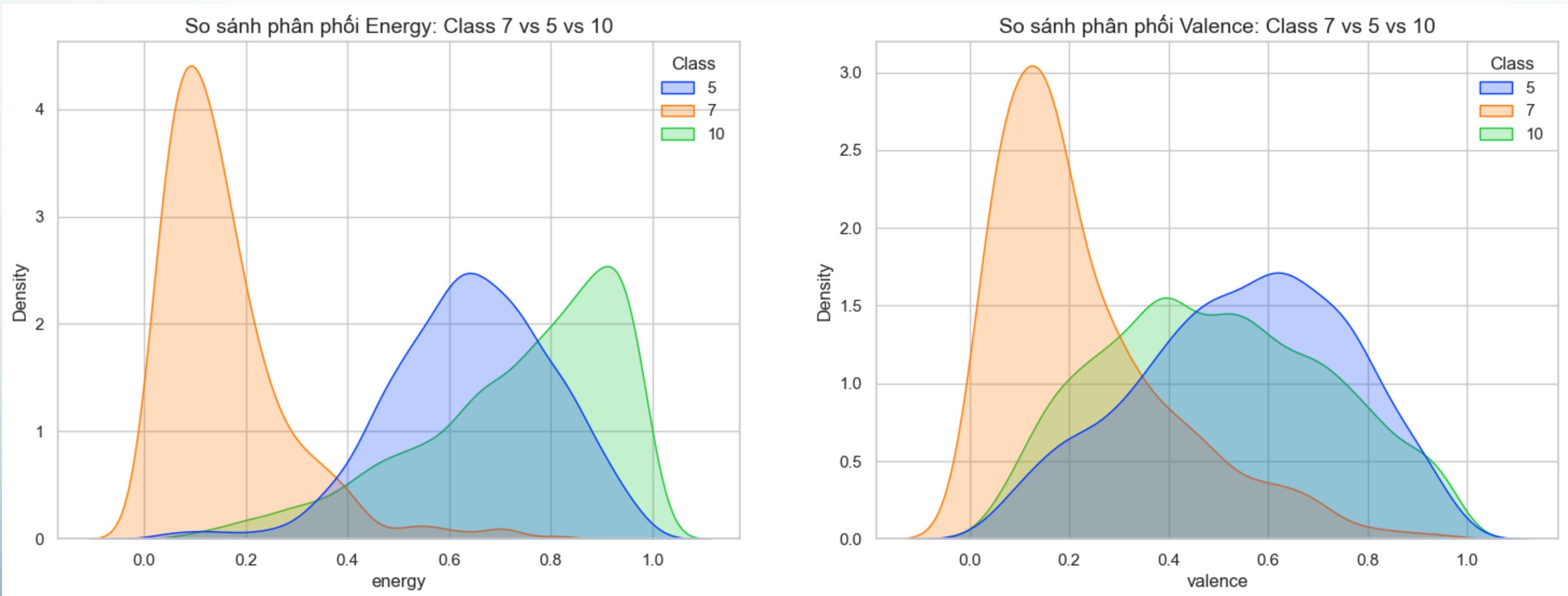


Nhận xét:

- Energy Loudness (0.77): Tương quan mạnh nhưng chưa gây đa công tuyến.
- Acousticness Energy (-0.75): Hai đặc trưng đối lập rõ rệt, rất hữu ích cho phân loại.
- Danceability Valence (0.44): Nhạc dễ nhảy thường vui hơn.
- Popularity: Gần như không liên quan đến các đặc trưng âm thanh.

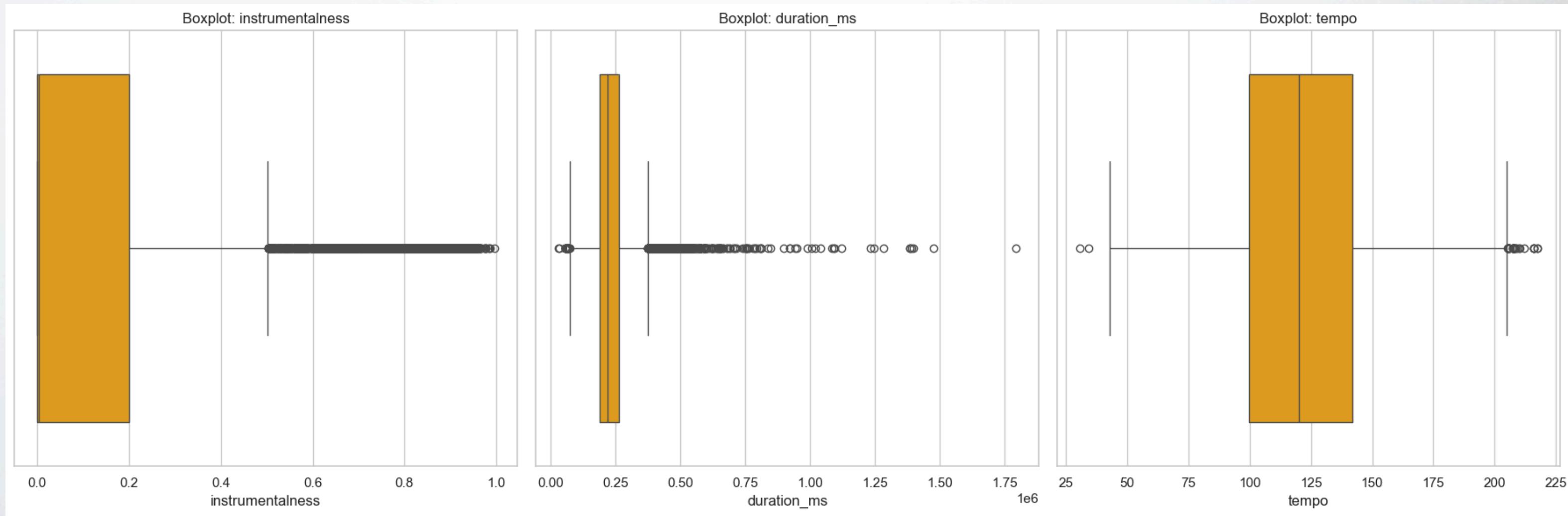
3. Phân tích khám phá dữ liệu - EDA

EDA theo từng thể loại nhạc



3. Phân tích khám phá dữ liệu - EDA

Kiểm tra outliers



Nhận xét:

- Instrumentalness (Nhạc không lời): Mặc dù tạo ra nhiều outlier giả tạo (13.48%) do phân phối lệch về 0, đặc trưng này bắt buộc phải được giữ nguyên vì nó là yếu tố cốt lõi để phân biệt nhạc có lời và nhạc không lời.
- Duration_ms (Thời lượng): Chứa outlier thực sự đáng lo ngại (bài hát dài tới 30 phút). Cần áp dụng kỹ thuật Capping (gán trần) (ví dụ: giới hạn ở 600,000ms) để ngăn các giá trị ngoại lai này làm lệch mô hình tối ưu.
- Tempo (Nhịp độ): Dữ liệu được xác định là rất sạch (chỉ 0.22% outlier) và có thể giữ nguyên mà không cần xử lý.

4. Xử lý dữ liệu

Tải Dữ liệu

Data columns (total 17 columns):			
#	Column	Non-Null Count	Dtype
0	Id	3600 non-null	int64
1	Artist Name	3600 non-null	object
2	Track Name	3600 non-null	object
3	Popularity	3505 non-null	float64
4	danceability	3600 non-null	float64
5	energy	3600 non-null	float64
6	key	3195 non-null	float64
7	loudness	3600 non-null	float64
8	mode	3600 non-null	int64
9	speechiness	3600 non-null	float64
10	acousticness	3600 non-null	float64
11	instrumentalness	2764 non-null	float64
12	liveness	3600 non-null	float64
13	valence	3600 non-null	float64
14	tempo	3600 non-null	float64
15	duration_in min/ms	3600 non-null	float64
16	time_signature	3600 non-null	int64

4. Xử lý dữ liệu

Xử lý Missing Values

Cột Popularity

- Có một số giá trị bị thiếu.
- Vì đây là biến định lượng, nhóm sử dụng median (trung vị) để thay thế → tránh bị ảnh hưởng bởi outlier.

Cột key

- Là biến phân loại có tần suất xuất hiện đều.
- Missing data rất ít.
- Được điền bằng mode (giá trị phổ biến nhất).

Cột instrumentalness

- Là biến dạng tỉ lệ [0,1].
- Hầu hết bài nhạc có instrumentalness = 0.
- Vì thế nhóm đặt mặc định missing = 0 (logic vì nhạc phổ biến thường có vocal).

Kỹ thuật được sử dụng:

- Median imputation
- Mode imputation
- Domain-based imputation (theo bản chất dữ liệu âm nhạc)

4. Xử lý dữ liệu

Xử lý dữ liệu bất thường (Data Correction)

Giá trị loudness bị dương

- Trong âm thanh:
- loudness luôn mang giá trị âm (đơn vị LUFS).
- Nếu giá trị $> 0 \rightarrow$ dữ liệu lỗi.

Giải pháp:

- `df_train['loudness'] = df_train['loudness'].clip(upper=0)`

Kết quả:

- Các điểm bất thường được kéo xuống giá trị 0.
- Phân phối loudness trở nên hợp lý hơn.
- Tránh gây nhiễu cho các mô hình nhạy với outlier như Logistic Regression, SVM.

4. Xử lý dữ liệu

Phân tích phân phối và biến đổi dữ liệu (Feature Transformation)

Một số thuộc tính âm nhạc có phân phối lệch phải (right-skewed):

- speechiness
- acousticness
- liveness
- instrumentalness

Đặc điểm:

- Nhiều bài nhạc có các chỉ số này gần 0.
- Một số ít có giá trị rất cao → gây skew.

Giải pháp: log1p transformation

- $df[col] = np.log1p(df[col])$

Hiệu quả:

- Giảm skew rõ rệt.
- Phân phối gần chuẩn hơn.
- Mô hình tuyến tính và KNN hoạt động ổn định hơn.

4. Xử lý dữ liệu

Mã hóa biến phân loại (One-Hot Encoding)

Các thuộc tính dạng categorical gồm:

- key
- mode
- time_signature

Giải pháp:

- df = pd.get_dummies(df, columns=['key','mode','time_signature'], drop_first=True)

Giải thích:

- One-Hot Encoding biến mỗi giá trị phân loại thành một cột riêng.
- Sử dụng drop_first=True để tránh bẫy đa cộng tuyến.
- Sau mã hóa, mô hình dễ dàng xử lý mối quan hệ giữa các thuộc tính âm nhạc.

4. Xử lý dữ liệu

Chuẩn hóa dữ liệu (Feature Scaling)

Các thuộc tính dạng số có đơn vị và độ lớn khác nhau, ví dụ:

- loudness (âm lượng) là số âm
- tempo (nhịp độ) có thể lên đến >200
- danceability, valence nằm trong [0,1]
- Nếu không chuẩn hóa:
 - SVM, KNN, Logistic Regression sẽ thiên vị thuộc tính có độ lớn lớn hơn.

Giải pháp: StandardScaler

- scaler = StandardScaler()
- df[scale_cols] = scaler.fit_transform(df[scale_cols])

Hiệu quả:

- Tất cả đặc trưng được đưa về trung bình 0 và độ lệch chuẩn 1.
- Mô hình học nhanh và ổn định hơn.
- Giảm rủi ro gradient vanishing/exploding trong neural networks.

4. Xử lý dữ liệu

Xử lý mất cân bằng dữ liệu (Class Imbalance Handling)

Nếu nhiệm vụ là phân loại genre hoặc phân loại mức độ phổ biến:

→ Các lớp thường không cân bằng (nhạc Pop rất nhiều, Jazz rất ít).

Giải pháp:

```
class_weight = compute_class_weight('balanced', classes, y)
```

Mục đích:

- Gán trọng số lớn hơn cho lớp ít dữ liệu.
- Tránh mô hình thiên lệch về lớp lớn.

Kỹ thuật này phù hợp với:

- Logistic Regression
- SVM
- Random Forest

4. Xử lý dữ liệu

Loại bỏ các cột không cần thiết

Ví dụ:

- Tên bài hát (name)
- Tên nghệ sĩ (artist)
- ID nhạc

Lý do:

- Không liên quan đến đặc trưng âm thanh
- Không có ý nghĩa trực tiếp trong mô hình học máy
- Tránh gây nhiễu và tạo ra sparse features quá lớn

4. Xử lý dữ liệu

Xuất dữ liệu đã xử lý

Dữ liệu cuối cùng được lưu lại:

- [./data/02_processed/train_clean.csv](#)
- [./data/02_processed/test_clean.csv](#)

Mục đích:

- Tách biệt rõ ràng dữ liệu raw và dữ liệu đã xử lý
- Dễ tái sử dụng cho bước Modeling
- Đảm bảo kết quả có thể tái lập (reproducibility)



5. Lựa chọn và huấn luyện mô hình

Chuẩn bị dữ liệu (Load & Preprocess) đã làm sạch

Các bước chính:

1. Load file train_clean.csv.

2. Loại bỏ các cột gây lỗi hoặc không dùng:

- Artist Name (text)
- Class (label)
- Id (không phải feature)

3. Tạo tập huấn luyện:

- X = train.drop(columns=[...])
- y = train["Class"]

4. Tách train/val theo:

- test_size = 0.2
- stratify = y
- random_state = 42

5. Lựa chọn và huấn luyện mô hình Định nghĩa bộ mô hình (6 models)

3 mô hình tuyến tính (Linear models) — ví dụ chuẩn theo nội dung notebook:

- Logistic Regression
- Linear SVM
- Ridge Classifier

3 mô hình phi tuyến tính (Non-linear models):

- Random Forest
- Gradient Boosting
- XGBoost

→ Các mô hình đều được đánh giá bằng K-Fold cross validation để so sánh công bằng.

5. Lựa chọn và huấn luyện mô hình Tối ưu siêu tham số bằng Optuna (Bayesian Optimization)

Nội dung chính:

1. Sử dụng Bayesian Optimization của Optuna để tìm bộ tham số tối ưu cho XGBoost.

2. Objective function gồm các hyperparameters:

- n_estimators: 200–1000
- max_depth
- learning_rate
- subsample
- colsample_bytree
- gamma, min_child_weight

3. XGBoost chạy với:

- objective = "multi:softmax"
- num_class = 11
- n_jobs = -1
- tree_method = "hist"

Kết quả:

Optuna tìm được best_params → sử dụng để train mô hình cuối cùng.

5. Lựa chọn và huấn luyện mô hình Huấn luyện mô hình tốt nhất (XGBoost)

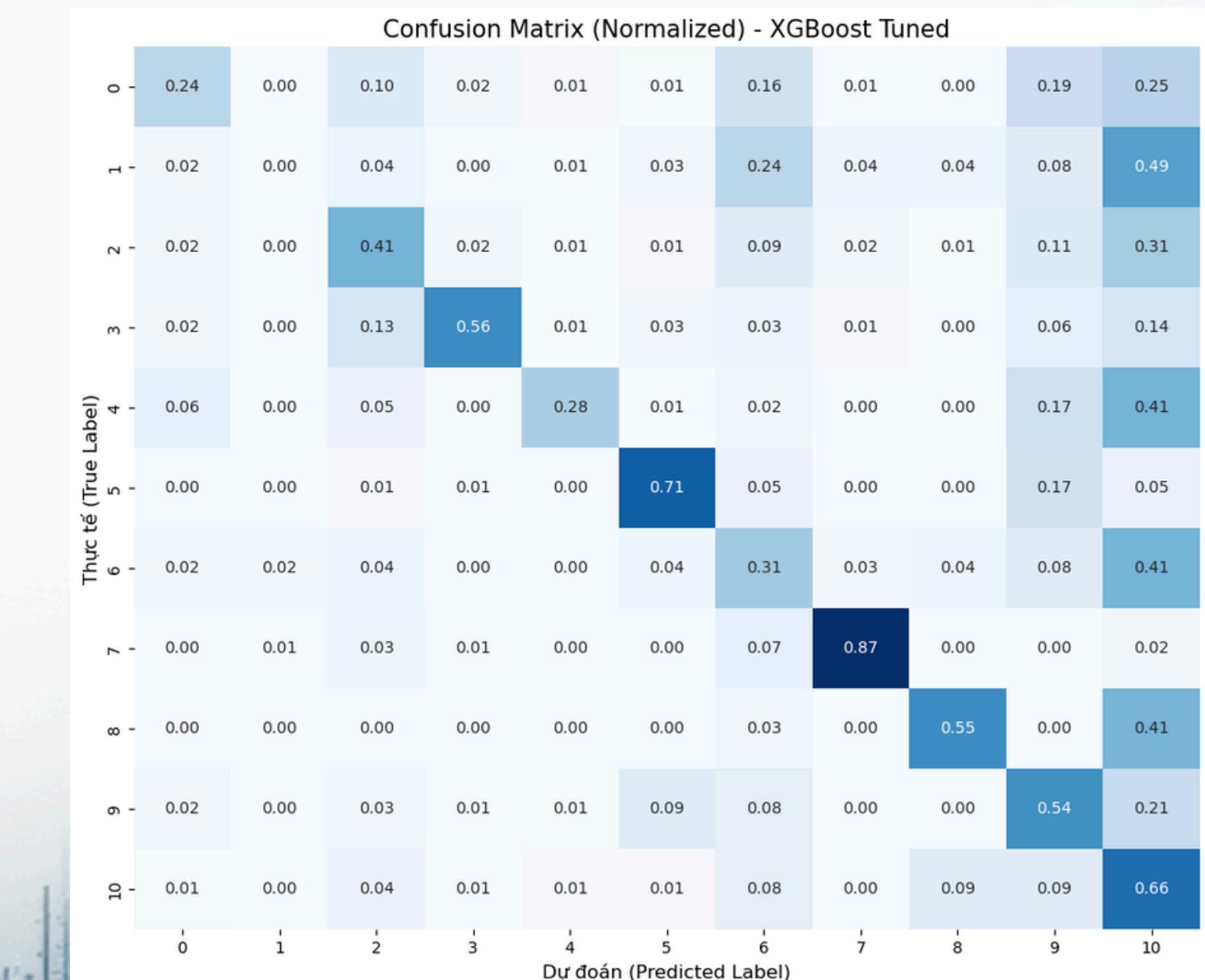
Quy trình:

1. Dùng best_params từ Optuna.
2. Train full data (không split nữa):
`final_model.fit(X, y)`

5. Lựa chọn và huấn luyện mô hình

Đánh giá hiệu suất của mô hình

===== CLASSIFICATION REPORT =====				
	precision	recall	f1-score	support
0	0.37	0.24	0.29	500
1	0.06	0.00	0.01	1098
2	0.45	0.41	0.43	1018
3	0.68	0.56	0.61	322
4	0.46	0.28	0.35	310
5	0.69	0.71	0.70	1157
6	0.38	0.31	0.35	2069
7	0.74	0.87	0.80	461
8	0.61	0.55	0.58	1483
9	0.51	0.54	0.52	2019
10	0.46	0.66	0.54	3959
accuracy			0.50	14396
macro avg	0.49	0.47	0.47	14396
weighted avg	0.47	0.50	0.48	14396



5. Lựa chọn và huấn luyện mô hình

Thực nghiệm và kết quả

Kết quả trên Kaggle

YOUR RECENT SUBMISSION

 **submission.csv**
Submitted by Thanh Thịnh Phan · Submitted 4 minutes ago

 [Jump to your leaderboard position](#)

Score: 0.51666
Public score: 0.50039

6. Đánh giá & Kết luận

Kết luận phần Modeling

Nhóm đã đánh giá 6 mô hình (tuyến tính và phi tuyến tính) bằng K-Fold Cross-Validation. XGBoost được chọn là mô hình tối ưu sau khi tinh chỉnh bằng Optuna, đạt hiệu suất F1-score cao, đặc biệt ở Class 7 và Class 5 (nhờ học tốt các đặc trưng Energy/Key). Mô hình đủ mạnh để tạo submission Kaggle, dù gặp khó khăn với Class 1 và Class 10 do đặc trưng bị trùng lặp.



6. Đánh giá & Kết luận

Định hướng cải thiện

Kỹ thuật Đặc trưng (Feature Engineering): Bổ sung đặc trưng tương tác mới (ví dụ: energy × loudness) và áp dụng chuẩn hóa/phân nhóm (binning) hoặc giảm chiều dữ liệu (PCA/UMAP) để tạo đặc trưng tổng hợp.

Xử lý Mất cân bằng Lớp: Giải quyết vấn đề dữ liệu không đồng đều (như Class 1) bằng các kỹ thuật như SMOTE/ADASYN, Focal Loss, hoặc Reweighting để tăng cường khả năng dự đoán các lớp ít mẫu.

Tập hợp Mô hình (Ensemble): Tăng độ ổn định và chính xác bằng cách kết hợp nhiều mô hình mạnh thông qua Soft Voting (XGBoost + Random Forest) hoặc Stacking.

Thử nghiệm Mô hình Nâng cao: Khám phá các mô hình học máy mạnh mẽ và nhanh hơn như LightGBM, CatBoost, hoặc triển khai Neural Networks (MLP) đơn giản.

Tối ưu Siêu tham số Sâu hơn: Mở rộng không gian tìm kiếm và tối ưu hóa đa mục tiêu (Multi-objective Optimization) bằng công cụ Optuna.

Phân tích Lỗi Chi tiết: Tiến hành kiểm tra Confusion Matrix theo từng cluster để xác định và tìm ra đặc điểm chung của các bài nhạc bị dự đoán sai (misclassify).

TÀI LIỆU THAM KHẢO

Machine Learning

1. Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd Edition). O'Reilly.
2. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
3. Hastie, T., Tibshirani, R., Friedman, J. (2017). The Elements of Statistical Learning. Springer.

XGBoost & Gradient Boosting

1. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. KDD.
2. Official XGBoost Documentation: <https://xgboost.readthedocs.io/>

Optuna – Bayesian Optimization

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework.
2. Optuna Documentation: <https://optuna.org/>

Data Imbalance

1. Chawla, N. et al. (2002). SMOTE: Synthetic Minority Over-sampling Technique.
2. Lin, T.-Y. et al. (2017). Focal Loss for Dense Object Detection. ICCV.

Spotify & Audio Feature Reference

1. Spotify for Developers – Audio Features API Documentation:
2. <https://developer.spotify.com/documentation/web-api/reference/get-audio-features>

Cảm ơn!