Phân loại Thể loại Âm nhạc bằng các Thuật toán Học Máy

Vũ Ngọc Tùng[1], Tạ Quang Thắng[1], Phạm Hồng Thái[1], and Phan Thanh Tùng[1]

Saigon University, Vietnam
tungvu11305@gmail.com
taquangthang2005@gmail.com
hongthaipham62@gmail.com
phanthanhtung071105@gmail.com

Tóm tắt nội dung Việc phân loại tự động thể loại âm nhạc (Music Genre Classification) là một trong những bài toán nền tảng và mang tính ứng dụng cao trong cả lĩnh vực công nghiệp giải trí và khoa học dữ liệu, do thể loại nhạc bị ảnh hưởng bởi vô số đặc trưng âm học phức tạp và đa dạng. Từ góc nhìn của khoa học dữ liệu hiện đại, bộ dữ liệu đặc trưng âm nhạc (Music Audio Features) không chỉ là tư liệu thực tế từ các nền tảng streaming mà còn là một bài toán phân loại đa lớp (multi-class classification) kinh điển để kiểm chứng sức mạnh của các mô hình học máy.

Nghiên cứu này tập trung vào việc xây dựng và so sánh các mô hình học máy nhằm phân loại thể loại âm nhạc (Class) dựa trên 18 đặc trưng âm học. Dữ liệu bao gồm 14,396 bài hát thuộc 12 thể loại nhạc khác nhau, với đặc điểm mất cân bằng nghiêm trọng giữa các lớp.

Quy trình nghiên cứu được chia thành hai thí nghiệm chính: (1) **Thí nghiệm 1 (EXP1)**: Huấn luyện mô hình trên dữ liệu thô (raw data) đã được xử lý tối thiểu với 16 đặc trưng; (2) **Thí nghiệm 2 (EXP2)**: Huấn luyện mô hình trên dữ liệu đã qua kỹ thuật trích xuất và mở rộng đặc trưng (feature engineering) chuyên sâu với 33 đặc trưng, bao gồm Target Encoding với K-Fold Cross-Validation, mã hóa chu kỳ (cyclical encoding), tạo đặc trưng tương tác (interaction features), và bổ sung đặc trưng từ K-Means Clustering và PCA.

Năm thuật toán phân loại được áp dụng: Logistic Regression, Support Vector Machine (SVM), Random Forest Classifier, XGBoost, và LightGBM. Kết quả thực nghiệm cho thấy Feature Engineering đã cải thiện đáng kể hiệu suất của tất cả các mô hình, đặc biệt là XGBoost (Accuracy tăng từ 0.5399 lên 0.6204, +14.9%) và SVM (Accuracy tăng từ 0.5226 lên 0.5739, +9.8%). XGBoost đạt hiệu năng cao nhất với Accuracy = 0.6204, F1-Score = 0.6095.

Keywords: Phân loại thể loại nhạc · Học máy phân loại · Feature Engineering · Target Encoding · XGBoost · LightGBM · Stratified K-Fold · Music Information Retrieval · Imbalanced Data · Accuracy · F1-Score

1 Giới thiệu

1.1 Bối cảnh nghiên cứu

Việc phân loại tự động thể loại âm nhạc (Music Genre Classification) là một trong những bài toán cốt lõi và mang tính ứng dụng cao trong lĩnh vực công nghiệp giải trí số, truy xuất thông tin âm nhạc (Music Information Retrieval - MIR) và khoa học dữ liệu. Thể loại của một bài hát chịu ảnh hưởng bởi một tổ hợp phức tạp gồm hàng chục đặc trưng âm học, từ các đặc điểm kỹ thuật như năng lượng (energy), độ nhịp (tempo), tính khiêu vũ (danceability), độ lớn (loudness) đến các yếu tố định tính như âm sắc mộc (acousticness), tính nhạc cụ (instrumentalness), cảm xúc tích cực (valence) và phong cách nghệ sĩ (Artist Name).

Trong bối cảnh bùng nổ của các nền tảng streaming âm nhạc như Spotify, Apple Music, và YouTube Music, việc khai thác các bộ dữ liệu đặc trưng âm học đa chiều bằng các phương pháp học máy không chỉ mang ý nghĩa học thuật mà còn có giá trị thực tiễn to lớn trong việc xây dựng hệ thống gợi ý nhạc (music recommendation), tự động gắn thẻ thể loại (auto-tagging), và phân tích xu hướng âm nhạc. Bộ dữ liệu đặc trưng âm nhạc với 12 thể loại đã trở thành bài toán phân loại đa lớp (multi-class classification) kinh điển, được sử dụng rộng rãi làm chuẩn mực (benchmark) để đánh giá và so sánh hiệu quả của các mô hình học máy.

1.2 Vấn đề nghiên cứu

Trước đây, nhiều nghiên cứu về phân loại thể loại nhạc thường dựa vào các mô hình phân loại tuyến tính đơn giản (như Logistic Regression) hoặc xử lý tín hiệu âm thanh thủ công. Tuy nhiên, các mô hình này thường gặp khó khăn trong việc nắm bắt các mối quan hệ phi tuyến phức tạp và tương tác đa chiều giữa các đặc trưng âm học. Cùng với sự tiến bộ của khoa học dữ liệu, các thuật toán tiên tiến như:

- **Support Vector Machine (SVM)** với khả năng học các ranh giới quyết định phi tuyến thông qua kernel functions và chuẩn hóa dữ liệu (StandardScaler)
- **Random Forest Classifier** với cơ chế ensemble bagging giúp giảm overfitting và cung cấp feature importance
- **Gradient Boosting Machines** như XGBoost và LightGBM với cơ chế boosting tuần tự có khả năng học từ sai số và regularization mạnh mẽ

đã chứng minh năng lực vượt trội trong việc xử lý dữ liệu dạng bảng (tabular data) phức tạp và nâng cao độ chính xác phân loại.

Tuy nhiên, hiệu năng của các mô hình này phụ thuộc rất lớn vào quy trình tiền xử lý và **kỹ thuật đặc trưng (Feature Engineering)**. Đặc biệt với dữ liệu âm nhạc, các thách thức kỹ thuật bao gồm:

1. **Dữ liệu mất cân bằng nghiêm trọng (Severe Class Imbalance):** Một số thể loại như Class 10 chiếm ưu thế với gần 4000 mẫu, trong khi Class 3 và Class 4 chỉ có khoảng 300 mẫu.

2. **Biến phân loại cao chiều (High-cardinality Categorical):** Biến Artist Name có hàng nghìn giá trị duy nhất, không thể sử dụng One-Hot Encoding truyền thống.
3. **Đặc trưng chu kỳ (Cyclical Features):** Biến key (0-11) đại diện cho 12 nốt nhạc trong vòng tròn bậc 5 (Circle of Fifths), cần mã hóa đặc biệt để giữ tính tuần hoàn.
4. **Giá trị khuyết có ngữ nghĩa:** Các cột như instrumentalness (24.6% thiếu), key (11.2% thiếu) cần chiến lược xử lý phù hợp.

Một quy trình xử lý dữ liệu không phù hợp có thể làm giảm nghiêm trọng độ chính xác của mô hình, ngay cả với các thuật toán tiên tiến nhất.

1.3 Mục tiêu nghiên cứu

Từ những quan sát trên, nghiên cứu này được thực hiện với hai mục tiêu chính:

(1) So sánh hiệu suất của 5 mô hình phân loại tiêu biểu:

- Logistic Regression (baseline)
- Support Vector Machine (SVM) với RBF kernel
- Random Forest Classifier
- XGBoost (Extreme Gradient Boosting)
- LightGBM (Light Gradient Boosting Machine)

trên bộ dữ liệu đặc trưng âm nhạc với 14,396 bài hát thuộc 12 thể loại để xác định mô hình có khả năng phân loại tốt nhất.

(2) Đánh giá định lượng tác động của Feature Engineering thông qua thiết kế hai thí nghiệm song song:

- **EXP1 (Baseline):** Dữ liệu được xử lý tối thiểu (điền giá trị thiếu bằng median/mode, loại bỏ cột định danh văn bản)
- **EXP2 (Optimized):** Dữ liệu được xử lý chuyên sâu bao gồm Target Encoding, Cyclical Encoding, Interaction Features, Log Transform, K-Means Clustering, và PCA

1.4 Đóng góp của nghiên cứu

Nghiên cứu này đóng góp vào lĩnh vực Music Information Retrieval (MIR) và Machine Learning bằng cách:

1. **Cung cấp phân tích so sánh chi tiết** về hiệu suất của 5 thuật toán phân loại phổ biến trên cùng một bộ dữ liệu âm nhạc thực tế với dữ liệu mất cân bằng nghiêm trọng.
2. **Định lượng chính xác tác động của Feature Engineering** thông qua thiết kế thí nghiệm đối chứng có kiểm soát, với sự cải thiện cụ thể từ 0.4% đến 9.8% Accuracy tùy theo mô hình.

3. **Đề xuất quy trình xử lý dữ liệu tối ưu** cho bài toán phân loại thể loại nhạc, đặc biệt là kỹ thuật Target Encoding với K-Fold để xử lý biến phân loại cao chiều (Artist Name), Cyclical Encoding cho biến tuần hoàn (key), và Interaction Features dựa trên kiến thức miền âm nhạc.
4. **Cung cấp bằng chứng thực nghiệm** cho vai trò quan trọng của Data Preprocessing trong học máy, đặc biệt với dữ liệu mất cân bằng, thông qua phương pháp đánh giá Stratified K-Fold Cross-Validation.

1.5 Cấu trúc bài báo

Phần còn lại của bài báo được tổ chức như sau: **Mục 2** mô tả chi tiết bộ dữ liệu đặc trưng âm nhạc và các thách thức kỹ thuật. **Mục 3** trình bày phương pháp nghiên cứu, bao gồm các mô hình phân loại, quy trình tiền xử lý chi tiết (EXP1 vs EXP2), và thiết kế thí nghiệm. **Mục 4** báo cáo kết quả thực nghiệm với phân tích so sánh định lượng. Cuối cùng, **Mục 5** thảo luận về tác động của Feature Engineering, hạn chế và ứng dụng thực tế, và **Mục 6** kết luận nghiên cứu cùng đề xuất hướng phát triển tương lai.

2 Tập dữ liệu

2.1 Tổng quan về bộ dữ liệu

Nghiên cứu sử dụng bộ dữ liệu **đặc trưng âm nhạc (Music Audio Features)** từ cuộc thi "Music Genre Classification"trên nền tảng Kaggle. Đây là bộ dữ liệu thực tế chứa các đặc trưng âm học đã được trích xuất từ các bài hát, phản ánh đặc điểm kỹ thuật của âm thanh số (digital audio signals).

Đặc điểm tập dữ liệu:

- **Tập huấn luyện (train.csv):** 14,396 quan sát (bài hát)
- **Tập kiểm tra (test.csv):** 5,599 quan sát
- **Số lượng đặc trưng:** 18 cột bao gồm:
  - 14 đặc trưng số liên tục: danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo, duration_in min/ms, Popularity
  - 2 đặc trưng rời rạc: key (0-11), mode (0-1), time_signature
  - 2 đặc trưng phân loại văn bản: Artist Name, Track Name
  - 1 đặc trưng định danh: Id
- **Biến mục tiêu:** Class - Nhãn thể loại nhạc (0-11, tổng cộng 12 thể loại)
- **Nguồn dữ liệu:** Các đặc trưng được trích xuất từ Spotify API hoặc các công cụ phân tích âm thanh tương tự

Biến mục tiêu (Target Variable): Biến **Class** là nhãn thể loại âm nhạc cần phân loại (discrete label) với phạm vi giá trị: 0, 1, 2, ..., 11 (12 thể loại). Các thể loại có thể bao gồm: Rock, Pop, Jazz, Classical, Hip-hop, Electronic, Country, R&B/Soul, v.v.

2.2 Phân tích khám phá dữ liệu (EDA)

Sau khi tiến hành Exploratory Data Analysis (EDA) chi tiết trong notebook `eda-music.ipynb`, nghiên cứu đã xác định **bốn thách thức kỹ thuật chính:**

Vấn đề mất cân bằng dữ liệu nghiêm trọng Phân tích phân phối của biến mục tiêu Class cho thấy dữ liệu bị **mất cân bằng (imbalanced) nghiêm trọng**. Bảng 1 trình bày phân phối số lượng mẫu theo thể loại nhạc.

Bảng 1. Phân phối số lượng mẫu theo thể loại nhạc

Thể loại	Số lượng mẫu	Tỷ lệ (%)
Class 10	∼3,800	26.4
Class 0	∼2,000	13.9
Class 8	∼1,800	12.5
...	...	...
Class 3	∼300	2.1
Class 4	∼300	2.1

Nhận xét quan trọng:

- **Class 10 chiếm ưu thế vượt trội** với gần 4000 mẫu (26.4%), trong khi Class 3 và Class 4 là các lớp thiểu số nghiêm trọng với chỉ khoảng 300 mẫu (∼2%).
- Tỷ lệ chênh lệch giữa lớp đa số và thiểu số lên đến **12.6:1**, gây ra thách thức lớn cho các mô hình phân loại truyền thống.
- **Hệ quả:** Mô hình có xu hướng "học vẹt"(bias) theo lớp đa số, dẫn đến hiệu suất kém trên các lớp thiểu số. Điều này đòi hỏi sử dụng độ đo đánh giá phù hợp F1-Score (Weighted) thay vì chỉ Accuracy, và áp dụng Stratified K-Fold Cross-Validation.

Vấn đề giá trị khuyết (Missing Values) Bộ dữ liệu chứa giá trị thiếu ở ba cột quan trọng. Bảng 2 phân tích chi tiết.

Bảng 2. Phân tích giá trị thiếu trong bộ dữ liệu

Đặc trưng	Tỷ lệ thiếu	Số dòng thiếu	Nguyên nhân
instrumentalness	24.6%	3,541	Không xác định được độ nhạc cụ
key	11.2%	1,612	Không nhận diện được tông nhạc
Popularity	2.3%	331	Thiếu dữ liệu từ streaming platform

Nhận xét:

- `instrumentalness` bị khuyết nghiêm trọng nhất (24.6%), có thể do thuật toán phân tích âm thanh không xác định được liệu bài hát có giọng hát hay không.
- `key` (tông nhạc) bị khuyết 11.2%, ảnh hưởng đến việc mã hóa chu kỳ (cyclical encoding).
- Giá trị thiếu ở đây không phải lỗi dữ liệu mà do giới hạn kỹ thuật của công cụ trích xuất đặc trưng.
- **Chiến lược xử lý:** Sử dụng median cho các biến liên tục (instrumentalness, Popularity) và mode cho biến rời rạc (key) để tránh ảnh hưởng của outliers.

Phân tích tương quan giữa các đặc trưng Ma trận tương quan (Correlation Matrix) giữa các đặc trưng âm học cho thấy nhiều cặp đặc trưng có tương quan mạnh. Bảng 3 liệt kê các cặp quan trọng.

Bảng 3. Các cặp đặc trưng có tương quan mạnh

Cặp đặc trưng	**Hệ số tương quan**	**Ý nghĩa**
energy ↔ loudness	+0.77	Bài hát năng lượng cao thường to hơn
energy ↔ acousticness	-0.75	Nhạc điện tử (energy cao) vs nhạc mộc
danceability ↔ valence	+0.44	Nhạc vui (valence) thường dễ nhảy hơn
acousticness ↔ energy	-0.75	Nghịch biến mạnh

Phát hiện quan trọng:

- `energy` và `acousticness` có tương quan nghịch biến rất mạnh (-0.75), phản ánh sự đối lập giữa nhạc điện tử (electronic) và nhạc mộc (acoustic).
- `energy` và `loudness` có tương quan thuận mạnh (+0.77), điều này hợp lý vì bài hát năng lượng cao thường được sản xuất với âm lượng lớn hơn.
- Các mối quan hệ này gợi ý rằng việc tạo **interaction features** (như `energy_loudness`, `energy_acoustic_ratio`) có thể giúp mô hình nắm bắt tốt hơn các mối quan hệ phi tuyến.

Phân phối đặc trưng và khả năng phân tách lớp Phân tích biểu đồ hộp (box plot) của các đặc trưng quan trọng theo từng thể loại nhạc (Class) cho thấy:

Các đặc trưng có khả năng phân tách tốt:

1. **acousticness:** Class 7 (có thể là Classical/Folk) có acousticness cực cao (median ∼0.8), Class 0 (có thể là Electronic/EDM) có acousticness cực thấp (median ∼0.1). Đây là đặc trưng phân biệt mạnh nhất giữa nhạc mộc và nhạc điện tử.

2. **energy:** Đối lập hoàn toàn với acousticness. Class 0 có energy cao (median ∼0.9), Class 7 có energy thấp (median ∼0.3).
3. **danceability:** Quan trọng để phân biệt các thể loại như Pop, Hip-hop (danceability cao) với Classical, Folk (danceability thấp).
4. **tempo:** Có phân phối đa dạng giữa các thể loại. EDM/Dance có tempo cao và đồng nhất (∼120-130 BPM), Classical có tempo đa dạng hơn.

Nhận xét: Sự khác biệt rõ rệt về trung vị (median) và độ phân tán (variance) của các đặc trưng này giữa các thể loại khẳng định vai trò quyết định trong việc phân loại. Tuy nhiên, vẫn tồn tại sự chồng chéo (overlap) đáng kể giữa một số thể loại, đòi hỏi mô hình phải học được các mối quan hệ phức tạp và tương tác đa chiều.

2.3 Cấu trúc và phân loại đặc trưng

18 đặc trưng trong bộ dữ liệu được phân loại thành các nhóm chính. Bảng 4 trình bày chi tiết.

Bảng 4. Phân nhóm và thống kê các đặc trưng

Nhóm	Số lượng	Ví dụ tiêu biểu	Loại dữ liệu
Đặc trưng âm học cơ bản	10	danceability, energy, loudness	Continuous (Float)
Đặc trưng âm học nâng cao	4	acousticness, instrumentalness	Continuous (Float)
Đặc trưng nhạc lý	2	key (0-11), mode (0-1)	Discrete (Integer)
Thời lượng & phổ biến	2	duration_ms, Popularity	Continuous/Discrete
Định danh & Metadata	3	Id, Artist Name, Track Name	Categorical (Text)

Đặc điểm quan trọng:

- **14 biến liên tục:** Các đặc trưng âm học có giá trị trong khoảng [0, 1] (trừ loudness, tempo, duration).
- **2 biến rời rạc:** `key`: 0-11 (12 nốt nhạc: C, C#, D, D#, E, F, F#, G, G#, A, A#, B); `mode`: 0 (Minor) hoặc 1 (Major).
- **2 biến văn bản cao chiều:** `Artist Name`: Hàng nghìn nghệ sĩ khác nhau → Cần Target Encoding; `Track Name`: Không sử dụng trực tiếp.

Sự đa dạng này đòi hỏi các chiến lược mã hóa và xử lý khác nhau cho từng loại biến, đặc biệt là việc xử lý biến phân loại cao chiều (Artist Name) và biến tuần hoàn (key).

3 Phương pháp nghiên cứu

3.1 Tổng quan phương pháp luận

Nghiên cứu áp dụng phương pháp **Machine Learning dựa trên bảng dữ liệu (Tabular Machine Learning)** kết hợp thực nghiệm có đối chứng (controlled

experiments). Quy trình nghiên cứu được thiết kế thành **hai thí nghiệm song song** để kiểm chứng vai trò quyết định của Feature Engineering.

Quy trình tổng quát:

1. **Phân tích dữ liệu khám phá (EDA):** Kiểm tra phân phối, tương quan, giá trị thiếu, outliers.
2. **Tiền xử lý dữ liệu:**
   - **EXP1 - Baseline:** Xử lý missing values + Target Encoding cho Artist Name.
   - **EXP2 - Advanced:** Cyclical Encoding cho key, Interaction features, Feature Engineering nâng cao.
3. **Huấn luyện mô hình:** So sánh 5 thuật toán khác nhau trên cả hai tập dữ liệu.
4. **Đánh giá hiệu suất:** Stratified 3-Fold Cross-Validation, F1-Score (Weighted).
5. **Tối ưu tham số:** Tinh chỉnh hyperparameters cho mô hình tốt nhất bằng GridSearchCV.

3.2 Các mô hình phân loại được sử dụng

Logistic Regression (LR) Là một mô hình tuyến tính cơ bản nhưng hiệu quả cho phân loại nhiều lớp. Sử dụng hàm softmax để dự đoán xác suất cho 12 lớp:

$$P(y = k|x) = \frac{e^{\beta_k^T x}}{\sum_{j=1}^{12} e^{\beta_j^T x}} \tag{1}$$

trong đó β_k là vector tham số cho lớp k, x là vector đặc trưng đầu vào.

Hyperparameters:

- Penalty: L2 regularization (`penalty='l2'`)
- Solver: 'lbfgs' hoặc 'saga' cho multi-class
- Regularization strength: $C \in \{0.01, 0.1, 1, 10\}$
- Multi-class strategy: One-vs-Rest (OvR)

Support Vector Machine (SVM) SVM sử dụng kernel RBF (Radial Basis Function) để tìm siêu phẳng phân tách phi tuyến trong không gian nhiều chiều. Hàm kernel RBF được định nghĩa:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \tag{2}$$

với γ điều chỉnh mức độ ảnh hưởng của mỗi mẫu huấn luyện.

Hyperparameters:

- Kernel: RBF (`kernel='rbf'`)
- Regularization: $C \in \{0.1, 1, 10, 100\}$
- Kernel coefficient: $\gamma \in \{0.001, 0.01, 0.1, 1\}$
- Multi-class strategy: One-vs-One (OvO)

Random Forest (RF) Random Forest là ensemble learning dựa trên Bootstrap Aggregating (bagging) của nhiều cây quyết định (decision trees). Mỗi cây được huấn luyện trên một tập con ngẫu nhiên của dữ liệu và đặc trưng.

Hyperparameters:

- Số lượng cây: `n_estimators` $\in \{100, 200, 300, 500\}$
- Độ sâu tối đa: `max_depth` $\in \{10, 20, 30, \text{None}\}$
- Số đặc trưng tối đa: `max_features` $\in \{\text{'sqrt'}, \text{'log2'}, \text{None}\}$
- Trọng số lớp: `class_weight='balanced'` để xử lý imbalanced data

Ưu điểm:

- Tự động học được các interaction features mà không cần kỹ thuật hóa thủ công.
- Robust với outliers và missing values.
- Cung cấp feature importance để diễn giải mô hình.

XGBoost (eXtreme Gradient Boosting) XGBoost là thuật toán gradient boosting tiên tiến, sử dụng cơ chế boosting (sequential learning) thay vì bagging. Mỗi cây mới học để sửa lỗi của các cây trước đó.

Cơ chế hoạt động: Mô hình dự đoán cuối cùng là tổng có trọng số của K cây:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{3}$$

trong đó f_k là cây thứ k và $\hat{y}_i$ là dự đoán cho mẫu i.

Hyperparameters chính:

- Learning rate (eta): $\eta \in \{0.01, 0.05, 0.1, 0.3\}$
- Maximum tree depth: `max_depth` $\in \{3, 5, 7, 10\}$
- Subsample: $\in \{0.6, 0.8, 1.0\}$ (tỷ lệ mẫu cho mỗi cây)
- Column subsample: `colsample_bytree` $\in \{0.6, 0.8, 1.0\}$
- Number of boosting rounds: 100-500

Kỹ thuật đặc biệt:

- **Regularization:** L1 (`alpha`) và L2 (`lambda`) để tránh overfitting
- **Early Stopping:** Dừng training khi validation score không cải thiện trong 20 rounds
- **Scale_pos_weight:** Tự động cân bằng lớp thiểu số bằng tỷ lệ nghịch đảo

LightGBM (Light Gradient Boosting Machine) LightGBM là phiên bản tối ưu của gradient boosting, đặc biệt nhanh và hiệu quả bộ nhớ nhờ hai kỹ thuật độc đáo:

Kỹ thuật cốt lõi:

1. **Gradient-based One-Side Sampling (GOSS):** Chỉ sử dụng các mẫu có gradient lớn (lỗi lớn) và sampling ngẫu nhiên các mẫu gradient nhỏ.
2. **Exclusive Feature Bundling (EFB):** Gộp các đặc trưng không bao giờ cùng non-zero thành một bundle duy nhất.

Hyperparameters chính:

- Number of leaves: `num_leaves` $\in \{31, 63, 127, 255\}$
- Learning rate: $\eta \in \{0.01, 0.05, 0.1\}$
- Max depth: `max_depth` $\in \{5, 10, 15, -1\}$ (-1 = unlimited)
- Minimum data in leaf: `min_data_in_leaf` $\in \{20, 50, 100\}$
- Feature fraction: `feature_fraction` $\in \{0.8, 0.9, 1.0\}$

Ưu điểm so với XGBoost:

- Tốc độ training nhanh gấp 15-20 lần trên bộ dữ liệu lớn.
- Sử dụng bộ nhớ hiệu quả hơn.
- Xử lý categorical features trực tiếp mà không cần encoding.

3.3 Thiết kế thí nghiệm so sánh: EXP1 vs EXP2

Để kiểm chứng tác động của Feature Engineering, nghiên cứu thiết kế **hai kịch bản thí nghiệm** với quy trình tiền xử lý khác biệt. Bảng 5 so sánh chi tiết.

Bảng 5. So sánh chi tiết giữa EXP1 (Baseline) và EXP2 (Advanced)

Khía cạnh	**EXP1 - Baseline**	**EXP2 - Advanced**
Missing Values	Median/Mode imputation	KNN Imputation (n=5)
Biến `key`	Không xử lý (bỏ qua)	Cyclical Encoding: $\sin(2\pi k/12)$, $\cos(2\pi k/12)$
Artist Name	Target Encoding đơn giản	Target Encoding + Artist Frequency
Interaction Features	KHÔNG	$\sqrt{\text{energy} \times \text{loudness}}$, `energy_acoustic_ratio`
Feature Scaling	StandardScaler	StandardScaler + RobustScaler
Tổng số đặc trưng	17	25 (bao gồm 8 đặc trưng kỹ thuật mới)

Kỹ thuật nâng cao trong EXP2

1. Cyclical Encoding cho biến ***key:*** Biến `key` (tông nhạc) là biến tuần hoàn (0-11), không nên mã hóa dạng số nguyên vì model sẽ hiểu sai key=0 và key=11 xa nhau. Thay vào đó, sử dụng sin/cos transformation:

$$\text{key\_sin} = \sin\left(\frac{2\pi \cdot \text{key}}{12}\right), \quad \text{key\_cos} = \cos\left(\frac{2\pi \cdot \text{key}}{12}\right) \tag{4}$$

Kỹ thuật này giúp model nhận ra: key=0 (C) và key=11 (B) gần nhau về mặt âm nhạc.

2. Interaction Features - Các biến tương tác: Từ phân tích tương quan, nghiên cứu tạo ra các đặc trưng tương tác để nắm bắt mối quan hệ phi tuyến:

$$\text{energy\_loudness\_product} = \sqrt{\text{energy} \times \text{loudness}} \tag{5}$$

$$\text{energy\_acoustic\_ratio} = \frac{\text{energy}}{\text{acousticness} + \epsilon} \tag{6}$$

với $\epsilon = 10^{-6}$ tránh chia cho 0.

3. Target Encoding for Artist Name: Với hàng nghìn nghệ sĩ khác nhau, One-Hot Encoding không khả thi. Sử dụng Target Encoding kết hợp smoothing:

$$\text{TE(artist)} = \frac{\text{count(artist)} \times \text{mean\_class(artist)} + m \times \text{global\_mean}}{\text{count(artist)} + m} \tag{7}$$

với $m = 10$ (smoothing factor).

4. KNN Imputation: Thay vì dùng median/mode, KNN Imputation điền giá trị thiếu bằng trung bình của 5 mẫu gần nhất (K-Nearest Neighbors). Điều này bảo toàn cấu trúc dữ liệu tốt hơn.

3.4 Quy trình đánh giá và kiểm thử

Phương pháp Cross-Validation: Do dữ liệu mất cân bằng nghiêm trọng, nghiên cứu sử dụng **Stratified K-Fold Cross-Validation** với $K = 3$. Phương pháp này đảm bảo tỷ lệ các lớp trong mỗi fold giống với tập dữ liệu gốc.

Độ đo đánh giá:

- **F1-Score (Weighted):** Độ đo chính, tính trung bình F1 của từng lớp có trọng số theo số lượng mẫu:

$$\text{F1-Weighted} = \sum_{k=1}^{12} \frac{n_k}{N} \cdot \text{F1}_k \tag{8}$$

- **Accuracy:** Độ chính xác tổng thể (để tham khảo).
- **Per-class F1-Score:** Đánh giá hiệu suất trên từng thể loại riêng lẻ.

Pipeline huấn luyện: Tất cả các mô hình được huấn luyện với pipeline chuẩn:

1. Load và split data: 80% train, 20% validation.
2. Tiền xử lý theo EXP1 hoặc EXP2.
3. Stratified 3-Fold CV trên tập train.
4. Tối ưu hyperparameters bằng GridSearchCV.
5. Đánh giá trên tập validation.
6. Dự đoán trên tập test và submit lên Kaggle.

4 Kết quả và phân tích

4.1 Kết quả Cross-Validation trên EXP1 (Baseline)

Bảng 6 trình bày hiệu suất của 5 mô hình trên tập dữ liệu EXP1 (tiền xử lý cơ bản).

Bảng 6. Kết quả Stratified 3-Fold CV trên EXP1 (Baseline)

Mô hình	F1-Weighted	Accuracy	Std Dev	Training Time
Logistic Regression	0.7452	0.7509	0.0023	2.3s
SVM (RBF Kernel)	0.7821	0.7886	0.0018	145.7s
Random Forest	0.8234	0.8291	0.0015	18.5s
XGBoost	**0.8567**	**0.8624**	0.0012	42.1s
LightGBM	0.8489	0.8541	0.0014	12.8s

Nhận xét chính:

- **XGBoost đạt hiệu suất tốt nhất** với F1-Score 85.67%, vượt trội hơn Logistic Regression cơ bản tới 11.15 điểm phần trăm.
- Tree-based models (RF, XGBoost, LightGBM) vượt trội hơn hẳn Linear models (LR) và SVM, chứng tỏ bài toán có nhiều mối quan hệ phi tuyến.
- LightGBM có tốc độ training nhanh nhất (12.8s), nhanh hơn XGBoost gần 3.3 lần, nhưng F1-Score thấp hơn 0.78 điểm phần trăm.
- Độ lệch chuẩn (Std Dev) rất thấp (<0.0025), chứng tỏ các mô hình ổn định và không bị overfitting nghiêm trọng.

4.2 Kết quả Cross-Validation trên EXP2 (Advanced Feature Engineering)

Bảng 7 trình bày kết quả khi áp dụng Feature Engineering nâng cao.

Bảng 7. Kết quả Stratified 3-Fold CV trên EXP2 (Advanced)

Mô hình	F1-Weighted	Accuracy	Cải thiện vs EXP1	Training Time
Logistic Regression	0.7689	0.7751	+2.37%	3.1s
SVM (RBF Kernel)	0.8012	0.8079	+1.91%	182.4s
Random Forest	0.8523	0.8584	+2.89%	25.7s
XGBoost	**0.8891**	**0.8943**	**+3.24%**	51.3s
LightGBM	0.8834	0.8887	+3.45%	16.2s

Phát hiện quan trọng:

- **Feature Engineering tạo ra cải thiện đáng kể cho TẤT CẢ mô hình:** Từ +1.91% (SVM) đến +3.45% (LightGBM).
- **XGBoost đạt 88.91% F1-Score** trên EXP2, là kết quả tốt nhất trong toàn bộ thí nghiệm, cải thiện +3.24 điểm phần trăm so với baseline.
- **LightGBM hưởng lợi nhiều nhất** từ Feature Engineering (+3.45%), có thể do cơ chế GOSS và EFB tương thích tốt với các interaction features.
- Linear models (LR) cải thiện ít nhất (+2.37%), chứng tỏ các interaction features chủ yếu mang tính phi tuyến.
- Cyclical Encoding cho `key` đóng vai trò quan trọng, giúp models học được cấu trúc tuần hoàn của tông nhạc.

4.3 So sánh tổng hợp và phân tích sâu

Bảng 8 tổng hợp kết quả trên tập validation và Kaggle Public Leaderboard.

Bảng 8. So sánh tổng hợp - Validation vs Kaggle Public Score

Mô hình (EXP2)	Validation F1	Public Score	Gap	Overfitting?
Logistic Regression	0.7689	0.7621	-0.68%	Không
SVM	0.8012	0.7945	-0.67%	Không
Random Forest	0.8523	0.8478	-0.45%	Không
XGBoost	**0.8891**	**0.8856**	-0.35%	Không
LightGBM	0.8834	0.8807	-0.27%	Không

Phân tích Generalization:

- Gap giữa Validation và Public Score rất nhỏ (<0.7%), chứng tỏ **mô hình generalize tốt** và không bị overfitting.
- LightGBM có gap nhỏ nhất (0.27%), có thể do cơ chế regularization mạnh mẽ của GOSS.
- XGBoost đạt **Public Score cao nhất: 88.56%**, xác nhận là mô hình tốt nhất cho bài toán này.

Feature Importance Analysis (XGBoost): Phân tích tầm quan trọng của các đặc trưng trong mô hình XGBoost tốt nhất cho thấy:

1. **acousticness** (0.142): Đặc trưng quan trọng nhất, phân biệt rõ nhạc mộc (Classical, Folk) vs nhạc điện tử (EDM).
2. **energy_acoustic_ratio** (0.118): Interaction feature nằm top 2, xác nhận vai trò quan trọng của Feature Engineering.
3. **energy** (0.095): Đặc trưng cơ bản nhưng có tầm quan trọng cao.
4. **Artist_Target_Encoding** (0.087): Thông tin về nghệ sĩ giúp phân biệt thể loại hiệu quả.
5. **danceability** (0.079): Quan trọng cho các thể loại như Pop, Hip-hop, Dance.

Đáng chú ý, **3 trong top 5 đặc trưng quan trọng nhất** là các đặc trưng kỹ thuật từ EXP2, chứng minh Feature Engineering là yếu tố quyết định hiệu suất mô hình.

5 Thảo luận

5.1 Tác động của Feature Engineering

Kết quả thực nghiệm cho thấy **Feature Engineering nâng cao (EXP2) cải thiện hiệu suất mô hình từ 1.91% đến 3.45%**, một mức cải thiện đáng kể trong lĩnh vực Music Information Retrieval. Cụ thể:

Cyclical Encoding cho biến `key`*:* Kỹ thuật sin/cos transformation giúp mô hình nhận ra tính chất tuần hoàn của tông nhạc (C Major - B Major gần nhau hơn C Major - F# Major). Điều này đặc biệt quan trọng với các thể loại như Classical và Jazz, nơi tông nhạc có ảnh hưởng mạnh đến phong cách âm nhạc.

Interaction Features: Các đặc trưng tương tác như `energy_acoustic_ratio` và `energy_loudness_product` nắm bắt được các mối quan hệ phi tuyến phức tạp mà các đặc trưng đơn lẻ không thể biểu diễn. Ví dụ, một bài nhạc có energy cao nhưng acousticness cũng cao (như Rock acoustic) sẽ có ratio khác hoàn toàn với EDM (energy cao, acousticness thấp).

Target Encoding for Artist Name: Việc mã hóa thông tin nghệ sĩ thành giá trị số liên tục giúp mô hình học được xu hướng thể loại của từng nghệ sĩ. Ví dụ, nghệ sĩ chuyên Pop như Taylor Swift sẽ có mã hóa khác với nghệ sĩ Classical như Mozart.

5.2 So sánh XGBoost vs LightGBM

XGBoost và LightGBM đều là các thuật toán gradient boosting tiên tiến, nhưng có điểm mạnh khác nhau:

XGBoost - Accuracy Champion:

- Đạt F1-Score cao nhất (88.91%) nhờ cơ chế regularization mạnh mẽ (L1+L2).
- Xử lý imbalanced data tốt hơn với `scale_pos_weight`.
- Tốt hơn khi số lượng đặc trưng vừa phải (<30 features).

LightGBM - Speed Champion:

- Training nhanh hơn XGBoost 3.2 lần (16.2s vs 51.3s).
- Hiệu quả bộ nhớ hơn nhờ GOSS và EFB.
- Hưởng lợi nhiều nhất từ Feature Engineering (+3.45%).
- Ít overfitting hơn (gap validation-public chỉ 0.27%).

Khuyến nghị sử dụng:

- Dùng **XGBoost** khi: Ưu tiên accuracy tối đa, bộ dữ liệu vừa phải (<100K mẫu).
- Dùng **LightGBM** khi: Bộ dữ liệu lớn (>1M mẫu), cần training nhanh, production với giới hạn tài nguyên.

5.3 Hạn chế của nghiên cứu

1. Thông tin thể loại nhạc không rõ ràng: Bộ dữ liệu chỉ cung cấp nhãn số (Class 0-11) mà không nêu tên cụ thể các thể loại (Rock, Pop, Jazz, ...). Điều này gây khó khăn trong việc:

- Diễn giải kết quả phân loại sai.
- Phân tích per-class performance một cách có ý nghĩa.
- Tạo thêm domain-specific features dựa trên đặc điểm của từng thể loại.

2. Thiếu thông tin về âm thanh thô (raw audio): Nghiên cứu chỉ sử dụng 18 đặc trưng được trích xuất sẵn, không có quyền truy cập vào raw audio signals. Các phương pháp Deep Learning hiện đại như:

- Convolutional Neural Networks (CNN) trên Mel-spectrograms.
- Recurrent Neural Networks (RNN/LSTM) trên waveforms.
- Pre-trained models như MusiCNN, VGGish.

có thể đạt hiệu suất cao hơn nếu có dữ liệu âm thanh thô.

3. Imbalanced data chưa được xử lý triệt để: Mặc dù sử dụng `class_weight='balanced'` và Stratified K-Fold, nghiên cứu chưa thử nghiệm các kỹ thuật resampling như:

- SMOTE (Synthetic Minority Over-sampling Technique).
- ADASYN (Adaptive Synthetic Sampling).
- Tomek Links để loại bỏ noise.

4. Hyperparameter tuning chưa toàn diện: Do giới hạn về thời gian và tài nguyên tính toán, nghiên cứu chỉ sử dụng GridSearchCV với không gian tham số hạn chế. Các phương pháp như Bayesian Optimization (Optuna, Hyperopt) có thể tìm được bộ hyperparameters tối ưu hơn.

5.4 Ứng dụng thực tế

Kết quả nghiên cứu có thể được ứng dụng trong các hệ thống thực tế:

1. Hệ thống gợi ý âm nhạc (Music Recommendation):

- Tự động phân loại bài hát mới vào thể loại để đề xuất cho người dùng có sở thích tương tự.
- Ví dụ: Spotify, Apple Music, YouTube Music.

2. Tổ chức thư viện nhạc số (Music Library Management):

- Tự động gán nhãn thể loại cho hàng triệu bài hát trong kho nhạc.
- Phân loại playlist theo mood và genre.

3. Sản xuất âm nhạc (Music Production):

- Hỗ trợ producer xác định thể loại của bản demo.
- Gợi ý các đặc trưng âm học cần điều chỉnh để bài hát phù hợp với thể loại mục tiêu.

4. Phân tích xu hướng âm nhạc (Music Trend Analysis):

- Nghiên cứu sự thay đổi của các thể loại nhạc theo thời gian.
- Phân tích ảnh hưởng của các đặc trưng âm học đến độ phổ biến của bài hát.

6 Kết luận và hướng phát triển

6.1 Kết luận

Nghiên cứu này đã thành công trong việc giải quyết bài toán phân loại thể loại âm nhạc (Music Genre Classification) sử dụng các thuật toán Machine Learning trên bộ dữ liệu đặc trưng âm học từ Kaggle. Các kết luận chính bao gồm:

1. Hiệu quả của Feature Engineering: Thí nghiệm so sánh EXP1 (Baseline) vs EXP2 (Advanced) đã chứng minh rằng **Feature Engineering nâng cao cải thiện hiệu suất từ 1.91% đến 3.45%** trên tất cả các mô hình. Đặc biệt, các kỹ thuật như:

- **Cyclical Encoding** cho biến tuần hoàn (`key`) giúp mô hình học được cấu trúc âm nhạc.
- **Interaction Features** (energy_acoustic_ratio, energy_loudness_product) nắm bắt mối quan hệ phi tuyến.
- **Target Encoding** cho Artist Name hiệu quả hơn One-Hot Encoding với biến phân loại cao chiều.

2. XGBoost đạt hiệu suất tốt nhất: Mô hình **XGBoost trên EXP2 đạt F1-Score 88.91%** (Validation) và **88.56%** (Kaggle Public Leaderboard), vượt trội hơn các mô hình truyền thống như Logistic Regression (76.89%) và SVM (80.12%). Điều này khẳng định sức mạnh của gradient boosting trong việc xử lý:

- Dữ liệu mất cân bằng nghiêm trọng (imbalanced data).
- Các mối quan hệ phi tuyến phức tạp giữa đặc trưng âm học và thể loại.
- Tương tác đa chiều giữa các đặc trưng.

3. Trade-off giữa Accuracy và Speed:

- **XGBoost**: Accuracy cao nhất (88.91%) nhưng training chậm (51.3s).
- **LightGBM**: Tốc độ nhanh gấp 3.2 lần (16.2s) với accuracy chỉ thấp hơn 0.57%, là lựa chọn tốt cho production.
- **Random Forest**: Cân bằng tốt giữa tốc độ (25.7s) và accuracy (85.23%), dễ tune hyperparameters.

4. Tầm quan trọng của các đặc trưng: Phân tích Feature Importance cho thấy:

1. `acousticness` (0.142) - Phân biệt nhạc mộc vs điện tử.
2. `energy_acoustic_ratio` (0.118) - Interaction feature quan trọng nhất.
3. `energy` (0.095) - Đặc trưng cơ bản nhưng có tầm quan trọng cao.

5. Generalization tốt: Gap giữa Validation và Public Score rất nhỏ (<0.7%), chứng tỏ mô hình không bị overfitting và có khả năng generalize tốt trên dữ liệu mới.

6.2 Đóng góp của nghiên cứu

Nghiên cứu đã đóng góp những kiến thức và kỹ thuật cụ thể cho lĩnh vực Music Information Retrieval:

1. **So sánh toàn diện 5 thuật toán Machine Learning** trên cùng một bộ dữ liệu âm nhạc, cung cấp cơ sở khoa học cho việc lựa chọn mô hình.
2. **Thiết kế thí nghiệm có đối chứng (Controlled Experiment)** với EXP1 vs EXP2 để kiểm chứng hiệu quả của Feature Engineering.
3. **Phát triển pipeline Feature Engineering** cho dữ liệu âm nhạc, bao gồm:
   - Cyclical Encoding cho biến tuần hoàn (key, time_signature).
   - Interaction features dựa trên phân tích tương quan.
   - Target Encoding cho biến phân loại cao chiều (Artist Name).
4. **Phân tích sâu về Feature Importance** để hiểu yếu tố nào quyết định thể loại âm nhạc.

6.3 Hướng phát triển tương lai

Nghiên cứu có thể được mở rộng theo các hướng sau:

1. Sử dụng Deep Learning trên Raw Audio:

- Sử dụng **Convolutional Neural Networks (CNN)** trên Mel-spectrograms để trích xuất đặc trưng tự động.
- Áp dụng **Transfer Learning** với pre-trained models như MusiCNN, VGGish, hoặc OpenL3.
- Kết hợp **Attention Mechanism** để model tập trung vào các phần quan trọng của bài hát (chorus, intro, drop).

2. Xử lý triệt để Imbalanced Data:

- Áp dụng **SMOTE (Synthetic Minority Over-sampling Technique)** để tạo mẫu tổng hợp cho lớp thiểu số.
- Sử dụng **Focal Loss** thay vì Cross-Entropy để model tập trung vào các mẫu khó phân loại.
- Thử nghiệm **Cost-Sensitive Learning** với ma trận chi phí (cost matrix) tùy chỉnh.

3. Tối ưu Hyperparameters với Bayesian Optimization:

- Sử dụng **Optuna** hoặc **Hyperopt** thay vì GridSearchCV để tìm kiếm không gian tham số rộng hơn.
- Áp dụng **AutoML** (Auto-sklearn, TPOT) để tự động hóa toàn bộ pipeline từ Feature Engineering đến Model Selection.

4. Ensemble Learning nâng cao:

- Kết hợp nhiều mô hình (Stacking Ensemble): XGBoost + LightGBM + CatBoost với meta-learner là Logistic Regression.
- Sử dụng **Voting Classifier** (soft voting) để tận dụng điểm mạnh của từng mô hình.

5. Phân loại đa nhãn (Multi-label Classification):

- Mở rộng từ single-label sang **multi-label** để một bài hát có thể thuộc nhiều thể loại (ví dụ: "Rock-Pop", "Jazz-Blues").
- Áp dụng **Label Powerset** hoặc **Classifier Chains** cho multi-label learning.

6. Phân tích cảm xúc âm nhạc (Music Emotion Recognition):

- Mở rộng bài toán từ phân loại thể loại sang phân loại cảm xúc (happy, sad, angry, relaxed) dựa trên mô hình Russell's Circumplex (Valence-Arousal).
- Tích hợp thông tin lời nhạc (lyrics) bằng NLP models (BERT, GPT) để cải thiện độ chính xác.

7. Triển khai Production System:

- Xây dựng **REST API** với FastAPI hoặc Flask để serving model.
- Containerize bằng **Docker** và deploy lên cloud (AWS SageMaker, Google Cloud AI Platform).
- Thiết lập **MLOps pipeline** với model monitoring, retraining tự động khi có dữ liệu mới.

Tài liệu

1. Tzanetakis, G., Cook, P. (2002). *Musical Genre Classification of Audio Signals*. IEEE Transactions on Speech and Audio Processing, 10(5), 293-302.
2. Chen, T., Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.
3. Ke, G., Meng, Q., Finley, T., et al. (2017). *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. Advances in Neural Information Processing Systems (NeurIPS), 3146-3154.
4. Pedregosa, F., et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825-2830.
5. McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. Proceedings of the 9th Python in Science Conference, 56-61.
6. Harris, C. R., et al. (2020). *Array Programming with NumPy*. Nature, 585, 357-362.
7. Hunter, J. D. (2007). *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering, 9(3), 90-95.
8. Waskom, M. (2021). *seaborn: Statistical Data Visualization*. Journal of Open Source Software, 6(60), 3021.
9. Costa, Y. M. G., Oliveira, L. S., Silla Jr, C. N. (2017). *An Evaluation of Convolutional Neural Networks for Music Classification Using Spectrograms*. Applied Soft Computing, 52, 28-38.
10. Choi, K., Fazekas, G., Sandler, M., Cho, K. (2017). *Convolutional Recurrent Neural Networks for Music Classification*. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2392-2396.
11. Spotify for Developers. (2023). *Web API Reference - Audio Features*. `https://developer.spotify.com/documentation/web-api/reference/get-audio-features`
12. Breiman, L. (2001). *Random Forests*. Machine Learning, 45(1), 5-32.
13. Cortes, C., Vapnik, V. (1995). *Support-Vector Networks*. Machine Learning, 20(3), 273-297.
14. Hosmer Jr, D. W., Lemeshow, S., Sturdivant, R. X. (2013). *Applied Logistic Regression* (3rd ed.). Wiley.
15. Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. (2002). *SMOTE: Synthetic Minority Over-sampling Technique*. Journal of Artificial Intelligence Research, 16, 321-357.