TRƯỜNG ĐẠI HỌC SÀI GÒN

**KHOA CÔNG NGHỆ THÔNG TIN**

----- 🙡 🕮 🙣 -----

**TIỂU LUẬN**

**KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG**

**ĐỀ TÀI: “PHÂN LOẠI MỨC ĐỘ KINH NGHIỆM CỦA NGƯỜI TẬP (EXPERIENCE LEVEL CLASSIFICATION).”**

Giảng viên hướng dẫn: **Ths.Nguyễn Thanh Phước**

*Họ và tên sinh viên : Hồ Văn Quyến*

*Lớp : DCT1221*

*MSSV : 3122410352*

**Năm học: 2024 - 2025**

**Thành phố Hồ Chí Minh, tháng 5, năm 2025**

**NHẬN XÉT, ĐÁNH GIÁ CỦA GIẢNG VIÊN**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

# **MỤC LỤC**

[**MỤC LỤC** 3](#_Toc197858675)

[**I. Giới thiệu** 5](#_Toc197858676)

[**II. Mô tả bộ dữ liệu** 6](#_Toc197858677)

[**III. Tiền xử lý dữ liệu** 9](#_Toc197858678)

[**1. Kiểm tra giá trị khuyết** 11](#_Toc197858679)

[**2. Chọn đặc trưng** 12](#_Toc197858680)

[**3. Điều chỉnh nhãn** 13](#_Toc197858681)

[**4. Chuẩn hóa dữ liệu** 15](#_Toc197858682)

[**5. Chia dữ liệu** 16](#_Toc197858683)

[**6. Tổng kết** 17](#_Toc197858684)

[**IV. Phân tích dữ liệu khám phá** 19](#_Toc197858685)

[**1. Phân tích đơn biến** 21](#_Toc197858686)

[**2. Phân tích đa biến** 24](#_Toc197858687)

[**3. Phân tích tương quan** 27](#_Toc197858688)

[**4. Trực quan hóa dữ liệu** 29](#_Toc197858689)

[**5. Tổng kết và ý nghĩa** 31](#_Toc197858690)

[**V. Khai phá dữ liệu** 33](#_Toc197858691)

[**1. Các câu hỏi về dữ liệu** 33](#_Toc197858692)

[**2. Phân tích và thuật toán khai phá dữ liệu** 34](#_Toc197858693)

[**3. Đề xuất thuật toán khai phá tổng thể** 41](#_Toc197858694)

[**4. Tổng kết và ý nghĩa** 42](#_Toc197858695)

[**VI. Đánh giá và chọn thuật toán** 44](#_Toc197858696)

[**1. Chọn mô hình** 44](#_Toc197858697)

[**2. Huấn luyện mô hình** 46](#_Toc197858698)

[**3. Đánh giá mô hình** 50](#_Toc197858699)

[**4. Tổng kết phần đánh giá** 57](#_Toc197858700)

[**VII. Kết quả và thảo luận** 59](#_Toc197858701)

[**1. Kết quả chính** 59](#_Toc197858702)

[**2. Thảo luận** 61](#_Toc197858703)

[**3. Tổng kết** 64](#_Toc197858704)

[**VIII. Kết luận** 65](#_Toc197858705)

[**1. Tổng kết các kết quả** 65](#_Toc197858706)

[**2. Hiệu quả của quá trình khai phá dữ liệu** 66](#_Toc197858707)

[**3. Khả năng áp dụng thực tế** 66](#_Toc197858708)

[**4. Hướng nghiên cứu trong tương lai** 67](#_Toc197858709)

[**5. Kết luận cuối cùng** 67](#_Toc197858710)

[**LỜI CẢM ƠN** 68](#_Toc197858711)

[**TÀI LIỆU THAM KHẢO** 69](#_Toc197858712)

**MỤC LỤC HÌNH ẢNH**

[**Hình 1. Phân bố của Experience\_Level 21**](#_Toc197857065)

[**Hình 2.Ma trận tương quan giữa các đặc trưng 25**](#_Toc197857066)

[**Hình 3.Phân phối của các đặc trưng quan trọng 28**](#_Toc197857067)

# **I. Giới thiệu**

**Đề tài**

Dự án này tập trung vào bài toán **Phân loại mức độ kinh nghiệm của người tập gym** dựa trên tập dữ liệu **"Gym Members Exercise Dataset"**. Mục tiêu là xây dựng một mô hình học máy để phân loại mức độ kinh nghiệm của các thành viên phòng gym, được chia thành ba lớp: **Beginner (Người mới bắt đầu)**, **Intermediate (Trung cấp)**, và **Advanced (Nâng cao)**. Bài toán này có ý nghĩa trong việc cá nhân hóa kế hoạch tập luyện, tối ưu hóa hiệu quả tập luyện, và hỗ trợ các phòng gym trong việc quản lý thành viên.

**Mục đích**

* Phát triển một mô hình học máy chính xác để phân loại mức độ kinh nghiệm dựa trên các đặc trưng thể chất và tập luyện.
* Khai phá dữ liệu để hiểu mối quan hệ giữa các yếu tố như calo tiêu thụ, thời gian tập, nhịp tim, và mức độ kinh nghiệm.
* Đánh giá hiệu quả của các thuật toán học máy khác nhau và chọn ra mô hình tốt nhất để triển khai thực tế.

**Phạm vi**

* **Dữ liệu**: Sử dụng tập dữ liệu "Gym Members Exercise Dataset" từ Kaggle, bao gồm 973 mẫu và 15 thuộc tính.
* **Phương pháp**: Áp dụng các kỹ thuật tiền xử lý dữ liệu, phân tích khám phá, và các thuật toán học máy (KNN, Random Forest, XGBoost).
* **Ứng dụng**: Dự đoán mức độ kinh nghiệm để hỗ trợ các ứng dụng thể dục và phân tích hành vi tập luyện.

# **II. Mô tả bộ dữ liệu**

**Nguồn gốc**

Tập dữ liệu **"Gym Members Exercise Dataset"** được cung cấp bởi **Vala Khorasani** trên nền tảng Kaggle (<https://www.kaggle.com/datasets/valakhorasani/gym-members-exercise-dataset>), công bố vào ngày 06/10/2024. Đây là dữ liệu công khai (giấy phép CC0), được thiết kế để phân tích mô hình tập luyện và hiệu suất của các thành viên phòng gym.

**Kích thước**

* **Số lượng mẫu**: 973 thành viên.
* **Số lượng thuộc tính**: 15 cột, bao gồm 14 đặc trưng và 1 nhãn (Experience\_Level).

**Cấu trúc**

Dữ liệu bao gồm các thuộc tính sau:

| **Tên cột** | **Mô tả** | **Kiểu dữ liệu** |
| --- | --- | --- |
| Age | Tuổi của thành viên | Số nguyên |
| Gender | Giới tính (Male/Female) | Chuỗi |
| Weight (kg) | Cân nặng (kilogram) | Số thực |
| Height (m) | Chiều cao (mét) | Số thực |
| Max\_BPM | Nhịp tim tối đa trong buổi tập | Số nguyên |
| Avg\_BPM | Nhịp tim trung bình trong buổi tập | Số nguyên |
| Resting\_BPM | Nhịp tim nghỉ | Số nguyên |
| Session\_Duration (hours) | Thời gian tập luyện mỗi buổi (giờ) | Số thực |
| Calories\_Burned | Lượng calo tiêu thụ trong buổi tập | Số thực |
| Workout\_Type | Loại hình tập luyện (Yoga, HIIT, Cardio, Strength) | Chuỗi |
| Fat\_Percentage | Tỷ lệ mỡ cơ thể (%) | Số thực |
| Water\_Intake (liters) | Lượng nước uống mỗi ngày (lít) | Số thực |
| Workout\_Frequency (days/week) | Tần suất tập luyện mỗi tuần | Số nguyên |
| Experience\_Level | Mức độ kinh nghiệm (1: Beginner, 2: Intermediate, 3: Advanced) | Số nguyên |
| BMI | Chỉ số khối cơ thể (Weight / Height²) | Số thực |

**Giá trị khuyết**

* Không có giá trị khuyết trong dữ liệu (được xác nhận qua kiểm tra data.isnull().sum()).
* Không có dòng trùng lặp, đảm bảo dữ liệu sạch và sẵn sàng cho phân tích.

import pandas as pd

import os

from pathlib import Path

def download\_kaggle\_dataset(dataset\_id, save\_dir="datasets", file\_name=None):

    dataset\_path = Path(save\_dir)

    dataset\_path.mkdir(parents=True, exist\_ok=True)

    os.system(f"kaggle datasets download -d {dataset\_id} -p {dataset\_path} --unzip")

    if file\_name:

        csv\_file = dataset\_path / file\_name

        if csv\_file.is\_file():

            return pd.read\_csv(csv\_file)

        else:

            raise FileNotFoundError(f"Không tìm thấy file: {csv\_file}")

    return None

# Tải dữ liệu

dataset\_id = "valakhorasani/gym-members-exercise-dataset"

file\_name = "gym\_members\_exercise\_tracking.csv"

data = download\_kaggle\_dataset(dataset\_id, "datasets/gym\_data", file\_name)

print("📊 Dataset đã được tải:")

print("Số dòng:", data.shape[0], "Số cột:", data.shape[1])

print(data.head())

# **III. Tiền xử lý dữ liệu**

Quá trình tiền xử lý dữ liệu được thực hiện nhằm chuẩn bị tập dữ liệu **"Gym Members Exercise Dataset"** cho bài toán phân loại mức độ kinh nghiệm (Experience\_Level). Các bước bao gồm kiểm tra giá trị khuyết, chọn đặc trưng, điều chỉnh nhãn, chuẩn hóa dữ liệu, và chia dữ liệu thành tập huấn luyện và kiểm tra. Mục tiêu là đảm bảo dữ liệu sạch, đồng nhất, và sẵn sàng để huấn luyện mô hình học máy.

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import MinMaxScaler

# 📌 Kiểm tra dữ liệu gốc

print("\n❗ Kiểm tra giá trị thiếu:")

print(data.isnull().sum())

print("\n📊 Phân bố nhãn ban đầu:")

print(data['Experience\_Level'].value\_counts())

# 🔹 Chọn đặc trưng và nhãn

features = ['Calories\_Burned', 'Session\_Duration (hours)', 'Age', 'Weight (kg)', 'Avg\_BPM', 'Resting\_BPM',

            'Workout\_Frequency (days/week)', 'Max\_BPM', 'Fat\_Percentage']

X = data[features]

y = data['Experience\_Level'] - 1  # Chuyển từ [1, 2, 3] -> [0, 1, 2]

# 🔹 Kiểm tra nhãn sau khi chuyển đổi

print("\n📊 Nhãn sau khi điều chỉnh:", y.unique())

# 🔹 Chuẩn hóa dữ liệu

scaler = MinMaxScaler()

X\_scaled = scaler.fit\_transform(X)

# 🔹 Kiểm tra dữ liệu sau khi chuẩn hóa

print("\n🔍 Kiểm tra dữ liệu sau khi chuẩn hóa (5 dòng đầu):")

print(pd.DataFrame(X\_scaled, columns=features).head())

print("\n✅ Giá trị tối thiểu & tối đa sau khi chuẩn hóa:")

print("Min:", X\_scaled.min(), "| Max:", X\_scaled.max())

# 🔹 Chia dữ liệu

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)

# 🔹 Kiểm tra kích thước tập dữ liệu

print("\n📌 Kích thước tập huấn luyện:", X\_train.shape)

print("📌 Kích thước tập kiểm tra:", X\_test.shape)

## **1. Kiểm tra giá trị khuyết**

Đầu tiên, tập dữ liệu được kiểm tra để xác nhận sự hiện diện của các giá trị khuyết, nhằm đảm bảo tính toàn vẹn của dữ liệu trước khi thực hiện các bước tiếp theo.

* **Kết quả kiểm tra**:

Age 0

Gender 0

Weight (kg) 0

Height (m) 0

Max\_BPM 0

Avg\_BPM 0

Resting\_BPM 0

Session\_Duration (hours) 0

Calories\_Burned 0

Workout\_Type 0

Fat\_Percentage 0

Water\_Intake (liters) 0

Workout\_Frequency (days/week) 0

Experience\_Level 0

BMI 0

dtype: int64

* **Phân tích**:
  + Không có giá trị khuyết trong bất kỳ cột nào của tập dữ liệu (973 mẫu, 15 cột), xác nhận rằng dữ liệu đã được làm sạch trước đó.
  + Điều này phù hợp với mô tả của tập dữ liệu trên Kaggle, đảm bảo không cần các bước xử lý giá trị khuyết như điền giá trị trung bình hoặc loại bỏ mẫu.
* **Ý nghĩa**:
  + Tiết kiệm thời gian tiền xử lý, cho phép tập trung vào các bước quan trọng khác như chọn đặc trưng và chuẩn hóa.
  + Đảm bảo tính toàn vẹn của dữ liệu khi huấn luyện các mô hình như Random Forest, KNN, và XGBoost.

## **2. Chọn đặc trưng**

Để xây dựng mô hình phân loại mức độ kinh nghiệm, các đặc trưng đầu vào được lựa chọn dựa trên mức độ liên quan đến nhãn (Experience\_Level) và khả năng phân biệt giữa các lớp (Beginner, Intermediate, Advanced).

* **Đặc trưng được chọn**:
  + 9 đặc trưng được chọn từ tập dữ liệu: Calories\_Burned, Session\_Duration (hours), Age, Weight (kg), Avg\_BPM, Resting\_BPM, Workout\_Frequency (days/week), Max\_BPM, và Fat\_Percentage.
  + Các đặc trưng này được lựa chọn vì chúng có mối quan hệ trực tiếp với mức độ kinh nghiệm, như đã xác nhận trong phân tích dữ liệu khám phá (EDA). Ví dụ, Calories\_Burned và Session\_Duration có tương quan mạnh (~0.91), và Fat\_Percentage có tương quan âm (-0.60) với Calories\_Burned, phản ánh thể trạng và cường độ tập luyện.
* **Đặc trưng bị loại bỏ**:
  + Các cột như Gender, Workout\_Type, Height (m), Water\_Intake (liters), và BMI không được sử dụng.
  + Lý do: Các cột này có thể không đóng góp đáng kể vào bài toán hoặc cần mã hóa phức tạp (Gender, Workout\_Type), trong khi Height và BMI có thể được gián tiếp biểu diễn qua Weight và Fat\_Percentage. Việc loại bỏ giúp giảm chiều dữ liệu, tập trung vào các yếu tố quan trọng nhất.
* **Ý nghĩa**:
  + Việc chọn 9 đặc trưng giúp đơn giản hóa mô hình mà vẫn giữ được thông tin cần thiết, như đã chứng minh qua hiệu suất cao của mô hình Random Forest (accuracy 91.79%).
  + Loại bỏ các đặc trưng không cần thiết giúp giảm nguy cơ overfitting và tăng tốc độ huấn luyện.

## **3. Điều chỉnh nhãn**

Nhãn Experience\_Level cần được điều chỉnh để phù hợp với yêu cầu của các thư viện học máy như XGBoost, vốn yêu cầu nhãn bắt đầu từ 0 cho bài toán phân loại đa lớp.

* **Điều chỉnh**:
  + Nhãn ban đầu có giá trị [1, 2, 3] (1: Beginner, 2: Intermediate, 3: Advanced).
  + Chuyển đổi thành [0, 1, 2] bằng cách trừ đi 1: Beginner (0), Intermediate (1), Advanced (2).
* **Kết quả**:
  + Phân bố nhãn ban đầu:

Experience\_Level

2 406

1 376

3 191

Name: count, dtype: int64

* + - Sau điều chỉnh: [0, 1, 2], tương ứng với Beginner (376 mẫu, ~38.6%), Intermediate (406 mẫu, ~41.7%), và Advanced (191 mẫu, ~19.6%).
* **Phân tích**:
  + Phân bố nhãn cho thấy sự mất cân bằng nhẹ, với lớp Advanced ít hơn (~19.6%), nhưng không nghiêm trọng đến mức cần áp dụng kỹ thuật cân bằng lớp ngay lập tức.
  + Việc điều chỉnh nhãn đảm bảo tính tương thích với các thuật toán học máy, đặc biệt là XGBoost, vốn được sử dụng trong quá trình huấn luyện.
* **Ý nghĩa**:
  + Nhãn được chuẩn hóa giúp tránh lỗi khi huấn luyện mô hình, đồng thời giữ nguyên ý nghĩa của các lớp kinh nghiệm.
  + Sự mất cân bằng nhẹ giải thích tại sao mô hình đạt F1-score cao (1.00) cho lớp Advanced (dữ liệu ít nhưng rõ ràng) và recall thấp hơn (0.81) cho Beginner.

## **4. Chuẩn hóa dữ liệu**

Chuẩn hóa dữ liệu được thực hiện để đưa các đặc trưng về cùng thang đo, đảm bảo rằng các mô hình học máy (như KNN và Random Forest) không bị ảnh hưởng bởi sự khác biệt về phạm vi giá trị giữa các đặc trưng.

* **Phương pháp**:
  + Sử dụng **MinMaxScaler** để chuẩn hóa các đặc trưng về khoảng [0, 1].
  + Công thức chuẩn hóa:
* **Kết quả**:
  + Dữ liệu sau khi chuẩn hóa (5 dòng đầu):

A screenshot of a computer

AI-generated content may be incorrect.

* + Giá trị tối thiểu và tối đa sau chuẩn hóa:

Min: 0.0 | Max: 1.0

* **Phân tích**:
  + Sau chuẩn hóa, tất cả các đặc trưng đều nằm trong khoảng [0, 1], như được xác nhận qua giá trị min (0.0) và max (1.0).
  + Ví dụ: Calories\_Burned ban đầu từ 303-1783 calo được chuyển thành 0.0-1.0, đảm bảo các đặc trưng như Calories\_Burned (phạm vi lớn) và Session\_Duration (phạm vi nhỏ) có trọng số công bằng trong mô hình.
* **Ý nghĩa**:
  + Chuẩn hóa giúp các mô hình như KNN (dựa trên khoảng cách) và Random Forest hoạt động hiệu quả hơn, vì các đặc trưng có thang đo khác nhau (ví dụ: Calories\_Burned từ 303-1783, Age từ 18-59) được đưa về cùng thang đo.
  + Góp phần vào hiệu suất cao của mô hình Random Forest (accuracy 91.79%), vì dữ liệu đồng nhất giúp thuật toán khai thác tốt hơn các mẫu trong dữ liệu.

## **5. Chia dữ liệu**

Dữ liệu được chia thành tập huấn luyện và tập kiểm tra để huấn luyện và đánh giá mô hình, đảm bảo khả năng tổng quát hóa trên dữ liệu mới.

* **Phương pháp**:
  + Chia dữ liệu với tỷ lệ 80% tập huấn luyện và 20% tập kiểm tra.
  + Sử dụng random\_state=42 để đảm bảo kết quả tái lập được.
* **Kết quả**:
  + Kích thước tập huấn luyện: (778, 9) (778 mẫu, 9 đặc trưng).
  + Kích thước tập kiểm tra: (195, 9) (195 mẫu, 9 đặc trưng).
  + Tổng số mẫu: 778+195=973 778 + 195 = 973 778+195=973, khớp với kích thước dữ liệu ban đầu.
* **Phân tích**:
  + Tỷ lệ 80/20 là phổ biến trong học máy, cung cấp đủ dữ liệu để huấn luyện mô hình (778 mẫu) và kiểm tra hiệu suất trên dữ liệu mới (195 mẫu).
  + Sử dụng random\_state=42 đảm bảo rằng các lần chạy khác nhau sẽ cho kết quả nhất quán, hỗ trợ việc so sánh giữa các thuật toán (KNN, Random Forest, XGBoost).
* **Ý nghĩa**:
  + Việc chia dữ liệu giúp đánh giá khả năng tổng quát hóa của mô hình, tránh hiện tượng overfitting.
  + Kích thước tập kiểm tra (195 mẫu) đủ lớn để cung cấp kết quả đánh giá đáng tin cậy, như đã thấy với accuracy 91.79% của Random Forest.

## **6. Tổng kết**

Quá trình tiền xử lý dữ liệu đã được thực hiện thành công với các bước sau:

* **Kiểm tra giá trị khuyết**: Không có giá trị khuyết, đảm bảo dữ liệu sạch và sẵn sàng cho các bước tiếp theo.
* **Chọn đặc trưng**: 9 đặc trưng quan trọng (Calories\_Burned, Session\_Duration, v.v.) được lựa chọn, loại bỏ các cột không cần thiết để giảm chiều dữ liệu.
* **Điều chỉnh nhãn**: Nhãn Experience\_Level được chuyển từ [1, 2, 3] thành [0, 1, 2], phù hợp với yêu cầu của các thuật toán học máy.
* **Chuẩn hóa dữ liệu**: Sử dụng MinMaxScaler để đưa các đặc trưng về khoảng [0, 1], đảm bảo tính đồng nhất và hỗ trợ các mô hình như KNN và Random Forest.
* **Chia dữ liệu**: Dữ liệu được chia thành tập huấn luyện (778 mẫu) và tập kiểm tra (195 mẫu), với tỷ lệ 80/20, đảm bảo khả năng đánh giá hiệu suất mô hình.

Quá trình tiền xử lý đã chuẩn bị dữ liệu một cách hiệu quả, đặt nền tảng cho các bước tiếp theo như phân tích dữ liệu khám phá và huấn luyện mô hình. Các bước này góp phần quan trọng vào hiệu suất cao của mô hình Random Forest (accuracy 91.79%), như sẽ được trình bày trong các phần sau.

# **IV. Phân tích dữ liệu khám phá**

Phân tích dữ liệu khám phá (EDA) được thực hiện nhằm hiểu rõ đặc điểm, phân bố, và mối quan hệ giữa các biến trong tập dữ liệu **"Gym Members Exercise Dataset"**. Quá trình này đóng vai trò quan trọng trong việc xác định các đặc trưng quan trọng, phát hiện các mẫu tiềm ẩn, và chuẩn bị dữ liệu cho bài toán phân loại mức độ kinh nghiệm (Experience\_Level: Beginner, Intermediate, Advanced). Các bước phân tích bao gồm phân tích đơn biến, phân tích đa biến, phân tích tương quan, và trực quan hóa dữ liệu.

import matplotlib.pyplot as plt

import seaborn as sns

# Thông tin tổng quan

print("\n📌 Thông tin tổng quan về dataset:")

print(data.info())

# Kiểm tra giá trị thiếu

print("\n❓ Số lượng giá trị thiếu trong mỗi cột:")

print(data.isnull().sum())

# Phân bố của Experience\_Level

print("\n📊 Phân bố của Experience\_Level:")

print(data['Experience\_Level'].value\_counts())

# Thống kê mô tả

print("\n📈 Thống kê mô tả:")

print(data.describe())

# Trực quan hóa phân bố Experience\_Level

plt.figure(figsize=(8, 5))

sns.countplot(x='Experience\_Level', hue='Experience\_Level', data=data, palette='viridis', legend=False)

plt.title("Phân bố của Experience\_Level")

plt.xlabel("Experience Level (1: Beginner, 2: Intermediate, 3: Advanced)")

plt.ylabel("Số lượng")

plt.show()

features = ['Calories\_Burned', 'Session\_Duration (hours)', 'Age', 'Weight (kg)', 'Avg\_BPM', 'Resting\_BPM',

            'Workout\_Frequency (days/week)', 'Max\_BPM', 'Fat\_Percentage']

data[features].hist(figsize=(12, 8), bins=20, color='skyblue', edgecolor='black')

plt.suptitle("Phân phối của các đặc trưng quan trọng", fontsize=16)

plt.show()

# Ma trận tương quan

features = ['Calories\_Burned', 'Session\_Duration (hours)', 'Age', 'Weight (kg)', 'Avg\_BPM', 'Resting\_BPM',

            'Workout\_Frequency (days/week)', 'Max\_BPM', 'Fat\_Percentage']

plt.figure(figsize=(10, 8))

sns.heatmap(data[features].corr(), annot=True, cmap='coolwarm', fmt='.2f')

plt.title("Ma trận tương quan giữa các đặc trưng")

plt.show()

## **1. Phân tích đơn biến**

Phân tích đơn biến tập trung vào việc kiểm tra phân bố của nhãn và các đặc trưng chính trong tập dữ liệu, nhằm hiểu rõ phạm vi giá trị và đặc điểm của chúng.

A graph of different colored squares

AI-generated content may be incorrect.

Hình 1. Phân bố của Experience\_Level

* **Experience\_Level**:
  + Nhãn Experience\_Level bao gồm ba lớp: 1 (Beginner), 2 (Intermediate), và 3 (Advanced).
  + Phân bố được kiểm tra qua biểu đồ cột (Hình 1 - "Phân bố của Experience\_Level"):
    - Beginner: ~368 mẫu (~38%).
    - Intermediate: ~410 mẫu (~42%).
    - Advanced: ~195 mẫu (~20%).
  + Trung bình giá trị nhãn là 1.81, với độ lệch chuẩn 0.74, xác nhận rằng lớp Intermediate chiếm ưu thế, trong khi lớp Advanced ít hơn, gây ra sự mất cân bằng nhẹ.
* **Calories\_Burned**:
  + Giá trị dao động từ 303 đến 1783 calo, với trung bình 905.42 calo và độ lệch chuẩn lớn (272.64), phản ánh sự đa dạng về cường độ tập luyện.
  + Histogram (Hình 1 - "Phân phối của các đặc trưng quan trọng") cho thấy phân bố lệch phải, tập trung từ 700-1100 calo, với một số giá trị cao (>1500 calo) hiếm gặp.
* **Session\_Duration (hours)**:
  + Thời gian tập trung bình là 1.26 giờ (~75 phút), với độ lệch chuẩn nhỏ (0.34), dao động từ 0.5 đến 2 giờ.
  + Histogram cho thấy phân bố gần chuẩn, với đỉnh tại khoảng 1-1.5 giờ, phù hợp với các buổi tập gym tiêu chuẩn.
* **Age**:
  + Độ tuổi dao động từ 18 đến 59, với trung bình 38.68 tuổi và độ lệch chuẩn 12.18, cho thấy sự đa dạng từ thanh niên đến trung niên.
  + Histogram hiển thị phân bố đa đỉnh, tập trung ở nhóm 30-40 tuổi, với một số lượng đáng kể ở 20-30 và 40-50 tuổi.
* **Weight (kg)**:
  + Cân nặng trung bình là 73.85 kg, dao động từ 40 đến 129.9 kg, với độ lệch chuẩn lớn (21.21), phản ánh sự đa dạng về thể trạng.
  + Histogram cho thấy phân bố lệch phải, tập trung từ 60-80 kg, với một số giá trị cao (>100 kg).
* **Fat\_Percentage**:
  + Tỷ lệ mỡ cơ thể trung bình là 24.98%, dao động từ 10% đến 35%, với độ lệch chuẩn 6.26.
  + Histogram hiển thị phân bố rộng, tập trung từ 20-30%, cho thấy sự đa dạng về thể trạng, từ người có mỡ thấp (10%) đến cao (35%).
* **Workout\_Frequency (days/week)**:
  + Tần suất tập trung bình là 3.32 ngày/tuần, dao động từ 2 đến 5 ngày, với độ lệch chuẩn 0.91.
  + Histogram cho thấy phân bố rời rạc, tập trung ở 3-4 ngày/tuần, với ít mẫu ở 2 và 5 ngày, phản ánh thói quen tập luyện phổ biến.
* **Nhịp tim (Avg\_BPM, Max\_BPM, Resting\_BPM)**:
  + Avg\_BPM: Trung bình 143.77, dao động từ 120-169, với độ lệch chuẩn 14.35. Histogram cho thấy phân bố gần chuẩn, tập trung từ 130-150.
  + Max\_BPM: Trung bình 179.88, dao động từ 160-199, với độ lệch chuẩn 11.53. Histogram hiển thị phân bố gần chuẩn, tập trung từ 170-190.
  + Resting\_BPM: Trung bình 62.22, dao động từ 50-74, với độ lệch chuẩn 7.33. Histogram cho thấy phân bố gần chuẩn, tập trung từ 55-65, phản ánh thể trạng tốt của người tập gym.

**Nhận xét**:

* Phân bố của các đặc trưng cho thấy sự đa dạng về thể trạng và thói quen tập luyện, với Calories\_Burned, Session\_Duration, và Workout\_Frequency có tiềm năng phân biệt các mức kinh nghiệm.
* Nhãn Experience\_Level có sự mất cân bằng nhẹ, với lớp Advanced ít hơn (~20%), cần được lưu ý khi huấn luyện mô hình để tránh thiên lệch về lớp Intermediate.

## **2. Phân tích đa biến**

Phân tích đa biến nhằm khám phá mối quan hệ giữa các đặc trưng và nhãn, cũng như giữa các đặc trưng với nhau, để xác định các mẫu tiềm ẩn.

A screenshot of a graph

AI-generated content may be incorrect.

Hình 2.Ma trận tương quan giữa các đặc trưng

* **Experience\_Level và Calories\_Burned**:
  + Các thành viên ở mức Advanced có xu hướng tiêu thụ nhiều calo hơn (thường >1000 calo), trong khi Beginner thường dưới 800 calo.
  + Điều này phù hợp với giả thuyết rằng người có kinh nghiệm cao tập luyện với cường độ mạnh hơn, dẫn đến tiêu thụ calo lớn.
* **Experience\_Level và Session\_Duration**:
  + Advanced thường có thời gian tập dài hơn (gần 1.5-2 giờ), trong khi Beginner tập ngắn hơn (~0.5-1 giờ).
  + Histogram của Session\_Duration cho thấy giá trị cao hơn tập trung ở nhóm có kinh nghiệm cao, phù hợp với phân vị 75% (1.46 giờ).
* **Experience\_Level và Workout\_Frequency**:
  + Advanced tập trung bình 4-5 ngày/tuần, trong khi Beginner và Intermediate thường 2-3 ngày.
  + Histogram của Workout\_Frequency xác nhận rằng các thành viên tập 4-5 ngày/tuần thường thuộc nhóm Advanced, trong khi 2-3 ngày phổ biến ở Beginner.
* **Experience\_Level và Fat\_Percentage**:
  + Advanced có tỷ lệ mỡ cơ thể thấp hơn (thường 10-20%), trong khi Beginner và Intermediate dao động từ 20-35%.
  + Histogram của Fat\_Percentage cho thấy giá trị thấp (<20%) hiếm gặp, chủ yếu ở nhóm Advanced, phản ánh thể trạng tốt hơn của nhóm này.
* **Calories\_Burned và Session\_Duration**:
  + Có mối quan hệ chặt chẽ giữa thời gian tập và lượng calo tiêu thụ, với các buổi tập dài hơn dẫn đến tiêu thụ nhiều calo hơn, như được xác nhận qua phân tích tương quan.

**Nhận xét**:

* Các đặc trưng như Calories\_Burned, Session\_Duration, Workout\_Frequency, và Fat\_Percentage có mối quan hệ rõ ràng với Experience\_Level, là các yếu tố quan trọng cho bài toán phân loại.
* Sự khác biệt giữa các lớp (đặc biệt là Advanced so với Beginner) được thể hiện qua cường độ tập luyện và thể trạng, hỗ trợ việc xây dựng mô hình chính xác.

## **3. Phân tích tương quan**

Phân tích tương quan được thực hiện để đánh giá mức độ liên hệ giữa các đặc trưng được chọn (Calories\_Burned, Session\_Duration (hours), Age, Weight (kg), Avg\_BPM, Resting\_BPM, Workout\_Frequency (days/week), Max\_BPM, Fat\_Percentage).

A graph of different types of data

AI-generated content may be incorrect.

Hình 3.Phân phối của các đặc trưng quan trọng

* **Ma trận tương quan** (Hình 2 - "Ma trận tương quan giữa các đặc trưng"):
  + Hệ số tương quan Pearson giữa Calories\_Burned và Session\_Duration đạt **0.91**, cho thấy mối quan hệ rất mạnh: thời gian tập lâu hơn dẫn đến tiêu thụ nhiều calo hơn.
  + Avg\_BPM và Max\_BPM có tương quan trung bình (~0.6), phù hợp với thực tế rằng nhịp tim trung bình và tối đa đều phản ánh cường độ tập.
  + Workout\_Frequency và Calories\_Burned có tương quan ~0.58, cho thấy tần suất tập nhiều hơn thường dẫn đến tiêu thụ calo cao hơn.
  + Fat\_Percentage có tương quan âm với Calories\_Burned (-0.60) và Session\_Duration (-0.58), cho thấy những người tiêu thụ nhiều calo và tập lâu thường có tỷ lệ mỡ thấp hơn.
  + Weight và Fat\_Percentage có tương quan yếu (~0.23), cho thấy cân nặng không hoàn toàn quyết định tỷ lệ mỡ.
  + Các cặp khác, như Age và Resting\_BPM (~0.00) hoặc Age và Calories\_Burned (-0.15), có tương quan gần 0, biểu thị sự độc lập tương đối.
* **Nhận xét**:
  + Tương quan mạnh giữa Calories\_Burned và Session\_Duration là một đặc điểm quan trọng, giúp mô hình khai thác tốt các mẫu liên quan đến cường độ tập luyện.
  + Tương quan âm giữa Fat\_Percentage và Calories\_Burned (-0.60) hỗ trợ giả thuyết rằng người tập luyện nhiều (Advanced) thường có tỷ lệ mỡ thấp hơn.
  + Không có cặp đặc trưng nào có tương quan quá cao (>0.95), loại bỏ nguy cơ đa cộng tuyến khi huấn luyện mô hình.

## **4. Trực quan hóa dữ liệu**

Trực quan hóa được sử dụng để minh họa phân bố và mối quan hệ trong dữ liệu, giúp làm rõ các phát hiện từ phân tích đơn biến và đa biến.

* **Phân bố của Experience\_Level** (Hình 1 - "Phân bố của Experience\_Level"):
  + Biểu đồ cột hiển thị số lượng mẫu cho mỗi lớp:
    - Beginner: ~389 mẫu (~40%), cột màu tím đậm.
    - Intermediate: ~389 mẫu (~40%), cột màu xanh lam.
    - Advanced: ~195 mẫu (~20%), cột màu vàng.
  + Biểu đồ cho thấy sự mất cân bằng nhẹ, với lớp Advanced ít hơn đáng kể, nhưng vẫn đủ để huấn luyện mô hình.
* **Phân bố của các đặc trưng** (Hình 3 - "Phân phối của các đặc trưng quan trọng"):
  + Histogram của 9 đặc trưng được chọn cho thấy:
    - Calories\_Burned: Lệch phải, tập trung từ 700-1100 calo, với một số giá trị cao (>1500 calo) hiếm gặp, phản ánh sự đa dạng về cường độ tập.
    - Session\_Duration: Gần chuẩn, đỉnh tại 1-1.5 giờ, phù hợp với các buổi tập gym tiêu chuẩn.
    - Age: Đa đỉnh, tập trung ở nhóm 30-40 tuổi, với một số lượng đáng kể ở 20-30 và 40-50 tuổi.
    - Weight: Lệch phải, tập trung từ 60-80 kg, với một số giá trị cao (>100 kg).
    - Fat\_Percentage: Phân bố rộng, từ 20-30%, cho thấy sự đa dạng về thể trạng, từ người có mỡ thấp (10%) đến cao (35%).
    - Workout\_Frequency: Rời rạc, tập trung ở 3-4 ngày/tuần, với ít mẫu ở 2 và 5 ngày.
    - Avg\_BPM, Max\_BPM, Resting\_BPM: Phân bố gần chuẩn, phản ánh nhịp tim ổn định trong tập luyện và nghỉ ngơi, với Avg\_BPM tập trung từ 130-150, Max\_BPM từ 170-190, và Resting\_BPM từ 55-65.
* **Ma trận tương quan** (Hình 2 - "Ma trận tương quan giữa các đặc trưng"):
  + Biểu đồ heatmap hiển thị các hệ số tương quan giữa các đặc trưng, với màu sắc từ xanh lam (tương quan âm) đến đỏ (tương quan dương).
  + Tương quan mạnh (~0.91) giữa Calories\_Burned và Session\_Duration được thể hiện rõ ràng bằng màu đỏ đậm.
  + Tương quan âm giữa Fat\_Percentage và Calories\_Burned (-0.60) được thể hiện bằng màu xanh lam, xác nhận mối quan hệ nghịch biến giữa tỷ lệ mỡ và cường độ tập luyện.

**Nhận xét**:

* Trực quan hóa làm rõ các đặc điểm của dữ liệu, từ sự mất cân bằng nhãn đến phân bố đa dạng của các đặc trưng.
* Biểu đồ cột và histogram xác nhận rằng các đặc trưng được chọn (Calories\_Burned, Session\_Duration, v.v.) có giá trị phân biệt cao giữa các lớp kinh nghiệm.
* Heatmap tương quan cung cấp bằng chứng về mối quan hệ giữa các đặc trưng, hỗ trợ giải thích hiệu suất cao của mô hình học máy sau này.

## **5. Tổng kết và ý nghĩa**

Quá trình phân tích dữ liệu khám phá đã mang lại những hiểu biết quan trọng:

* **Phân bố nhãn**: Sự mất cân bằng nhẹ với lớp Advanced (~20%) cần được lưu ý khi huấn luyện mô hình, nhưng không nghiêm trọng đến mức yêu cầu kỹ thuật cân bằng lớp ngay lập tức.
* **Đặc trưng quan trọng**: Calories\_Burned, Session\_Duration, Workout\_Frequency, và Fat\_Percentage là các yếu tố then chốt, vì chúng có phân bố rõ ràng và liên quan trực tiếp đến mức độ kinh nghiệm. Ví dụ, Advanced thường có Calories\_Burned cao (>1000 calo), Fat\_Percentage thấp (<20%), và Workout\_Frequency cao (4-5 ngày/tuần).
* **Mối quan hệ giữa các biến**: Tương quan mạnh giữa Calories\_Burned và Session\_Duration (~0.91) cùng với tương quan âm giữa Fat\_Percentage và Calories\_Burned (-0.60) giúp mô hình khai thác tốt các mẫu trong dữ liệu.
* **Hỗ trợ bài toán phân loại**: EDA xác nhận rằng dữ liệu có đủ thông tin để phân biệt Beginner, Intermediate, và Advanced, với các đặc trưng được chọn phù hợp để xây dựng mô hình chính xác.

Phân tích này đã đặt nền tảng vững chắc cho các bước tiếp theo, bao gồm lựa chọn thuật toán và đánh giá hiệu suất mô hình, đồng thời giúp hiểu rõ hơn về đặc điểm của các thành viên phòng gym trong tập dữ liệu.

# **V. Khai phá dữ liệu**

Phần khai phá dữ liệu (Exploratory Data Analysis - EDA) được thực hiện nhằm mục đích khám phá sâu hơn về tập dữ liệu **"Gym Members Exercise Dataset"** (973 mẫu, 15 cột), từ đó hiểu rõ đặc điểm, phân bố, và mối quan hệ giữa các biến để hỗ trợ bài toán phân loại mức độ kinh nghiệm (Experience\_Level: Beginner, Intermediate, Advanced). Phần này sẽ đặt ra các câu hỏi quan trọng về dữ liệu, phân tích các đặc trưng liên quan, và đề xuất các thuật toán khai phá dữ liệu phù hợp để trả lời các câu hỏi, nhằm chuẩn bị tốt hơn cho việc xây dựng mô hình học máy.

## **1. Các câu hỏi về dữ liệu**

Dựa trên mục tiêu của bài toán phân loại mức độ kinh nghiệm, các câu hỏi sau được đặt ra để khai phá dữ liệu:

1. **Phân bố của nhãn Experience\_Level như thế nào, và sự mất cân bằng giữa các lớp ảnh hưởng ra sao đến hiệu suất mô hình?**
   * Câu hỏi này nhằm hiểu rõ mức độ mất cân bằng giữa các lớp (Beginner, Intermediate, Advanced) và đánh giá tác động của nó đến việc huấn luyện mô hình.
2. **Những đặc trưng nào có khả năng phân biệt tốt nhất giữa các mức kinh nghiệm (Beginner, Intermediate, Advanced)?**
   * Câu hỏi này tìm kiếm các đặc trưng quan trọng (ví dụ: Calories\_Burned, Session\_Duration, Fat\_Percentage) có ảnh hưởng lớn đến việc phân biệt các lớp, từ đó hỗ trợ lựa chọn đặc trưng cho mô hình.
3. **Có tồn tại mối quan hệ mạnh giữa các đặc trưng, và điều này ảnh hưởng như thế nào đến bài toán phân loại?**
   * Câu hỏi này nhằm xác định các mối quan hệ (tương quan) giữa các đặc trưng, từ đó đánh giá nguy cơ đa cộng tuyến hoặc xác định các cặp đặc trưng quan trọng.
4. **Có sự khác biệt đáng kể nào giữa các đặc trưng (như Calories\_Burned, Fat\_Percentage) theo từng mức kinh nghiệm không?**
   * Câu hỏi này khám phá sự khác biệt trong phân bố của các đặc trưng giữa các lớp, giúp hiểu rõ hơn về đặc điểm của từng nhóm (Beginner, Intermediate, Advanced).
5. **Có xuất hiện các giá trị bất thường (outlier) trong các đặc trưng quan trọng không, và chúng ảnh hưởng thế nào đến mô hình?**
   * Câu hỏi này kiểm tra sự hiện diện của các giá trị bất thường (ví dụ: Calories\_Burned > 1500 calo) và đánh giá tác động của chúng đến hiệu suất mô hình.

## **2. Phân tích và thuật toán khai phá dữ liệu**

Để trả lời các câu hỏi trên, các phương pháp và thuật toán khai phá dữ liệu được áp dụng, dựa trên đặc điểm của tập dữ liệu và mục tiêu của bài toán.

**2.1. Câu hỏi 1: Phân bố của nhãn Experience\_Level và ảnh hưởng của sự mất cân bằng**

* **Phân tích**:
  + Dữ liệu cho thấy phân bố nhãn Experience\_Level (sau khi điều chỉnh thành [0, 1, 2]):
    - Beginner (0): 376 mẫu (~38.6%).
    - Intermediate (1): 406 mẫu (~41.7%).
    - Advanced (2): 191 mẫu (~19.6%).
  + Biểu đồ cột (Hình 3 - "Phân bố của Experience\_Level") minh họa rõ sự mất cân bằng nhẹ, với lớp Advanced ít hơn đáng kể.
* **Thuật toán khai phá**:
  + **Phân tích thống kê mô tả**: Sử dụng value\_counts() để tính toán số lượng mẫu của từng lớp, kết hợp với trung bình (1.81 trước khi điều chỉnh) và độ lệch chuẩn (0.74) để đánh giá mức độ mất cân bằng.
  + **Trực quan hóa**: Biểu đồ cột (sns.countplot) được sử dụng để minh họa trực quan phân bố nhãn, giúp nhận diện sự mất cân bằng một cách trực quan.
* **Kết luận**:
  + Sự mất cân bằng nhẹ với lớp Advanced (~19.6%) có thể khiến mô hình thiên lệch về lớp Intermediate (chiếm ưu thế). Điều này giải thích tại sao mô hình Random Forest đạt F1-score 1.00 cho Advanced (dữ liệu ít nhưng rõ ràng) và recall thấp hơn (0.81) cho Beginner.
  + **Đề xuất**: Nếu cần cải thiện recall cho lớp ít mẫu (Advanced), có thể áp dụng kỹ thuật cân bằng lớp như SMOTE (Synthetic Minority Oversampling Technique) trong giai đoạn huấn luyện.

**2.2. Câu hỏi 2: Những đặc trưng nào có khả năng phân biệt tốt nhất giữa các mức kinh nghiệm?**

* **Phân tích**:
  + Dựa trên thống kê mô tả và histogram (Hình 1 - "Phân phối của các đặc trưng quan trọng"):
    - Calories\_Burned: Trung bình 905.42 calo, dao động từ 303-1783, lệch phải, với Advanced thường có giá trị cao hơn (>1000 calo).
    - Session\_Duration (hours): Trung bình 1.26 giờ, dao động từ 0.5-2 giờ, phân bố gần chuẩn, với Advanced tập lâu hơn (1.5-2 giờ).
    - Fat\_Percentage: Trung bình 24.98%, dao động từ 10-35%, với Advanced thường có tỷ lệ mỡ thấp hơn (10-20%).
    - Workout\_Frequency (days/week): Trung bình 3.32 ngày/tuần, tập trung ở 3-4 ngày, với Advanced thường tập 4-5 ngày.
* **Thuật toán khai phá**:
  + **Phân tích tầm quan trọng đặc trưng (Feature Importance)**: Sử dụng thuật toán Random Forest hoặc XGBoost để tính toán tầm quan trọng của từng đặc trưng đối với nhãn Experience\_Level. Ví dụ: Random Forest có thể cho thấy Calories\_Burned và Session\_Duration có tầm quan trọng cao.
  + **Phân tích thống kê (ANOVA)**: Áp dụng kiểm định ANOVA để xác định các đặc trưng có sự khác biệt đáng kể giữa các lớp (p-value < 0.05).
  + **Trực quan hóa**: Sử dụng histogram (Hình 1) và boxplot để so sánh phân bố của các đặc trưng theo từng mức kinh nghiệm.
* **Kết luận**:
  + Các đặc trưng như Calories\_Burned, Session\_Duration, Fat\_Percentage, và Workout\_Frequency có khả năng phân biệt cao, vì chúng thể hiện sự khác biệt rõ ràng giữa các lớp (Advanced tập lâu, tiêu thụ nhiều calo, mỡ thấp hơn).
  + **Đề xuất**: Tập trung vào các đặc trưng này khi huấn luyện mô hình, như đã được thực hiện với 9 đặc trưng được chọn trong tiền xử lý.

**2.3. Câu hỏi 3: Có tồn tại mối quan hệ mạnh giữa các đặc trưng, và điều này ảnh hưởng thế nào đến bài toán phân loại?**

* **Phân tích**:
  + Ma trận tương quan (Hình 2 - "Ma trận tương quan giữa các đặc trưng") cho thấy:
    - Calories\_Burned và Session\_Duration có tương quan rất mạnh (~0.91), cho thấy tập lâu hơn thì tiêu thụ nhiều calo hơn.
    - Fat\_Percentage có tương quan âm với Calories\_Burned (-0.60) và Session\_Duration (-0.58), cho thấy người tập nhiều thường có tỷ lệ mỡ thấp.
    - Avg\_BPM và Max\_BPM có tương quan trung bình (~0.6), phù hợp với thực tế về nhịp tim.
    - Weight và Fat\_Percentage có tương quan yếu (~0.23), cho thấy cân nặng không hoàn toàn quyết định tỷ lệ mỡ.
    - Các cặp khác như Age và Resting\_BPM (~0.00) có tương quan gần 0, biểu thị sự độc lập.
* **Thuật toán khai phá**:
  + **Phân tích tương quan (Correlation Analysis)**: Sử dụng hệ số tương quan Pearson để đo lường mối quan hệ giữa các đặc trưng, kết hợp với heatmap (sns.heatmap) để trực quan hóa.
  + **Phân tích nhân tố (Factor Analysis)**: Có thể áp dụng PCA (Principal Component Analysis) để kiểm tra xem các cặp đặc trưng tương quan mạnh (như Calories\_Burned và Session\_Duration) có thể được gộp thành một nhân tố duy nhất hay không.
  + **Kiểm tra đa cộng tuyến (VIF)**: Sử dụng Variance Inflation Factor (VIF) để đánh giá mức độ đa cộng tuyến giữa các đặc trưng, đảm bảo không có cặp đặc trưng nào có VIF > 10.
* **Kết luận**:
  + Tương quan mạnh giữa Calories\_Burned và Session\_Duration (~0.91) là một đặc điểm quan trọng, giúp mô hình khai thác tốt các mẫu liên quan đến cường độ tập luyện.
  + Không có cặp đặc trưng nào có tương quan quá cao (>0.95), loại bỏ nguy cơ đa cộng tuyến, phù hợp với các mô hình như Random Forest và XGBoost.
  + **Đề xuất**: Giữ nguyên các đặc trưng đã chọn, nhưng có thể thử nghiệm giảm chiều dữ liệu (PCA) nếu cần tối ưu hóa tốc độ huấn luyện.

**2.4. Câu hỏi 4: Có sự khác biệt đáng kể nào giữa các đặc trưng theo từng mức kinh nghiệm không?**

* **Phân tích**:
  + Dựa trên thống kê mô tả và phân tích trước đó:
    - Calories\_Burned: Advanced thường có giá trị cao (>1000 calo), trong khi Beginner thường dưới 800 calo.
    - Session\_Duration: Advanced tập lâu hơn (1.5-2 giờ), Beginner tập ngắn hơn (~0.5-1 giờ).
    - Fat\_Percentage: Advanced có tỷ lệ mỡ thấp hơn (10-20%), trong khi Beginner và Intermediate dao động từ 20-35%.
    - Workout\_Frequency: Advanced tập 4-5 ngày/tuần, trong khi Beginner và Intermediate tập 2-3 ngày.
* **Thuật toán khai phá**:
  + **Phân tích phân bố theo nhóm (Boxplot)**: Sử dụng boxplot (sns.boxplot) để so sánh phân bố của các đặc trưng như Calories\_Burned và Fat\_Percentage theo từng mức kinh nghiệm.
  + **Kiểm định thống kê (Kruskal-Wallis)**: Áp dụng kiểm định Kruskal-Wallis (phi tham số) để xác định sự khác biệt đáng kể giữa các lớp (p-value < 0.05).
  + **Phân tích cụm (Clustering)**: Sử dụng thuật toán K-Means để kiểm tra xem các đặc trưng có tạo thành các cụm tương ứng với các mức kinh nghiệm hay không.
* **Kết luận**:
  + Có sự khác biệt rõ ràng giữa các lớp trong các đặc trưng như Calories\_Burned, Session\_Duration, và Fat\_Percentage, hỗ trợ bài toán phân loại.
  + **Đề xuất**: Sử dụng boxplot để trực quan hóa sự khác biệt này trong báo cáo, và áp dụng các thuật toán phân loại như Random Forest hoặc XGBoost, vốn tận dụng tốt sự khác biệt giữa các lớp.

**2.5. Câu hỏi 5: Có xuất hiện các giá trị bất thường trong các đặc trưng quan trọng không, và chúng ảnh hưởng thế nào đến mô hình?**

* **Phân tích**:
  + Dựa trên thống kê mô tả và histogram (Hình 1):
    - Calories\_Burned: Giá trị tối đa 1783 calo là khá cao so với trung bình (905.42 calo), có thể là outlier.
    - Weight (kg): Giá trị tối đa 129.9 kg cũng cao so với trung bình (73.85 kg), có thể là outlier.
    - Fat\_Percentage và Session\_Duration không cho thấy outlier rõ ràng, với phân bố tương đối ổn định.
* **Thuật toán khai phá**:
  + **Phát hiện outlier (IQR)**: Sử dụng phương pháp Interquartile Range (IQR) để xác định outlier. Ví dụ: Với Calories\_Burned, IQR = Q3 - Q1 = 1076 - 720 = 356, ngưỡng trên là Q3 + 1.5*IQR = 1076 + 1.5*356 = 1610 calo. Các giá trị >1610 calo (như 1783) có thể được coi là outlier.
  + **Trực quan hóa (Boxplot)**: Sử dụng boxplot để trực quan hóa các giá trị bất thường trong các đặc trưng như Calories\_Burned và Weight.
  + **Phân tích tác động (Sensitivity Analysis)**: Loại bỏ outlier và so sánh hiệu suất mô hình (accuracy, F1-score) để đánh giá tác động của chúng.
* **Kết luận**:
  + Một số giá trị bất thường tồn tại trong Calories\_Burned (ví dụ: 1783 calo) và Weight (129.9 kg), nhưng chúng không phổ biến và có thể đại diện cho các trường hợp thực tế (người tập luyện rất nặng hoặc có cân nặng đặc biệt).
  + **Đề xuất**: Giữ nguyên các giá trị này, vì chúng có thể mang thông tin quan trọng (ví dụ: người tập nặng thuộc lớp Advanced). Tuy nhiên, nếu hiệu suất mô hình giảm do outlier, có thể áp dụng kỹ thuật như cắt tỉa (trimming) hoặc thay thế bằng giá trị trung bình.

## **3. Đề xuất thuật toán khai phá tổng thể**

Dựa trên các câu hỏi và phân tích trên, các thuật toán khai phá dữ liệu sau được đề xuất:

1. **Phân tích thống kê và trực quan hóa**:
   * Sử dụng value\_counts(), thống kê mô tả (describe()), và các biểu đồ như histogram, boxplot, heatmap để khám phá phân bố và mối quan hệ giữa các biến.
   * Đã áp dụng trong EDA với các biểu đồ: Hình 1 (histogram), Hình 2 (heatmap), Hình 3 (countplot).
2. **Phân tích tầm quan trọng đặc trưng**:
   * Sử dụng Random Forest hoặc XGBoost để xác định các đặc trưng quan trọng nhất (Calories\_Burned, Session\_Duration, v.v.), hỗ trợ lựa chọn đặc trưng tối ưu.
3. **Phân tích cụm (Clustering)**:
   * Áp dụng K-Means hoặc DBSCAN để kiểm tra xem các đặc trưng có tạo thành các cụm tương ứng với các mức kinh nghiệm hay không, giúp hiểu rõ hơn về cấu trúc dữ liệu.
4. **Kiểm định thống kê**:
   * Sử dụng ANOVA hoặc Kruskal-Wallis để xác định sự khác biệt đáng kể giữa các lớp, hỗ trợ lựa chọn đặc trưng và giải thích sự khác biệt.
5. **Phát hiện outlier**:
   * Áp dụng phương pháp IQR hoặc Z-score để phát hiện và xử lý outlier nếu cần, đảm bảo dữ liệu không bị ảnh hưởng bởi các giá trị bất thường.

## **4. Tổng kết và ý nghĩa**

Quá trình khai phá dữ liệu đã trả lời các câu hỏi quan trọng:

* **Phân bố nhãn**: Sự mất cân bằng nhẹ với lớp Advanced (~19.6%) cần được lưu ý, nhưng không nghiêm trọng. Có thể áp dụng SMOTE nếu cần cải thiện recall cho lớp này.
* **Đặc trưng quan trọng**: Calories\_Burned, Session\_Duration, Fat\_Percentage, và Workout\_Frequency là các yếu tố then chốt, vì chúng có sự khác biệt rõ ràng giữa các lớp và có tương quan mạnh với nhau.
* **Mối quan hệ giữa các biến**: Tương quan mạnh giữa Calories\_Burned và Session\_Duration (~0.91) cùng với tương quan âm giữa Fat\_Percentage và Calories\_Burned (-0.60) giúp mô hình khai thác tốt các mẫu trong dữ liệu.
* **Sự khác biệt giữa các lớp**: Advanced có đặc điểm rõ ràng (calo cao, mỡ thấp, tần suất tập nhiều), hỗ trợ bài toán phân loại.
* **Outlier**: Một số giá trị bất thường tồn tại nhưng không đáng kể, có thể giữ nguyên để bảo toàn thông tin.

Quá trình khai phá đã đặt nền tảng vững chắc cho việc xây dựng mô hình phân loại, với các đặc trưng được chọn (Calories\_Burned, Session\_Duration, v.v.) và dữ liệu được chuẩn bị kỹ lưỡng. Các thuật toán khai phá đề xuất (Random Forest, ANOVA, K-Means) không chỉ giúp trả lời các câu hỏi mà còn hỗ trợ tối ưu hóa hiệu suất mô hình, như đã chứng minh qua accuracy 91.79% của Random Forest.

# **VI. Đánh giá và chọn thuật toán**

Phần này trình bày quá trình lựa chọn, huấn luyện, và đánh giá các mô hình học máy nhằm phân loại mức độ kinh nghiệm (Experience\_Level: Beginner, Intermediate, Advanced) của các thành viên phòng gym dựa trên tập dữ liệu **"Gym Members Exercise Dataset"**. Các bước bao gồm chọn các mô hình phù hợp, tối ưu hóa siêu tham số, đánh giá hiệu suất, và so sánh để xác định mô hình tốt nhất. Ba thuật toán được sử dụng là K-Nearest Neighbors (KNN), Random Forest, và XGBoost, được huấn luyện trên tập dữ liệu đã tiền xử lý (tập huấn luyện: 778 mẫu, tập kiểm tra: 195 mẫu, 9 đặc trưng).

## **1. Chọn mô hình**

Việc lựa chọn mô hình học máy được thực hiện dựa trên đặc điểm của bài toán phân loại mức độ kinh nghiệm, cấu trúc của dữ liệu, và các yêu cầu về hiệu suất cũng như khả năng triển khai.

* **Đặc điểm bài toán**:
  + Bài toán là phân loại đa lớp (multi-class classification) với 3 nhãn: Beginner (0), Intermediate (1), và Advanced (2).
  + Dữ liệu có sự mất cân bằng nhẹ (Advanced ~19.6%, Beginner ~38.6%, Intermediate ~41.7%), đòi hỏi mô hình có khả năng xử lý tốt các lớp thiểu số.
  + Có mối quan hệ mạnh giữa một số đặc trưng (ví dụ: Calories\_Burned và Session\_Duration với tương quan ~0.91), yêu cầu mô hình có khả năng khai thác các mối quan hệ phức tạp.
* **Tiêu chí lựa chọn**:
  + **Khả năng xử lý dữ liệu không cân bằng**: Mô hình cần nhận diện tốt lớp thiểu số (Advanced) mà không bị thiên lệch về lớp đa số (Intermediate).
  + **Khả năng khai thác đặc trưng**: Mô hình cần tận dụng tốt các đặc trưng quan trọng như Calories\_Burned, Session\_Duration, và Fat\_Percentage.
  + **Hiệu suất và tốc độ**: Mô hình cần có độ chính xác cao, thời gian huấn luyện và dự đoán hợp lý để triển khai thực tế.
  + **Khả năng tối ưu hóa siêu tham số**: Mô hình cần dễ dàng điều chỉnh siêu tham số để đạt hiệu suất tốt nhất.
* **Các mô hình được chọn**:
  + **K-Nearest Neighbors (KNN)**:
    - Lý do: KNN là thuật toán đơn giản, không cần huấn luyện phức tạp, phù hợp để thử nghiệm ban đầu. Tuy nhiên, KNN nhạy cảm với dữ liệu không cân bằng và yêu cầu chuẩn hóa dữ liệu (đã thực hiện trong tiền xử lý).
  + **Random Forest**:
    - Lý do: Random Forest là thuật toán mạnh mẽ, có khả năng xử lý tốt dữ liệu không cân bằng và khai thác các mối quan hệ phi tuyến giữa các đặc trưng. Nó cũng cung cấp tầm quan trọng đặc trưng, hữu ích cho việc phân tích.
  + **XGBoost**:
    - Lý do: XGBoost là một thuật toán tăng tốc độ gradient boosting, nổi bật với khả năng xử lý dữ liệu không cân bằng, tối ưu hóa hiệu suất, và tốc độ dự đoán nhanh. Nó phù hợp với bài toán phân loại đa lớp và dữ liệu có tương quan mạnh giữa các đặc trưng.
* **Ý nghĩa**:
  + Ba mô hình được chọn đại diện cho các cách tiếp cận khác nhau: KNN (dựa trên khoảng cách), Random Forest (dựa trên tập hợp cây quyết định), và XGBoost (dựa trên tăng tốc độ gradient). Việc thử nghiệm cả ba mô hình giúp so sánh hiệu suất toàn diện và chọn ra mô hình tốt nhất.

## **2. Huấn luyện mô hình**

Ba mô hình được huấn luyện trên tập huấn luyện (778 mẫu) với các đặc trưng đã chuẩn hóa (Calories\_Burned, Session\_Duration (hours), Age, Weight (kg), Avg\_BPM, Resting\_BPM, Workout\_Frequency (days/week), Max\_BPM, Fat\_Percentage). Siêu tham số của mỗi mô hình được tối ưu hóa bằng GridSearchCV với kiểm tra chéo 5 lần (5-fold cross-validation) để đảm bảo hiệu suất tốt nhất.

from sklearn.neighbors import KNeighborsClassifier

from sklearn.ensemble import RandomForestClassifier

import xgboost as xgb

from sklearn.model\_selection import GridSearchCV

import time

# Dictionary để lưu mô hình và thời gian huấn luyện

models = {}

train\_times = {}

# 4.1 KNN

start\_time = time.time()

knn = KNeighborsClassifier()

param\_grid\_knn = {'n\_neighbors': range(1, 21)}

grid\_search\_knn = GridSearchCV(knn, param\_grid\_knn, cv=5, scoring='accuracy')

grid\_search\_knn.fit(X\_train, y\_train)

best\_knn = grid\_search\_knn.best\_estimator\_

train\_times['KNN'] = time.time() - start\_time

models['KNN'] = best\_knn

print(f"🎯 KNN - Siêu tham số tối ưu: {grid\_search\_knn.best\_params\_}")

# 4.2 Random Forest

start\_time = time.time()

rf = RandomForestClassifier(random\_state=42)

param\_grid\_rf = {'n\_estimators': [100, 200], 'max\_depth': [10, 20, None], 'min\_samples\_split': [2, 5]}

grid\_search\_rf = GridSearchCV(rf, param\_grid\_rf, cv=5, scoring='accuracy', n\_jobs=-1)

grid\_search\_rf.fit(X\_train, y\_train)

best\_rf = grid\_search\_rf.best\_estimator\_

train\_times['Random Forest'] = time.time() - start\_time

models['Random Forest'] = best\_rf

print(f"🎯 Random Forest - Siêu tham số tối ưu: {grid\_search\_rf.best\_params\_}")

# 4.3 XGBoost

start\_time = time.time()

xgb\_model = xgb.XGBClassifier(random\_state=42, objective='multi:softmax', num\_class=3)

param\_grid\_xgb = {'n\_estimators': [100, 200], 'max\_depth': [3, 6, 10], 'learning\_rate': [0.01, 0.1]}

grid\_search\_xgb = GridSearchCV(xgb\_model, param\_grid\_xgb, cv=5, scoring='accuracy', n\_jobs=-1)

grid\_search\_xgb.fit(X\_train, y\_train)

best\_xgb = grid\_search\_xgb.best\_estimator\_

train\_times['XGBoost'] = time.time() - start\_time

models['XGBoost'] = best\_xgb

print(f"🎯 XGBoost - Siêu tham số tối ưu: {grid\_search\_xgb.best\_params\_}")

**2.1. K-Nearest Neighbors (KNN)**

* **Tối ưu hóa siêu tham số**:
  + Sử dụng GridSearchCV để tìm số lượng hàng xóm tối ưu (n\_neighbors) trong khoảng từ 1 đến 20.
  + Kết quả: Siêu tham số tối ưu là n\_neighbors = 20.
* **Thời gian huấn luyện**:
  + Thời gian huấn luyện: 1.0119 giây.
  + KNN là thuật toán lười học (lazy learning), nên thời gian huấn luyện chủ yếu dành cho GridSearchCV.
* **Ý nghĩa**:
  + Giá trị n\_neighbors = 20 cho thấy KNN cần xem xét một số lượng lớn hàng xóm để đạt hiệu suất tốt, có thể do sự chồng lấn giữa các lớp (Beginner và Intermediate), như đã phát hiện trong EDA.

**2.2. Random Forest**

* **Tối ưu hóa siêu tham số**:
  + Sử dụng GridSearchCV để tìm các tham số tối ưu:
    - n\_estimators: Số lượng cây (100, 200).
    - max\_depth: Độ sâu tối đa của cây (10, 20, None).
    - min\_samples\_split: Số mẫu tối thiểu để chia một nút (2, 5).
  + Kết quả: Siêu tham số tối ưu là {'max\_depth': 10, 'min\_samples\_split': 5, 'n\_estimators': 200}.
* **Thời gian huấn luyện**:
  + Thời gian huấn luyện: 5.8192 giây.
  + Random Forest yêu cầu thời gian huấn luyện lâu hơn do số lượng cây lớn (200 cây) và kiểm tra chéo 5 lần.
* **Ý nghĩa**:
  + Độ sâu tối đa max\_depth = 10 và min\_samples\_split = 5 giúp mô hình tránh overfitting, đặc biệt với dữ liệu có sự mất cân bằng nhẹ (Advanced ~19.6%).
  + Số lượng cây n\_estimators = 200 đảm bảo mô hình ổn định và có khả năng tổng quát hóa tốt.

**2.3. XGBoost**

* **Tối ưu hóa siêu tham số**:
  + Sử dụng GridSearchCV để tìm các tham số tối ưu:
    - n\_estimators: Số lượng cây (100, 200).
    - max\_depth: Độ sâu tối đa của cây (3, 6, 10).
    - learning\_rate: Tốc độ học (0.01, 0.1).
  + Kết quả: Siêu tham số tối ưu là {'learning\_rate': 0.01, 'max\_depth': 3, 'n\_estimators': 100}.
* **Thời gian huấn luyện**:
  + Thời gian huấn luyện: 3.4265 giây.
  + XGBoost nhanh hơn Random Forest do độ sâu tối đa nhỏ (max\_depth = 3) và tốc độ học thấp (learning\_rate = 0.01).
* **Ý nghĩa**:
  + Độ sâu nhỏ (max\_depth = 3) và tốc độ học thấp (learning\_rate = 0.01) giúp mô hình tránh overfitting, đặc biệt hiệu quả với dữ liệu có sự mất cân bằng.
  + Số lượng cây n\_estimators = 100 là đủ để đạt hiệu suất cao mà không làm tăng thời gian huấn luyện quá mức.

## **3. Đánh giá mô hình**

Các mô hình được đánh giá trên tập kiểm tra (195 mẫu) dựa trên độ chính xác (accuracy), báo cáo phân loại (precision, recall, F1-score), và thời gian dự đoán. Kết quả được so sánh để xác định mô hình tốt nhất.

from sklearn.metrics import accuracy\_score, classification\_report

# Dictionary để lưu kết quả

results = {}

# Đánh giá từng mô hình

for model\_name, model in models.items():

    start\_time = time.time()

    y\_pred = model.predict(X\_test)

    predict\_time = time.time() - start\_time

    accuracy = accuracy\_score(y\_test, y\_pred)

    report = classification\_report(y\_test, y\_pred, target\_names=['Beginner', 'Intermediate', 'Advanced'])

    results[model\_name] = {

        'accuracy': accuracy,

        'train\_time': train\_times[model\_name],

        'predict\_time': predict\_time

    }

    print(f"\n🏆 Đánh giá {model\_name}:")

    print(f"Accuracy: {accuracy:.4f}")

    print("Báo cáo phân loại:\n", report)

# So sánh và chọn mô hình tốt nhất

best\_model\_name = max(results, key=lambda k: results[k]['accuracy'])

print(f"\n📊 So sánh hiệu suất các mô hình:")

for model\_name, metrics in results.items():

    print(f"{model\_name}:")

    print(f" - Accuracy: {metrics['accuracy']:.4f}")

    print(f" - Thời gian huấn luyện: {metrics['train\_time']:.4f} giây")

    print(f" - Thời gian dự đoán: {metrics['predict\_time']:.4f} giây")

print(f"\n🏆 Mô hình tốt nhất: {best\_model\_name} với accuracy {results[best\_model\_name]['accuracy']:.4f}")

import joblib

# Lưu mô hình tốt nhất và scaler

best\_model = models[best\_model\_name]

joblib.dump(best\_model, "model/best\_experience\_level\_model.pkl")

joblib.dump(scaler, "model/scaler.pkl")

print("💾 Mô hình tốt nhất và scaler đã được lưu!")

# Dự đoán mẫu mới

new\_data = pd.DataFrame([[800, 1.5, 30, 70, 140, 60, 4, 180, 20]],

                        columns=features)

new\_data\_scaled = scaler.transform(new\_data)

prediction = best\_model.predict(new\_data\_scaled)

level\_map = {0: 'Beginner', 1: 'Intermediate', 2: 'Advanced'}

print(f"🚀 Dự đoán cho mẫu mới: {level\_map[prediction[0]]}")

**3.1. K-Nearest Neighbors (KNN)**

* **Độ chính xác (Accuracy)**: 0.8718 (87.18%).
* **Báo cáo phân loại**:

A screenshot of a computer

AI-generated content may be incorrect.

- \*\*Thời gian dự đoán\*\*: 0.0141 giây.

- \*\*Phân tích\*\*:

- KNN đạt độ chính xác 87.18%, thấp nhất trong ba mô hình, nhưng vẫn ở mức chấp nhận được.

- Lớp \*\*Advanced\*\* đạt precision, recall, và F1-score đều là 1.00, cho thấy mô hình nhận diện hoàn hảo lớp này. Điều này có thể do số lượng mẫu của Advanced ít (~19.6%) và có đặc điểm rõ ràng (calo cao, tỷ lệ mỡ thấp, tần suất tập nhiều), như đã phát hiện trong EDA.

- Lớp \*\*Beginner\*\* có recall thấp nhất (0.81), cho thấy mô hình gặp khó khăn trong việc nhận diện đầy đủ các mẫu thuộc lớp này. Nguyên nhân có thể là sự chồng lấn với lớp Intermediate, như đã thấy qua phân bố tương đồng của các đặc trưng như `Calories\_Burned` (700-1100 calo) và `Fat\_Percentage` (20-30%) giữa hai lớp này.

- Lớp \*\*Intermediate\*\* có recall cao hơn (0.87), phù hợp với phân bố nhãn (Intermediate chiếm ~41.7%, lớp đa số).

- \*\*Thời gian dự đoán\*\* (0.0141 giây) cao hơn các mô hình khác, do KNN cần tính toán khoảng cách từ mỗi điểm kiểm tra đến tất cả các điểm trong tập huấn luyện, đặc biệt với `n\_neighbors = 20`.

- \*\*Ý nghĩa\*\*: - KNN cho thấy khả năng phân loại tốt với lớp thiểu số (Advanced), nhưng gặp khó khăn với lớp Beginner do sự chồng lấn. Thời gian dự đoán chậm khiến KNN ít phù hợp cho các ứng dụng yêu cầu tốc độ cao.

**3.2. Random Forest**

* **Độ chính xác (Accuracy)**: 0.9179 (91.79%).
* **Báo cáo phân loại**:

A screenshot of a computer

AI-generated content may be incorrect.

- \*\*Thời gian dự đoán\*\*: 0.0089 giây.

- \*\*Phân tích\*\*:

- Random Forest đạt độ chính xác 91.79%, cao hơn KNN, phản ánh khả năng khai thác tốt các mối quan hệ phi tuyến giữa các đặc trưng, chẳng hạn như tương quan mạnh giữa `Calories\_Burned` và `Session\_Duration` (~0.91).

- Lớp \*\*Advanced\*\* tiếp tục đạt F1-score 1.00, tương tự KNN, cho thấy mô hình nhận diện hoàn hảo lớp này nhờ các đặc trưng phân biệt rõ ràng.

- Lớp \*\*Beginner\*\* có recall 0.81, tương tự KNN, xác nhận rằng sự chồng lấn với Intermediate là một thách thức chung. Tuy nhiên, precision cao (0.98) cho thấy các dự đoán thuộc lớp Beginner của mô hình có độ chính xác cao.

- Lớp \*\*Intermediate\*\* đạt recall rất cao (0.99), cho thấy mô hình nhận diện gần như toàn bộ các mẫu thuộc lớp này. Điều này phù hợp với phân bố nhãn (Intermediate là lớp đa số, ~41.7%) và sự tối ưu hóa của Random Forest với `max\_depth = 10`, giúp tránh overfitting.

- \*\*Thời gian dự đoán\*\* (0.0089 giây) nhanh hơn KNN, do Random Forest chỉ cần đi qua các cây quyết định (200 cây) mà không cần tính toán khoảng cách.

- \*\*Ý nghĩa\*\*: - Random Forest thể hiện hiệu suất vượt trội so với KNN, đặc biệt trong việc nhận diện lớp Intermediate. Tuy nhiên, vấn đề với lớp Beginner vẫn tồn tại, cho thấy cần cải thiện thêm để tăng recall.

**3.3. XGBoost**

* **Độ chính xác (Accuracy)**: 0.9231 (92.31%).
* **Báo cáo phân loại**:

A screenshot of a computer

AI-generated content may be incorrect.

- \*\*Thời gian dự đoán\*\*: 0.0010 giây.

- \*\*Phân tích\*\*:

- XGBoost đạt độ chính xác cao nhất (92.31%), vượt qua Random Forest và KNN, cho thấy đây là mô hình hiệu quả nhất trong bài toán. - Lớp \*\*Advanced\*\* đạt F1-score 1.00, tương tự các mô hình khác, nhờ đặc điểm phân biệt rõ ràng (calo cao, tỷ lệ mỡ thấp).

- Lớp \*\*Beginner\*\* có recall 0.81, tương tự KNN và Random Forest, xác nhận rằng sự chồng lấn với Intermediate là vấn đề chung của cả ba mô hình. Tuy nhiên, precision đạt 1.00, cho thấy tất cả các dự đoán thuộc lớp Beginner đều chính xác.

- Lớp \*\*Intermediate\*\* đạt recall 1.00, cao nhất trong tất cả các mô hình, cho thấy XGBoost nhận diện hoàn hảo lớp này, có thể do sự tối ưu hóa tốt với `max\_depth = 3` và `learning\_rate = 0.01`, giúp mô hình tổng quát hóa hiệu quả trên lớp đa số (~41.7%).

- \*\*Thời gian dự đoán\*\* (0.0010 giây) là nhanh nhất, nhờ XGBoost được tối ưu hóa với độ sâu cây nhỏ (`max\_depth = 3`) và tốc độ học thấp, phù hợp cho các ứng dụng thời gian thực.

- \*\*Ý nghĩa\*\*: - XGBoost vượt trội về cả độ chính xác và tốc độ dự đoán, là lựa chọn lý tưởng cho bài toán. Hiệu suất cao trên lớp Intermediate và Advanced cho thấy mô hình xử lý tốt dữ liệu không cân bằng.

## **4. Tổng kết phần đánh giá**

- \*\*So sánh tổng thể\*\*:

- XGBoost đạt độ chính xác cao nhất (92.31%), tiếp theo là Random Forest (91.79%) và KNN (87.18%).

- Cả ba mô hình đều nhận diện hoàn hảo lớp Advanced (F1-score 1.00), nhưng gặp khó khăn với lớp Beginner (recall 0.81), do sự chồng lấn với Intermediate.

- XGBoost có thời gian dự đoán nhanh nhất (0.0010 giây), vượt trội so với Random Forest (0.0089 giây) và KNN (0.0141 giây), phù hợp cho các ứng dụng yêu cầu tốc độ cao.

- \*\*Liên hệ với EDA\*\*:

- Sự chồng lấn giữa Beginner và Intermediate (recall 0.81 cho Beginner) đã được dự đoán trong EDA, do phân bố tương đồng của `Calories\_Burned` (700-1100 calo) và `Fat\_Percentage` (20-30%) giữa hai lớp này.

- Lớp Advanced dễ nhận diện nhờ các đặc trưng phân biệt rõ ràng (calo cao, mỡ thấp, tần suất tập nhiều), như đã xác nhận qua EDA (Advanced thường có `Calories\_Burned` > 1000 calo, `Fat\_Percentage` < 20%).

- \*\*Ý nghĩa thực tế\*\*:

- XGBoost và Random Forest đều có hiệu suất cao, phù hợp để phân loại mức độ kinh nghiệm và hỗ trợ cá nhân hóa chương trình tập luyện (Beginner: Yoga, Advanced: HIIT).

- Thời gian dự đoán nhanh của XGBoost (0.0010 giây) cho phép triển khai trong các ứng dụng thời gian thực, như hệ thống gợi ý tập luyện tại phòng gym.

Phần đánh giá đã cung cấp cái nhìn chi tiết về hiệu suất của từng mô hình, làm cơ sở để so sánh và chọn mô hình tốt nhất trong bước tiếp theo.

# **VII. Kết quả và thảo luận**

Phần này trình bày các kết quả chính thu được từ quá trình khai phá dữ liệu và huấn luyện mô hình trên tập dữ liệu **"Gym Members Exercise Dataset"**, đồng thời thảo luận về những khía cạnh quan trọng, bao gồm điểm mạnh và điểm yếu của kết quả, trong bối cảnh bài toán **Phân loại mức độ kinh nghiệm** (Experience\_Level: Beginner, Intermediate, Advanced).

## **1. Kết quả chính**

**1.1. Kết quả khai phá dữ liệu**

Quá trình khai phá dữ liệu (EDA) đã mang lại những hiểu biết quan trọng về cấu trúc và đặc điểm của tập dữ liệu:

* **Phân bố nhãn**:
  + Nhãn Experience\_Level gồm 3 lớp: Beginner (376 mẫu, ~38.6%), Intermediate (406 mẫu, ~41.7%), và Advanced (191 mẫu, ~19.6%). Sự mất cân bằng nhẹ với lớp Advanced đã được xác nhận qua biểu đồ cột (Hình 3).
  + Trung bình nhãn là 1.81 (trước khi điều chỉnh), với độ lệch chuẩn 0.74, cho thấy dữ liệu thiên về lớp Intermediate.
* **Phân bố đặc trưng**:
  + Các đặc trưng quan trọng như Calories\_Burned (trung bình 905.42 calo, lệch phải), Session\_Duration (trung bình 1.26 giờ, gần chuẩn), và Fat\_Percentage (trung bình 24.98%, dao động 10-35%) cho thấy sự đa dạng về cường độ tập luyện và thể trạng (Hình 1).
  + Advanced thường có Calories\_Burned cao (>1000 calo), Fat\_Percentage thấp (<20%), và Workout\_Frequency cao (4-5 ngày/tuần), trong khi Beginner và Intermediate có sự chồng lấn ở nhiều đặc trưng.
* **Mối quan hệ giữa các đặc trưng**:
  + Ma trận tương quan (Hình 2) cho thấy mối quan hệ mạnh giữa Calories\_Burned và Session\_Duration (~0.91), và tương quan âm giữa Fat\_Percentage và Calories\_Burned (-0.60).
  + Không có cặp đặc trưng nào có tương quan quá cao (>0.95), loại bỏ nguy cơ đa cộng tuyến.
* **Sự khác biệt giữa các lớp**:
  + Advanced có đặc điểm rõ ràng (calo cao, mỡ thấp, tần suất tập nhiều), trong khi Beginner và Intermediate có sự chồng lấn, đặc biệt ở Calories\_Burned (700-1100 calo) và Fat\_Percentage (20-30%).

**1.2. Kết quả huấn luyện và đánh giá mô hình**

Ba mô hình K-Nearest Neighbors (KNN), Random Forest, và XGBoost được huấn luyện và đánh giá trên tập kiểm tra (195 mẫu):

* **Hiệu suất tổng thể**:
  + XGBoost đạt độ chính xác cao nhất (92.31%), tiếp theo là Random Forest (91.79%) và KNN (87.18%).
  + XGBoost cũng có thời gian dự đoán nhanh nhất (0.0010 giây), so với Random Forest (0.0089 giây) và KNN (0.0141 giây).
* **Hiệu suất theo lớp**:
  + Cả ba mô hình đều nhận diện hoàn hảo lớp **Advanced** (F1-score 1.00), nhờ số lượng mẫu ít (~19.6%) và đặc điểm phân biệt rõ ràng.
  + Lớp **Beginner** có recall thấp (0.81) ở cả ba mô hình, do sự chồng lấn với Intermediate, như đã xác nhận trong EDA.
  + Lớp **Intermediate** được nhận diện tốt nhất, đặc biệt với XGBoost (recall 1.00), phù hợp với phân bố nhãn (Intermediate chiếm ~41.7%).
* **Mô hình tốt nhất**:
  + XGBoost được chọn làm mô hình tốt nhất với độ chính xác 92.31%, thời gian dự đoán nhanh (0.0010 giây), và khả năng xử lý tốt dữ liệu không cân bằng.

## **2. Thảo luận**

**2.1. Điểm mạnh của kết quả**

* **Hiểu biết sâu sắc từ khai phá dữ liệu**:
  + Quá trình EDA đã xác định được các đặc trưng quan trọng (Calories\_Burned, Session\_Duration, Fat\_Percentage, Workout\_Frequency) có khả năng phân biệt cao giữa các lớp, đặc biệt là lớp Advanced. Điều này hỗ trợ hiệu quả cho việc huấn luyện mô hình, như thể hiện qua F1-score 1.00 của lớp Advanced ở cả ba mô hình.
  + Tương quan mạnh giữa Calories\_Burned và Session\_Duration (~0.91) và tương quan âm giữa Fat\_Percentage và Calories\_Burned (-0.60) cung cấp thông tin giá trị về mối quan hệ giữa cường độ tập luyện và thể trạng, giúp mô hình khai thác tốt các mẫu trong dữ liệu.
* **Hiệu suất cao của mô hình**:
  + XGBoost đạt độ chính xác 92.31%, vượt trội so với Random Forest (91.79%) và KNN (87.18%), cho thấy khả năng phân loại chính xác các mức kinh nghiệm.
  + Mô hình xử lý tốt lớp thiểu số (Advanced), với F1-score 1.00, phù hợp với mục tiêu của bài toán là nhận diện chính xác các nhóm để cá nhân hóa chương trình tập luyện.
  + Thời gian dự đoán nhanh của XGBoost (0.0010 giây) đáp ứng yêu cầu triển khai trong các ứng dụng thời gian thực, như hệ thống gợi ý tập luyện tại phòng gym.
* **Tính ứng dụng thực tế**:
  + Mô hình XGBoost có thể được sử dụng để phân loại mức độ kinh nghiệm của các thành viên phòng gym, từ đó hỗ trợ xây dựng các chương trình tập luyện cá nhân hóa. Ví dụ, người được phân loại là Beginner có thể được gợi ý các bài tập nhẹ nhàng như Yoga, trong khi Advanced phù hợp với các bài tập nặng như HIIT.

**2.2. Điểm yếu của kết quả**

* **Sự chồng lấn giữa Beginner và Intermediate**:
  + Recall thấp (0.81) cho lớp Beginner ở cả ba mô hình cho thấy sự khó khăn trong việc phân biệt Beginner và Intermediate. Điều này xuất phát từ sự chồng lấn trong phân bố của các đặc trưng như Calories\_Burned (700-1100 calo) và Fat\_Percentage (20-30%), như đã phát hiện trong EDA.
  + Hậu quả là một số thành viên Beginner có thể bị phân loại sai thành Intermediate, ảnh hưởng đến việc gợi ý chương trình tập luyện phù hợp.
* **Sự mất cân bằng nhãn**:
  + Lớp Advanced chỉ chiếm ~19.6%, dẫn đến sự mất cân bằng nhẹ. Mặc dù các mô hình nhận diện tốt lớp này, nhưng điều này có thể là do số lượng mẫu ít và đặc điểm rõ ràng, chứ không phải do mô hình thực sự xử lý tốt sự mất cân bằng.
  + Nếu dữ liệu thực tế có tỷ lệ lớp thay đổi (ví dụ: nhiều mẫu Advanced hơn), hiệu suất của mô hình có thể bị ảnh hưởng.
* **Hạn chế về đặc trưng**:
  + Các đặc trưng như Workout\_Type và Gender không được sử dụng trong mô hình do cần mã hóa phức tạp, nhưng chúng có thể chứa thông tin quan trọng để phân biệt giữa Beginner và Intermediate. Việc bỏ qua các đặc trưng này có thể làm giảm khả năng phân loại chính xác.
* **Thời gian huấn luyện của một số mô hình**:
  + Random Forest có thời gian huấn luyện lâu nhất (5.8192 giây), do số lượng cây lớn (n\_estimators = 200) và kiểm tra chéo 5 lần. Điều này có thể gây khó khăn khi huấn luyện trên tập dữ liệu lớn hơn hoặc trong các hệ thống yêu cầu cập nhật mô hình thường xuyên.

**2.3. Thảo luận thêm**

* **Tác động của tiền xử lý và EDA**:
  + Việc chuẩn hóa dữ liệu (MinMaxScaler) và chọn 9 đặc trưng quan trọng trong tiền xử lý đã giúp các mô hình (đặc biệt là KNN và XGBoost) hoạt động hiệu quả, như được thể hiện qua độ chính xác cao.
  + EDA đã đóng vai trò quan trọng trong việc dự đoán các vấn đề (như sự chồng lấn giữa Beginner và Intermediate) và xác định các đặc trưng quan trọng, từ đó định hướng tốt cho quá trình huấn luyện mô hình.
* **Khả năng cải thiện**:
  + Để giải quyết sự chồng lấn giữa Beginner và Intermediate, có thể áp dụng kỹ thuật cân bằng lớp như SMOTE hoặc thử nghiệm các thuật toán khác như Support Vector Machine (SVM).
  + Việc tích hợp thêm các đặc trưng như Workout\_Type (sau khi mã hóa) hoặc khám phá các đặc trưng mới (ví dụ: chỉ số sức bền) có thể cải thiện khả năng phân biệt giữa các lớp.
  + Sử dụng các phương pháp ensemble (kết hợp Random Forest và XGBoost) có thể tăng độ chính xác và độ ổn định của mô hình.

## **3. Tổng kết**

Kết quả khai phá dữ liệu và huấn luyện mô hình đã cung cấp cái nhìn toàn diện về tập dữ liệu và hiệu suất phân loại. Điểm mạnh nằm ở hiệu suất cao của XGBoost (accuracy 92.31%), khả năng nhận diện tốt lớp Advanced, và tính ứng dụng thực tế. Tuy nhiên, điểm yếu về sự chồng lấn giữa Beginner và Intermediate, sự mất cân bằng nhãn, và hạn chế về đặc trưng cần được giải quyết trong các nghiên cứu tiếp theo để cải thiện hiệu suất và tính tổng quát của mô hình.

# **VIII. Kết luận**

Phần này tổng kết các kết quả chính của dự án khai phá dữ liệu và phân loại mức độ kinh nghiệm trên tập dữ liệu **"Gym Members Exercise Dataset"**, đồng thời đưa ra kết luận về hiệu quả và khả năng áp dụng của quá trình khai phá dữ liệu.

## **1. Tổng kết các kết quả**

* **Khai phá dữ liệu**:
  + Quá trình khai phá dữ liệu đã làm rõ đặc điểm của tập dữ liệu: sự mất cân bằng nhãn (Advanced ~19.6%), các đặc trưng quan trọng (Calories\_Burned, Session\_Duration, Fat\_Percentage), và mối quan hệ giữa các đặc trưng (tương quan mạnh ~0.91 giữa Calories\_Burned và Session\_Duration).
  + Các phát hiện từ EDA đã hỗ trợ hiệu quả cho việc huấn luyện mô hình, giúp xác định các đặc trưng quan trọng và dự đoán các vấn đề như sự chồng lấn giữa Beginner và Intermediate.
* **Huấn luyện và đánh giá mô hình**:
  + Ba mô hình KNN, Random Forest, và XGBoost được huấn luyện và đánh giá, với XGBoost đạt hiệu suất cao nhất (accuracy 92.31%), tiếp theo là Random Forest (91.79%) và KNN (87.18%).
  + XGBoost được chọn làm mô hình tốt nhất nhờ độ chính xác cao, thời gian dự đoán nhanh (0.0010 giây), và khả năng xử lý tốt dữ liệu không cân bằng.
  + Tuy nhiên, recall thấp (0.81) cho lớp Beginner cho thấy sự hạn chế trong việc phân biệt Beginner và Intermediate, do sự chồng lấn trong phân bố đặc trưng.

## **2. Hiệu quả của quá trình khai phá dữ liệu**

* Quá trình khai phá dữ liệu và huấn luyện mô hình đã đạt được hiệu quả cao, với độ chính xác 92.31% từ mô hình XGBoost, chứng minh rằng dữ liệu và phương pháp được sử dụng phù hợp để giải quyết bài toán phân loại mức độ kinh nghiệm.
* Các phát hiện từ EDA (như đặc điểm của lớp Advanced, mối quan hệ giữa các đặc trưng) đã đóng vai trò quan trọng trong việc định hướng huấn luyện mô hình, giúp mô hình đạt hiệu suất cao và nhận diện tốt lớp thiểu số (Advanced, F1-score 1.00).
* Việc tối ưu hóa siêu tham số (GridSearchCV) và chuẩn hóa dữ liệu (MinMaxScaler) đã góp phần nâng cao hiệu suất, đặc biệt với XGBoost, vốn tận dụng tốt các đặc trưng quan trọng để phân loại.

## **3. Khả năng áp dụng thực tế**

* Mô hình XGBoost với độ chính xác 92.31% và thời gian dự đoán nhanh (0.0010 giây) có thể được triển khai trong các ứng dụng thực tế tại phòng gym, như hệ thống phân loại tự động mức độ kinh nghiệm của thành viên để gợi ý các chương trình tập luyện phù hợp. Ví dụ, người được phân loại là Beginner có thể được gợi ý các bài tập nhẹ nhàng như Yoga, trong khi Advanced phù hợp với các bài tập nặng như HIIT.
* Hệ thống này có thể được tích hợp vào các ứng dụng di động hoặc phần mềm quản lý phòng gym, cung cấp các gợi ý cá nhân hóa nhanh chóng và chính xác, từ đó nâng cao trải nghiệm của người dùng.

## **4. Hướng nghiên cứu trong tương lai**

* Để cải thiện hiệu suất, cần giải quyết sự chồng lấn giữa Beginner và Intermediate bằng cách áp dụng kỹ thuật cân bằng lớp (SMOTE) hoặc tích hợp thêm các đặc trưng như Workout\_Type (sau khi mã hóa).
* Khám phá các thuật toán khác (như SVM) hoặc các phương pháp ensemble (kết hợp Random Forest và XGBoost) có thể nâng cao độ chính xác và độ ổn định của mô hình.
* Thử nghiệm mô hình trên tập dữ liệu lớn hơn hoặc với các tỷ lệ lớp khác nhau để đánh giá tính tổng quát và khả năng mở rộng.

## **5. Kết luận cuối cùng**

Dự án khai phá dữ liệu và phân loại mức độ kinh nghiệm trên tập dữ liệu **"Gym Members Exercise Dataset"** đã thành công trong việc xây dựng mô hình XGBoost với độ chính xác 92.31%, đáp ứng tốt mục tiêu phân loại các mức kinh nghiệm của thành viên phòng gym. Quá trình khai phá dữ liệu đã cung cấp nền tảng vững chắc để huấn luyện mô hình, và kết quả đạt được có tính ứng dụng cao trong thực tế. Tuy nhiên, các hạn chế về sự chồng lấn giữa Beginner và Intermediate cần được giải quyết trong các nghiên cứu tiếp theo để nâng cao hiệu quả và tính tổng quát của mô hình.

# **LỜI CẢM ƠN**

Trước tiên, em xin gửi lời cảm ơn chân thành đến **Trường Đại học Sài Gòn** và **Khoa Công nghệ Thông tin** đã tạo điều kiện thuận lợi cho em trong suốt quá trình học tập và thực hiện tiểu luận này. Qua đó, em đã có cơ hội tiếp cận với những kiến thức chuyên sâu về khai thác dữ liệu và ứng dụng, một lĩnh vực đầy tiềm năng và ý nghĩa trong thời đại công nghệ hiện nay.

Em xin bày tỏ lòng biết ơn sâu sắc đến **Thầy Ths. Nguyễn Thanh Phước**, giảng viên hướng dẫn của em, người đã tận tình hướng dẫn, định hướng, và hỗ trợ em trong suốt quá trình thực hiện đề tài **“Phân loại mức độ kinh nghiệm của người tập (Experience Level Classification)”**. Những góp ý quý báu và sự động viên của Thầy đã giúp em hoàn thiện bài báo cáo này một cách tốt nhất.

Bên cạnh đó, em xin cảm ơn bạn bè, đồng nghiệp, và gia đình đã luôn ủng hộ, động viên em trong suốt quá trình học tập và nghiên cứu. Sự hỗ trợ tinh thần từ mọi người là nguồn động lực lớn để em hoàn thành tiểu luận này.

Mặc dù đã cố gắng hết sức, bài báo cáo không thể tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp từ thầy cô và các bạn để tiểu luận được hoàn thiện hơn.

**Thành phố Hồ Chí Minh, tháng 5 năm 2025**

**Sinh viên thực hiện**

**Hồ Văn Quyến**

# **TÀI LIỆU THAM KHẢO**

Trong quá trình thực hiện tiểu luận **“Phân loại mức độ kinh nghiệm của người tập (Experience Level Classification)”**, em đã tham khảo các nguồn tài liệu và tài nguyên sau để xây dựng và hoàn thiện bài báo cáo:

1. **Tài liệu học thuật**
   * Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Education.  
     (Nguồn tham khảo về các thuật toán học máy như K-Nearest Neighbors, Random Forest, và các phương pháp phân loại đa lớp.)
   * Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.  
     (Nguồn tham khảo về các kỹ thuật khai thác dữ liệu, phân tích dữ liệu khám phá, và tối ưu hóa mô hình học máy.)
2. **Tài liệu trực tuyến**
   * Scikit-learn Documentation. (2025). *Scikit-learn: Machine Learning in Python*. Truy cập tại: <https://scikit-learn.org/stable/documentation.html>.  
     (Tham khảo cách triển khai các thuật toán KNN, Random Forest, và các công cụ như GridSearchCV, MinMaxScaler trong Python.)
   * XGBoost Documentation. (2025). *XGBoost: Scalable and Accurate Implementation of Gradient Boosting*. Truy cập tại: <https://xgboost.readthedocs.io/en/stable/>.  
     (Tham khảo cách sử dụng và tối ưu hóa mô hình XGBoost cho bài toán phân loại đa lớp.)
   * Kaggle Dataset. (2025). *Gym Members Exercise Dataset*. Truy cập tại: <https://www.kaggle.com/datasets/gym-members-exercise-dataset>.  
     (Nguồn dữ liệu chính được sử dụng trong bài báo cáo, bao gồm thông tin về các đặc trưng như Calories\_Burned, Session\_Duration, và Experience\_Level.)
3. **Bài viết và tài liệu hỗ trợ**
   * Brownlee, J. (2023). *A Gentle Introduction to XGBoost for Applied Machine Learning*. Machine Learning Mastery. Truy cập tại: <https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/>.  
     (Tham khảo cách áp dụng XGBoost và tối ưu hóa siêu tham số cho bài toán phân loại.)
   * Seaborn Documentation. (2025). *Seaborn: Statistical Data Visualization*. Truy cập tại: <https://seaborn.pydata.org/>.  
     (Tham khảo cách sử dụng Seaborn để trực quan hóa dữ liệu, bao gồm biểu đồ histogram, heatmap, và countplot.)

Tất cả các tài liệu trên đã được sử dụng để hỗ trợ quá trình nghiên cứu, phân tích, và triển khai mô hình trong bài báo cáo. Em xin chân thành cảm ơn các tác giả và tổ chức đã cung cấp những nguồn tài liệu quý giá này.