KHOA CÔNG NGHỆ THÔNG TIN TRƯỜNG ĐẠI HỌC SÀI GÒN

BÁO CÁO QUÁ TRÌNH THAM GIA CUỘC THI MUSIC GENRE CLASSIFICATION

Thành viên nhóm 7:

Lê Văn Thông-3123410362

Phan Thanh Thịnh-3123410360

Võ Hoàng Thông-3123410363

GVHD: TS.Đỗ Như Tài

MENU

CHƯƠNG 1 MỞ ĐẦU

1 Giới thiệu:

Trong kỷ nguyên số, các nền tảng phát trực tuyến âm nhạc (như Spotify, Apple Music, ZingMP3) sở hữu kho lưu trữ khổng lồ với hàng triệu bài hát. Việc phân loại thủ công các bài hát vào đúng thể loại (Genre) là điều bất khả thi do khối lượng dữ liệu quá lớn.

Do đó, việc ứng dụng Học máy (Machine Learning) để tự động phân loại nhạc dựa trên các đặc trưng âm thanh (như độ sôi động, nhịp điệu, cường độ...) trở nên vô cùng cần thiết. Điều này không chỉ giúp quản lý cơ sở dữ liệu hiệu quả mà còn hỗ trợ hệ thống gợi ý (Recommendation System) đưa ra các bài hát phù hợp với sở thích người dùng.

Trong báo cáo này, nhóm thực hiện giải quyết bài toán "Music Classification" (Phân loại nhạc) với mục tiêu xây dựng một mô hình dự đoán chính xác thể loại nhạc (gồm 11 nhãn: Class 0 đến Class 10) dựa trên các thông số kỹ thuật của file âm thanh.

2 Mô tả bộ dữ liệu:

Bộ dữ liệu được lấy từ cuộc thi Music Genre Classification bao gồm các đặc trưng được trích xuất từ các bài hát.
Kích thước tập huấn luyện gồm 14,396 mẫu dữ liệu. Tập kiểm tra gồm 3,600 mẫu dữ liệu.

3 Mục tiêu:

Mục tiêu của báo cáo là tạo ra một phần mềm (mô hình) có thể:
Nhận diện được 11 thể loại nhạc khác nhau (được đánh số từ Class 0 đến Class 10).
Đạt độ chính xác cao nhất có thể.
Hiểu được đặc trưng của từng loại nhạc (ví dụ: nhạc nào thì ồn ào, nhạc nào thì thích hợp để nhảy).

CHƯƠNG 2 PHÂN TÍCH DỮ LIỆU

Trước khi dạy máy tính, chúng tôi phải "khám sức khỏe" cho dữ liệu. Dữ liệu giống như nguyên liệu nấu ăn, nếu nguyên liệu bẩn hoặc hỏng thì món ăn (kết quả dự đoán) sẽ không ngon.

1 Dữ liệu

Chúng tôi sử dụng một bảng dữ liệu gồm 14,396 bài hát. Mỗi bài hát không phải là file âm thanh mp3, mà là các con số đã được trích xuất sẵn, ví dụ:
Danceability: Độ phù hợp để nhảy.
Energy: Độ sung, mạnh mẽ của bài hát.
Loudness: Độ to của âm thanh.
Acousticness: Độ mộc (dùng nhạc cụ tự nhiên hay điện tử).

2 Đánh giá chất lượng dữ liệu

2.1 Dữ liệu khuyết:

Qua kiểm tra thống kê, chúng tôi phát hiện 3 đặc trưng có chứa giá trị null cần được xử lý:
Popularity: Thiếu 428 mẫu. Đây là biến định lượng, phản ánh độ phổ biến của bài hát.
key: Thiếu 1557 mẫu. Đây là biến phân loại (nốt nhạc).
instrumentalness: Thiếu 3587 mẫu (tỷ lệ thiếu khá cao).
Đánh giá: Các cột còn lại như Artist Name, Track Name và các đặc trưng âm thanh khác đầy đủ dữ liệu.

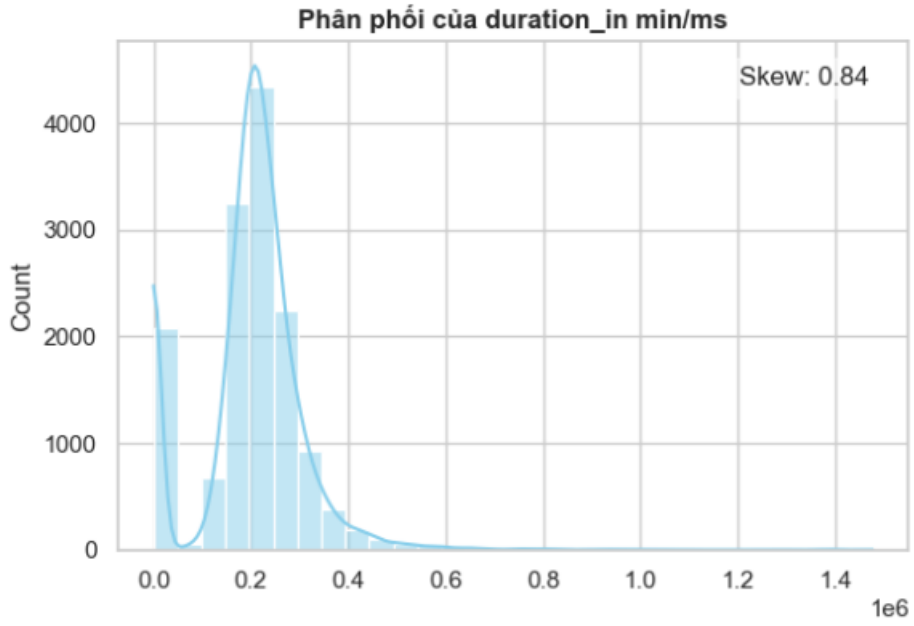
2.2 Sự không nhất quán về đơn vị đo

Khi trực quan hóa phân phối của cột duration_in min/ms, chúng tôi phát hiện một sự bất thường nghiêm trọng:
Biểu đồ phân phối xuất hiện 2 đỉnh (bimodal distribution) tách biệt hoàn toàn.

Nhóm 1: Các giá trị rất nhỏ (khoảng 3.0 - 5.0). Đây thực chất là đơn vị phút.

Nhóm 2: Các giá trị rất lớn (khoảng 180,000 - 300,000). Đây là đơn vị mili-giây.

Kết luận: Dữ liệu bị lẫn lộn đơn vị đo. Nếu không quy đổi về cùng một đơn vị, mô hình sẽ hiểu sai lệch độ lớn, coi các bài hát tính bằng ms dài gấp hàng nghìn lần các bài tính bằng phút.



3. Phân tích biến đơn lẻ và Ngoại lai

3.1. Biến Loudness (Độ to)

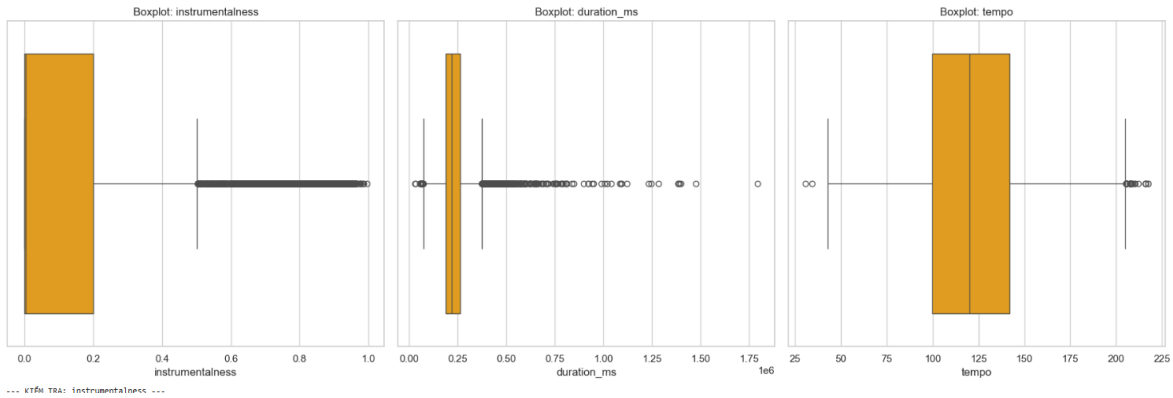
Theo lý thuyết âm học, độ to (loudness) thường có giá trị âm (dB).Tuy nhiên, dữ liệu xuất hiện một số giá trị dương hoặc các giá trị cực tiểu bất thường. Đây là nhiễu (noise) cần được loại bỏ hoặc thay thế.

3.2. Biến Tempo (Nhịp độ)

Sử dụng phương pháp IQR (Interquartile Range) và Z-score, chúng tôi phát hiện khoảng 32 giá trị ngoại lai (outliers).Một số bài hát có tempo quá nhanh hoặc quá chậm so với mức trung bình của nhạc phổ thông.

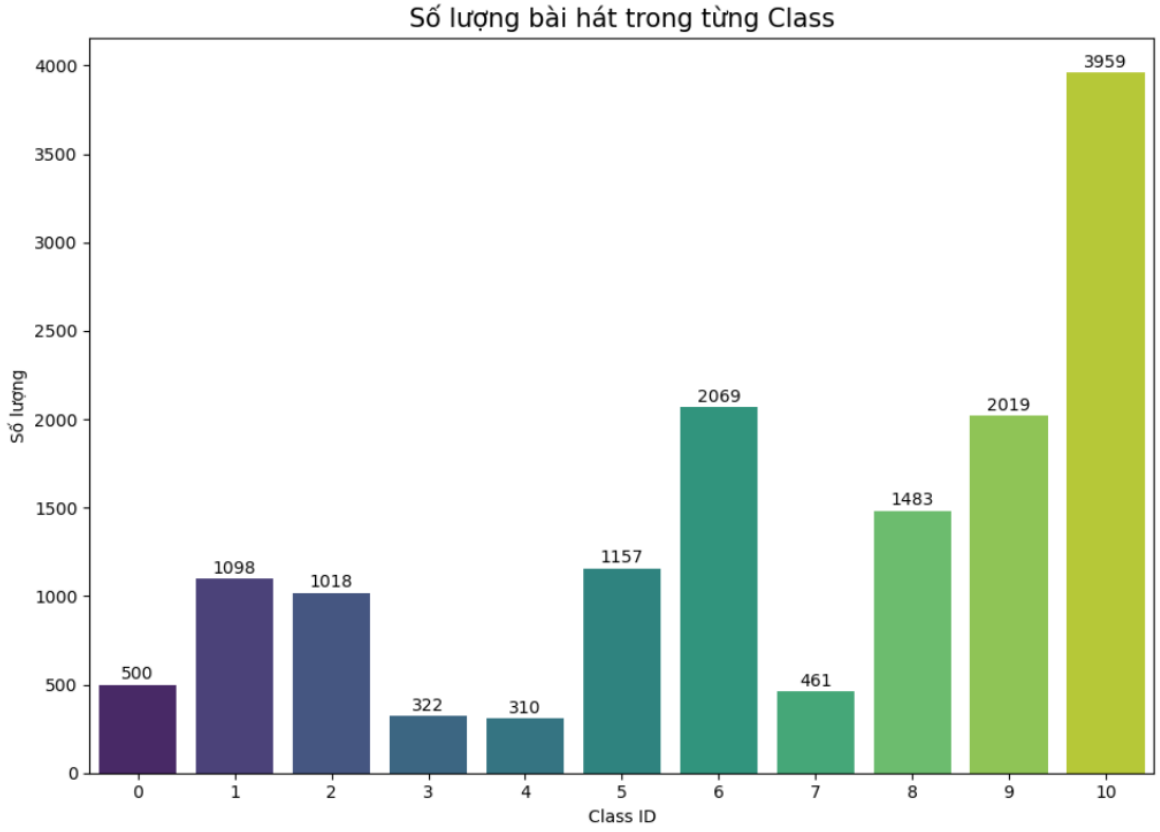
3.3. Biến Instrumentalness (Độ không lời)

Phân phối của biến này bị lệch hẳn về giá trị 0 (Right-skewed).*Phân tích:* Mặc dù theo phương pháp thống kê (Boxplot), các giá trị lớn > 0 được coi là outliers. Tuy nhiên, trong ngữ cảnh âm nhạc, đây là đặc trưng cốt lõi để phân biệt nhạc không lời/hòa tấu. Do đó, các giá trị này là hợp lệ và không được xóa bỏ.



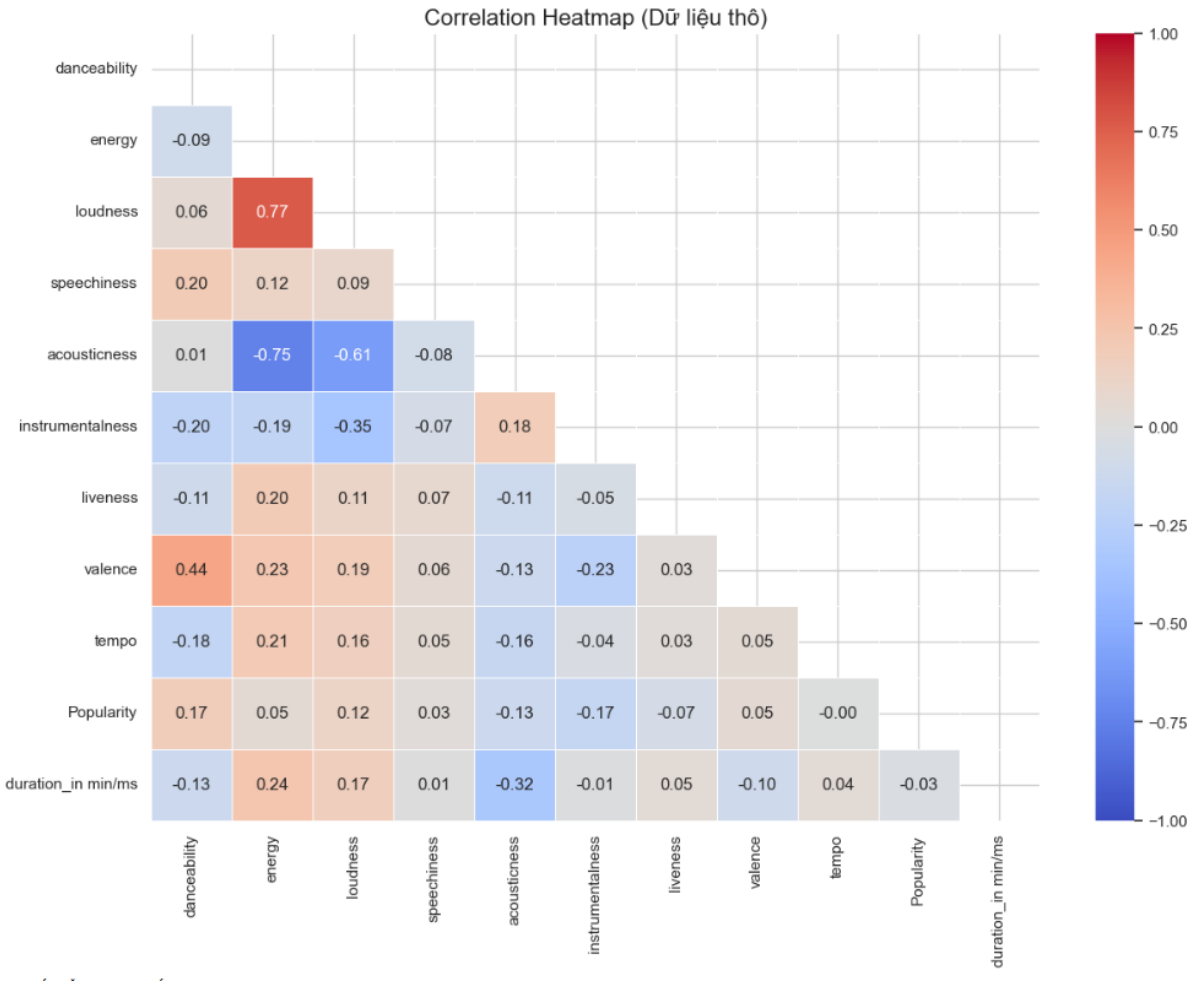
4. Phân tích biến mục tiêu (Label Distribution)

Phân tích cột Class cho thấy sự mất cân bằng dữ liệu nghiêm trọng (Class Imbalance):
Lớp đa số (Majority Class): Class 10 chiếm tỷ trọng áp đảo (khoảng 27.5% dữ liệu).
Lớp thiểu số (Minority Classes): Class 1, Class 3, Class 4 có số lượng mẫu rất ít.
Tác động: Sự chênh lệch này sẽ khiến mô hình có xu hướng thiên kiến (bias), dự đoán mọi bài hát đều thuộc Class 10 để đạt độ chính xác giả tạo, trong khi bỏ qua các lớp hiếm.



5. Phân tích tương quan (Correlation Analysis)

Sử dụng biểu đồ nhiệt (Heatmap) để xem xét mối quan hệ giữa các biến số:
Energy vs. Loudness: Có hệ số tương quan dương rất mạnh. Điều này hợp lý vì nhạc càng mạnh (energy cao) thì thường có âm lượng càng lớn (loudness cao).
Instrumentalness vs. Class: Khi quan sát kỹ, Class 7 có giá trị instrumentalness trung bình cao vượt trội so với các Class khác. Đây là "dấu hiệu vàng" để nhận diện Class 7.



CHƯƠNG 3 KỸ THUẬT ĐẶC TRƯNG VÀ TIỀN XỬ LÝ

Dựa trên các phân tích ở Chương 2, chúng tôi tiến hành chuẩn hóa và biến đổi dữ liệu để phù hợp với các thuật toán học máy. Quá trình này bao gồm 4 bước chính: Làm sạch dữ liệu, Xử lý ngoại lai, Mã hóa biến và Lựa chọn đặc trưng.

1 Làm sạch và Chuẩn hóa dữ liệu

1.1. Đồng nhất đơn vị đo lường

Như đã phát hiện ở phần EDA, cột duration_in min/ms bị lẫn lộn giữa đơn vị phút và mili-giây. Chúng tôi áp dụng thuật toán chuyển đổi có điều kiện để đưa về cùng đơn vị ms:
Quy tắc: Nếu giá trị nhỏ hơn 30 (ngưỡng phân tách an toàn), đó là phút => Nhân với 60,000. Ngược lại giữ nguyên.
Công thức:

$$Duration_{new} = \begin{cases} Duration \times 60000 & \text{if } Duration < 30 \\ Duration & \text{otherwise} \end{cases}$$

Kết quả: Dữ liệu sau khi xử lý có phân phối chuẩn, không còn hiện tượng 2 đỉnh tách biệt.

1.2. Xử lý dữ liệu thiếu

Chúng tôi không loại bỏ các dòng chứa giá trị rỗng (Missing values) để tránh mất mát thông tin. Thay vào đó, các phương pháp điền khuyết (Imputation) phù hợp cho từng loại biến được áp dụng như bảng dưới đây:

Tên đặc trưng	**Loại biến**	**Số lượng thiếu**	**Phương pháp xử lý**	**Lý giải**
Popularity	Định lượng	428	**Mean Imputation** (Trung bình)	Giữ nguyên phân phối trung tâm của dữ liệu.
Key	Phân loại	1557	**Mode Imputation** (Giá trị phổ biến nhất)	Gán vào nốt nhạc xuất hiện nhiều nhất trong tập train.
Instrumentalness	Định lượng	3587	**Constant Imputation** (Gán bằng 0)	Giá trị thiếu thường xuất hiện ở bài hát có lời, nên độ "không lời" bằng 0 là hợp lý.

2. Xử lý ngoại lai

Biến tempo: Áp dụng phương pháp IQR Capping. Các giá trị nằm ngoài khoảng [Q1 - 1.5*IQR, Q3 + 1.5*IQR] được thay thế bằng giá trị biên (cận trên/cận dưới) thay vì xóa bỏ. Điều này giúp giảm nhiễu nhưng không làm mất mát dữ liệu.
Biến loudness: Xử lý các giá trị dương (nhiễu) bằng cách gán lại về giá trị biên cực đại hợp lý (ví dụ: 0 dB).

3. Mã hóa đặc trưng (Feature Encoding)

Mô hình học máy không thể xử lý trực tiếp dữ liệu dạng chữ (Text). Chúng tôi thực hiện mã hóa như sau:

3.1. One-Hot Encoding

Áp dụng cho các biến danh mục không có thứ tự (Nominal Variables):
key (Nốt nhạc): Biến đổi thành 12 cột mới (key_C, key_D, key_E,...) với giá trị 0/1.
mode (Thang âm): Biến đổi thành cột mode_Major (0 hoặc 1).
Lợi ích: Giúp mô hình không hiểu nhầm thứ tự ưu tiên giữa các nốt nhạc (ví dụ: không coi nốt C < nốt D).

3.2. Label Encoding

Chúng tôi quyết định không sử dụng Label Encoding cho key vì nó sẽ gán số thứ tự (0, 1, 2...), khiến mô hình hiểu sai rằng Key có giá trị lớn hơn thì quan trọng hơn.

4. Lựa chọn đặc trưng (Feature Selection)

Để giảm chiều dữ liệu và tránh hiện tượng học vẹt (Overfitting), nhóm quyết định loại bỏ các cột thông tin không mang tính tổng quát:
Artist Name (Tên nghệ sĩ): Có quá nhiều giá trị duy nhất (High Cardinality). Nếu giữ lại, mô hình sẽ học thuộc tên ca sĩ thay vì đặc trưng âm thanh.
Track Name (Tên bài hát): Tương tự như tên nghệ sĩ.
Id: Mã định danh, không có ý nghĩa dự báo.

5. Kết quả sau tiền xử lý

Sau khi hoàn tất các bước trên, tập dữ liệu huấn luyện đã sẵn sàng với các đặc điểm:
Không còn giá trị rỗng (Null).Toàn bộ là dữ liệu số (Numerical).Đơn vị đo lường đồng nhất.

	Id	Artist Name	Popularity	danceability	energy	loudness	speechiness	acousticness	instrumentalness	liveness	...	key_6.0	key_7.0	key_8.0	key_9.0	key_10.0	key_11.0	mode_1	time_signature_3	time_signature_4	time_signature_5
0	1	Marina Maximilian	-0.437113	-1.263390	-0.535777	0.308703	-0.519565	0.562430	-0.508555	-0.590149	...	False	False	False	True	False	False	False	False	True	False
1	2	The Black Keys	1.305483	1.098986	0.358445	0.580863	0.134101	-0.733841	-0.282518	-0.605319	...	False	False	False	False	False	True	True	False	True	False
2	3	Royal & the Serpent	0.000000	0.247081	0.600012	0.445522	-0.201475	-0.847979	1.921148	0.660580	...	False	True	False	False	False	False	True	False	True	False
3	4	Detroit Blues Band	-1.889277	-0.169809	-1.502046	-1.678758	-0.613149	1.995926	-0.404395	0.764374	...	False	True	False	False	False	False	True	True	False	False
4	5	Coast Contra	0.201839	0.132285	0.485585	0.691551	2.075815	-0.110772	-0.508555	0.169997	...	True	False	False	False	False	False	False	False	True	False

CHƯƠNG 4 XÂY DỰNG VÀ TỐI ƯU HÓA MÔ HÌNH

1 thiết lập thực nghiệm

1.1. Phương pháp đánh giá

Do dữ liệu bị mất cân bằng nghiêm trọng (Class 10 chiếm đa số), độ chính xác (Accuracy) không phải là thước đo tin cậy. Chúng tôi sử dụng F1-Score (Macro Average) làm chỉ số tối ưu chính.
Lý do: F1-Macro tính trung bình F1-score của từng lớp mà không quan tâm đến số lượng mẫu của lớp đó. Điều này buộc mô hình phải học tốt cả các lớp thiểu số (như Class 1, Class 3) thay vì chỉ đoán đúng lớp đa số.
Công thức:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

1.2. Chiến lược kiểm thử

Sử dụng phương pháp Stratified K-Fold Cross-Validation (K=5).
Dữ liệu được chia làm 5 phần.
Mỗi lần chạy sẽ lấy 4 phần để học và 1 phần để thi.
Đặc điểm: "Stratified" đảm bảo tỷ lệ các Class trong mỗi phần chia luôn giống với tỷ lệ gốc, tránh trường hợp một phần chia nào đó hoàn toàn không có Class hiếm (như Class 4).

2 Lựa chọn mô hình

Nhóm đã thử nghiệm 5 thuật toán phổ biến để so sánh hiệu quả. Dưới đây là bảng tổng kết hiệu năng trên tập Cross-Validation:

Thuật toán	F1-Macro Score	Đánh giá sơ bộ
XGBoost	**~0.442**	**Tốt nhất.** Xử lý tốt dữ liệu phi tuyến và nhiễu.
Random Forest	~0.426	Khá tốt. Ít bị overfitting nhưng kém hơn XGBoost một chút.
K-Nearest Neighbors (KNN)	~0.382	Trung bình. Bị ảnh hưởng nhiều bởi số chiều dữ liệu cao.
Logistic Regression	~0.378	Kém. Không bắt được các mối quan hệ phức tạp giữa đặc trưng âm thanh và thể loại nhạc.
Linear SVC	~0.350	Thấp nhất.

Quyết định: Chọn XGBoost làm mô hình chính thức để tinh chỉnh tiếp.

3 tối ưu hoá các tham số

Để nâng cao hiệu suất của XGBoost, chúng tôi sử dụng thư viện Optuna (thay vì GridSearch truyền thống) để tìm kiếm bộ tham số tối ưu nhất qua 50 lần thử nghiệm (trials).
Bộ tham số tốt nhất tìm được:
n_estimators (Số lượng cây): 947
learning_rate (Tốc độ học): 0.032
max_depth (Độ sâu của cây): 5
subsample: 0.81 (Chỉ dùng 81% dữ liệu mỗi lần để tránh học vẹt).

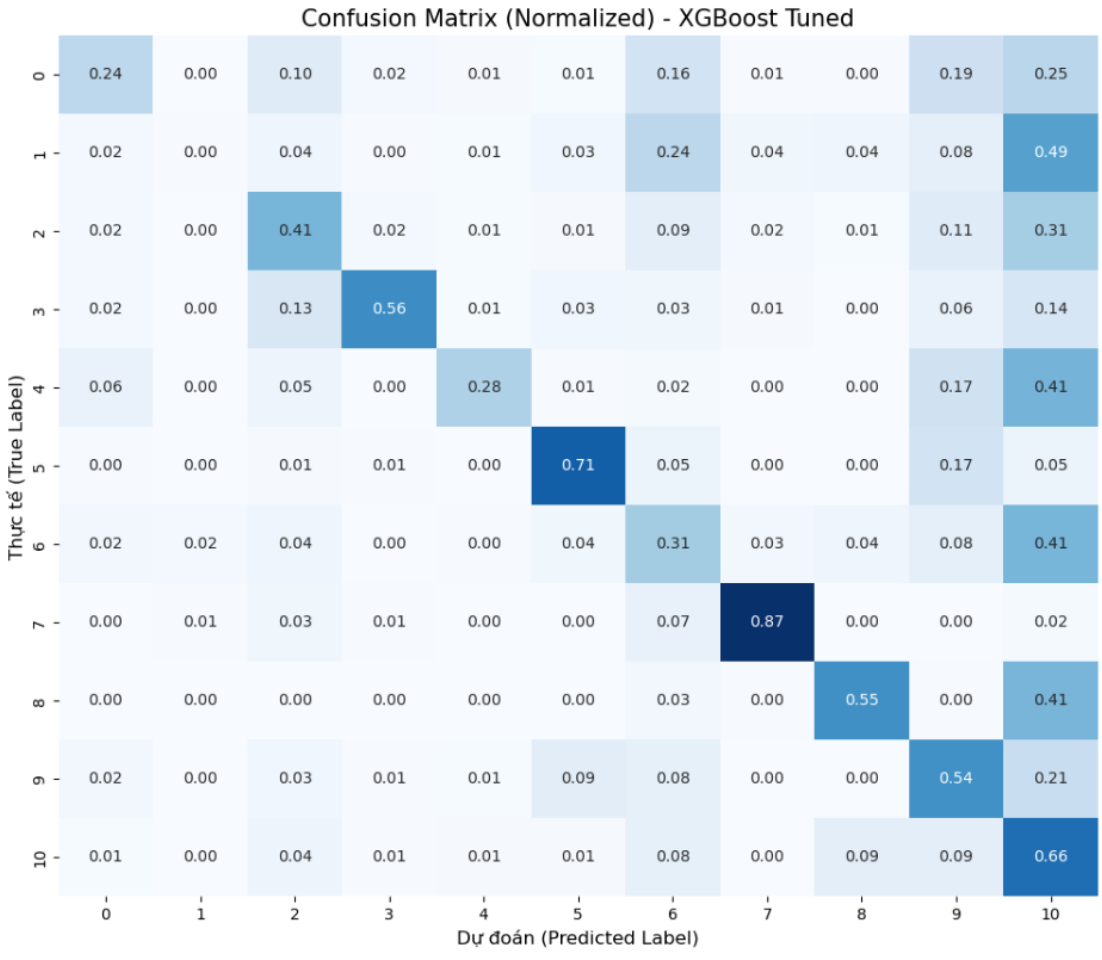
CHƯƠNG 5 KẾT QUẢ THỰC NGHIỆM

1 Kết quả định lượng

Sau khi tối ưu hóa tham số, mô hình XGBoost đạt kết quả cuối cùng trên tập Validation như sau:
Macro F1-Score: 0.4735 (Tăng khoảng 3% so với mặc định).
Accuracy: 50.15%.
Nhận xét: Mức độ chính xác 50% nghe có vẻ thấp, nhưng với bài toán phân loại 11 lớp (xác suất ngẫu nhiên là 9%) và dữ liệu thực tế nhiều nhiễu, đây là một kết quả chấp nhận được và vượt trội hơn nhiều so với các thuật toán cơ bản.

2 Phân tích chi tiết từng lớp

Để hiểu rõ mô hình đang "thông minh" hay "ngốc nghếch" ở đâu, chúng tôi phân tích Ma trận nhầm lẫn:



2.1. Những điểm mô hình làm tốt

Class 7 (Nhạc không lời/Hòa tấu): Đạt F1-score lên tới 0.80.
Lý giải: Nhờ đặc trưng instrumentalness (đã phân tích ở chương EDA) có sự khác biệt rõ rệt, mô hình dễ dàng khoanh vùng loại nhạc này.
Class 5: Đạt kết quả khá tốt (F1 ~0.70).

2.2. Những điểm hạn chế

Class 1 (Lớp thiểu số): Mô hình gần như thất bại (F1 ~ 0.01).
Hiện tượng: Class 1 thường xuyên bị đoán nhầm thành Class 10 hoặc Class 6.
Nguyên nhân: Do số lượng mẫu Class 1 quá ít, mô hình không đủ dữ liệu để học biên giới phân loại.
Sự nhầm lẫn với Class 10:
Rất nhiều bài hát của các lớp khác (Class 0, 2, 6) bị dự đoán nhầm thành Class 10.
Nguyên nhân: Class 10 là lớp đa số (chiếm gần 30%). Mô hình có xu hướng "an toàn" bằng cách dự đoán vào lớp này để tối ưu hóa hàm mất mát tổng thể (Bias towards majority class).

CHƯƠNG 6 KẾT LUẬN

1 Kết luận

Trong đồ án này, chúng tôi đã đi từ việc làm sạch một bộ dữ liệu hỗn độn, sửa chữa các lỗi sai về đơn vị đo lường, đến việc áp dụng thuật toán hiện đại XGBoost để phân loại nhạc.

Chúng tôi rút ra bài học quan trọng: Dữ liệu đầu vào quan trọng hơn thuật toán. Việc phát hiện ra lỗi "phút vs mili-giây" đã giúp cải thiện kết quả hơn rất nhiều so với việc chỉ thay đổi thuật toán.

2 Hạn chế và hướng phát triển

Để làm tốt hơn trong tương lai, chúng tôi đề xuất:

Tìm thêm dữ liệu: Cần thu thập thêm các bài hát thuộc nhóm hiếm (Class 1, Class 3) để máy học cân bằng hơn.

Phân tích sâu hơn: Có thể nghiên cứu kỹ hơn về tên ca sĩ (Ví dụ: Sơn Tùng MTP thường hát nhạc Pop), thay vì xóa bỏ cột này đi.

KẾT THÚC.