

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN CLC**

-----□□-----



Báo cáo hệ thống SmartFit
Môn: Thực tập cơ sở

Giảng viên: Đặng Hoàng Long

Sinh viên:

Phạm Anh Minh – B22DCAT192

Phạm Tuấn Anh - B22DCVT033

Lê Huy Hoàng – B22DCVT205

Nguyễn Đình Anh - B22DCVT023

Hà Nội 2025

Mục lục

Mục lục	2
Lời nói đầu	4
I. Giới thiệu	6
1. Mục tiêu báo cáo	6
2. Tóm tắt hệ thống	6
2.1. Cách thức hoạt động của hệ thống	6
2.2. Các chức năng chính của hệ thống	6
2.3. Kết quả đạt được	7
II. Mô hình AI trong hệ thống	7
A. Mô hình dự đoán lượng năng lượng tiêu thụ (Calories Burnt)	7
1. Mục tiêu mô hình	7
2. Tổng quan quy trình huấn luyện mô hình	7
3. Dữ liệu huấn luyện	9
3.1. Mô tả dữ liệu	9
3.2. Quy trình thu thập và xử lý dữ liệu (Data Preprocessing)	9
4. Mô hình học máy (ML Models)	10
5. Kỹ thuật huấn luyện	10
5.1. Quy trình huấn luyện và kiểm thử	10
5.2. Các chỉ số đánh giá mô hình	11
6. Đánh giá kết quả mô hình	12
6.1. Kết quả mô hình	12
6.2. Đánh giá mô hình	12
6.3. Phân tích tầm quan trọng của đặc trưng (Feature Importance)	12
6.4. Chọn mô hình tối ưu	14
7. Thách thức và điều chỉnh	14
7.1. Các thách thức trong quá trình huấn luyện mô hình	14
7.2. Các điều chỉnh đã thực hiện	15
B. Mô hình gợi ý thực đơn dinh dưỡng sau buổi tập	15
1. Mục tiêu mô hình	15
2. Dữ liệu huấn luyện	15
2.1. Mô tả dữ liệu	15
2.2. Quy trình thu thập và xử lý dữ liệu (Data Preprocessing)	16
3. Mô hình gợi ý thực đơn (Rule-Based Recommendation System)	17
3.1. Cách hoạt động	17
3.2. Lý do lựa chọn phương pháp	17

3.3. Quy trình gợi ý thực đơn	17
4. Đánh giá kết quả	18
III. Kiến trúc hệ thống (Software Engineering)	19
1. Tổng quan kiến trúc hệ thống	19
2. Kiến trúc Phần mềm (Software Architecture)	20
2.1 Kiến trúc tổng thể	22
2.2. Biểu đồ Use Case	23
2.3 Biểu đồ lớp (Class Diagram)	24
2.4 Biểu đồ tuần tự (Sequence Diagram)	26
2.4.1 Tính lượng năng lượng tiêu thụ	26
2.4.2 Xem lịch sử	28
2.4.3 Quản lý thông tin cá nhân	29
2.4.4 Quản lý thực đơn	30
3. Các thành phần hệ thống	31
3.1 Front-end	31
3.2 Back-end	32
3.3 Tích hợp mô hình AI	33
4. Tích hợp và Triển khai	33
4.1 Quy trình triển khai	33
4.2 Quy trình kiểm thử	36
IV. Tổng kết dự án	40
1. Tổng kết	40
2. Kết luận	40
3. Đề xuất cải tiến	41
Lời cảm ơn	44

Lời nói đầu

Trong kỷ nguyên số hóa mạnh mẽ và sự bùng nổ của thông tin như hiện nay, con người ngày càng nhận thức sâu sắc về tầm quan trọng của việc duy trì một lối sống lành mạnh, đặc biệt là thông qua chế độ tập luyện thể chất và dinh dưỡng hợp lý. Tuy nhiên, việc xây dựng và tuân thủ một kế hoạch cá nhân hóa phù hợp với thể trạng và mục tiêu sức khỏe riêng biệt của mỗi người lại thường gặp phải nhiều thách thức, từ việc thiếu kiến thức chuyên sâu đến việc khó khăn trong việc theo dõi và điều chỉnh liên tục.

Nhận thấy nhu cầu cấp thiết này, chúng tôi đã phát triển một hệ thống thông minh ứng dụng Trí tuệ Nhân tạo (AI) nhằm mang đến một giải pháp toàn diện và cá nhân hóa cao cho người dùng. Hệ thống của chúng tôi không chỉ đơn thuần là một công cụ hỗ trợ mà còn là một "huấn luyện viên" ảo, đồng hành cùng người dùng trên hành trình tối ưu hóa sức khỏe của mình. Với sự hỗ trợ của AI, chúng tôi có thể vượt qua những hạn chế của các phương pháp truyền thống, cung cấp các phân tích sâu sắc và gợi ý chính xác dựa trên dữ liệu cá nhân của mỗi người.

Cụ thể, hệ thống này tập trung vào hai trụ cột chính: ước tính lượng năng lượng tiêu thụ trong mỗi buổi tập và đề xuất các thực đơn dinh dưỡng khoa học sau khi vận động. Bằng cách phân tích các yếu tố như tuổi, giới tính, chiều cao, cân nặng, cùng với thời gian và cường độ tập luyện (nhịp tim, nhiệt độ cơ thể), mô hình AI của chúng tôi có khả năng dự đoán lượng calo đốt cháy một cách đáng tin cậy. Dựa trên thông tin này và các mục tiêu dinh dưỡng cá nhân hóa (như giảm cân, tăng cơ, giữ cân, hoặc giảm mỡ), hệ thống sẽ đưa ra các gợi ý thực đơn phù hợp, đảm bảo người dùng nạp vào đúng lượng và loại dinh dưỡng cần thiết để phục hồi và đạt được kết quả mong muốn.

Không dừng lại ở việc đưa ra các khuyến nghị một chiều, hệ thống còn hướng đến khả năng theo dõi tiến độ và tự động điều chỉnh kế hoạch theo thời gian, học hỏi từ dữ liệu và phản hồi của người dùng để ngày càng trở nên thông minh và phù hợp hơn. Tất cả các tính năng này được gói gọn trong một giao diện người dùng (UI) thân thiện, trực quan, giúp người dùng dễ dàng tương tác và quản lý kế hoạch sức khỏe của mình mọi lúc, mọi nơi.

Báo cáo này sẽ đi sâu vào mô tả chi tiết về kiến trúc tổng thể của hệ thống, phân tích kỹ lưỡng các mô hình AI đã được lựa chọn và huấn luyện, bao gồm cả quy trình thu thập, xử lý dữ liệu và các kỹ thuật đánh giá hiệu suất.

Chúng tôi tin tưởng rằng, với việc ứng dụng mạnh mẽ AI và sự chú trọng vào trải nghiệm người dùng, hệ thống này không chỉ là một dự án nghiên cứu mà còn là một bước tiến tiềm năng trong việc hỗ trợ cộng đồng xây dựng một lối sống khỏe mạnh và bền vững.

I. Giới thiệu

1. Mục tiêu báo cáo

Báo cáo này nhằm mục đích trình bày chi tiết về **Hệ thống SmartFit** do chúng tôi phát triển. Báo cáo sẽ mô tả tổng thể hệ thống, đi sâu vào kiến trúc phần mềm, các mô hình Trí tuệ Nhân tạo (AI) cốt lõi đã được huấn luyện, quy trình hoạt động, và những thành tựu đạt được. Mục tiêu chính là cung cấp cái nhìn toàn diện về cách thức hệ thống sử dụng AI để cá nhân hóa các đề xuất, từ dự đoán lượng calo tiêu thụ đến gợi ý thực đơn dinh dưỡng, nhằm hỗ trợ người dùng đạt được mục tiêu sức khỏe một cách khoa học và bền vững.

2. Tóm tắt hệ thống

Hệ thống quản lý kế hoạch tập luyện và thực đơn dinh dưỡng được thiết kế với mục đích hỗ trợ người dùng đạt được mục tiêu sức khỏe cá nhân thông qua các kế hoạch tập luyện được cá nhân hóa và thực đơn dinh dưỡng được đề xuất dựa trên các yếu tố sức khỏe cụ thể. Hệ thống tích hợp AI để xử lý và phân tích dữ liệu, từ đó đưa ra các gợi ý chính xác và phù hợp cho từng người dùng.

2.1. Cách thức hoạt động của hệ thống

Hệ thống hoạt động theo quy trình ba bước chính:

1. Thu thập và xử lý dữ liệu đầu vào: Người dùng sẽ nhập các thông tin cá nhân như tuổi, giới tính, chiều cao, cân nặng, cũng như các chỉ số liên quan đến cường độ tập luyện như nhịp tim và thời gian tập.
2. Phân tích và dự đoán: Các mô hình AI sẽ phân tích các thông số đã nhập và sử dụng các thuật toán như Linear Regression, Random Forest Regressor, và XGBoost để tính toán lượng năng lượng tiêu thụ và đề xuất thực đơn dinh dưỡng phù hợp với mục tiêu sức khỏe của người dùng.
3. Cung cấp kết quả và theo dõi tiến độ: Dựa trên các phân tích và dự đoán từ mô hình AI, hệ thống sẽ cung cấp các gợi ý về kế hoạch tập luyện và thực đơn dinh dưỡng. Đồng thời, người dùng cũng có thể theo dõi tiến độ và nhận các đề xuất điều chỉnh khi cần thiết.

2.2. Các chức năng chính của hệ thống

- Dự đoán năng lượng tiêu thụ:
Hệ thống sẽ tính toán và đưa ra lượng calo tiêu thụ dựa trên thời gian và cường độ tập luyện của người dùng.
- Gợi ý thực đơn dinh dưỡng:

Sau mỗi buổi tập, hệ thống sẽ cung cấp thực đơn dinh dưỡng phù hợp để bù đắp lại lượng năng lượng tiêu thụ và đảm bảo sự phát triển cơ thể tối ưu.

- Cập nhật kế hoạch và theo dõi tiến trình:

Sau mỗi buổi tập, hệ thống sẽ tự động cập nhật kế hoạch tập luyện và dinh dưỡng dựa trên các dữ liệu mới nhất, giúp người dùng dễ dàng theo dõi và điều chỉnh các mục tiêu sức khỏe của mình.

2.3. Kết quả đạt được

Với sự tích hợp của mô hình AI mạnh mẽ và giao diện người dùng dễ sử dụng, hệ thống đã giúp người dùng đạt được các mục tiêu sức khỏe cá nhân một cách hiệu quả hơn. Các thuật toán AI đã cung cấp các dự đoán chính xác về lượng năng lượng tiêu thụ và thực đơn dinh dưỡng phù hợp, giúp người dùng duy trì thể trạng tốt và phát triển cơ thể một cách khoa học.

Ngoài ra, hệ thống cũng giúp người dùng có thể theo dõi sự tiến bộ của mình và nhận được các lời khuyên thiết thực để điều chỉnh kế hoạch tập luyện và dinh dưỡng sao cho phù hợp nhất với nhu cầu và mục tiêu cá nhân. Kết quả cuối cùng là một hệ thống toàn diện, vừa hiệu quả về mặt khoa học, vừa dễ tiếp cận và sử dụng đối với người dùng.

II. Mô hình AI trong hệ thống

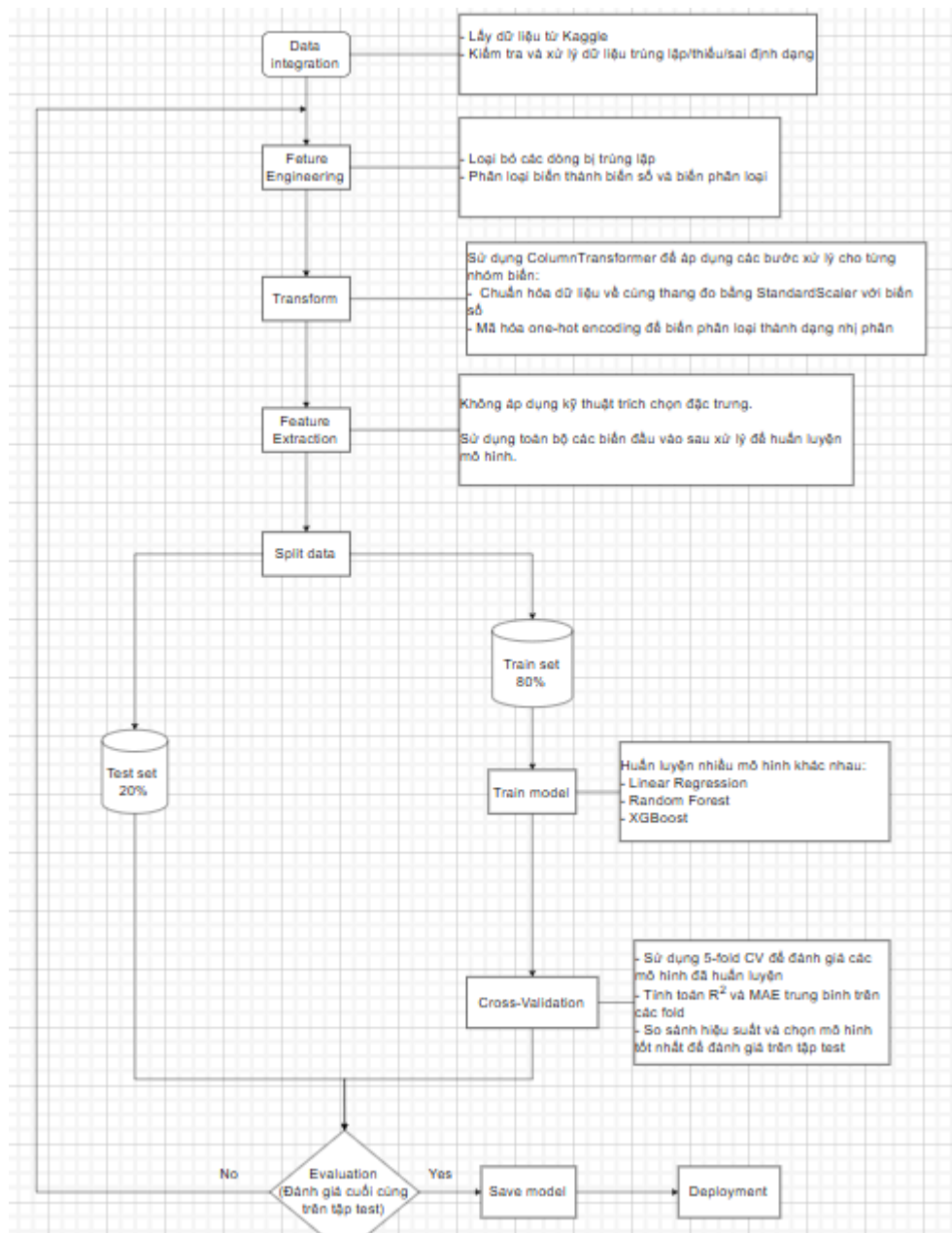
A. Mô hình dự đoán lượng năng lượng tiêu thụ (Calories Burnt)

1. Mục tiêu mô hình

Mô hình sẽ ước tính lượng calo mà người dùng đốt cháy trong quá trình tập luyện dựa trên các yếu tố đầu vào như tuổi, giới tính, chiều cao, cân nặng, thời gian và cường độ tập luyện (nhịp tim, nhiệt độ cơ thể).

2. Tổng quan quy trình huấn luyện mô hình

Để đạt được mục tiêu dự đoán lượng calo tiêu thụ, chúng tôi đã xây dựng một pipeline xử lý và huấn luyện dữ liệu. Sơ đồ dưới đây minh họa tổng thể các bước trong quy trình này, từ giai đoạn tiền xử lý dữ liệu đầu vào đến việc tích hợp và huấn luyện mô hình hồi quy.



Sơ đồ pipeline

Pipeline này được thiết kế để tự động hóa toàn bộ quy trình từ dữ liệu thô đến kết quả dự đoán. Nó bao gồm hai thành phần chính:

- Tiền xử lý dữ liệu (ColumnTransformer):

Đây là thành phần chịu trách nhiệm tiền xử lý dữ liệu đầu vào. Nó bao gồm OrdinalEncoder để chuyển đổi các đặc trưng phân loại (Gender) thành dạng số, và StandardScaler để chuẩn hóa các đặc trưng số (Age, Height, Weight, Duration, Heart_Rate, Body_Temp). Mục đích là chuẩn bị dữ liệu về cùng một định dạng và tỉ lệ, giúp các thuật toán học máy hoạt động hiệu quả.

- Mô hình hồi quy:

Sau khi dữ liệu được tiền xử lý, chúng sẽ được đưa vào các mô hình hồi quy như `LinearRegression`, `RandomForestRegressor`, hoặc `XGBRegressor` để thực hiện nhiệm vụ dự đoán lượng calo tiêu thụ.

Các bước tiếp theo trong báo cáo sẽ đi sâu vào chi tiết của từng thành phần này, dữ liệu sử dụng, và cách chúng được huấn luyện, đánh giá.

3. Dữ liệu huấn luyện

3.1. Mô tả dữ liệu

Dữ liệu dự đoán lượng calo tiêu thụ (`calories.csv` và `exercise.csv`):

- Đặc trưng (Features): `User_ID`, `Gender`, `Age`, `Height`, `Weight`, `Duration`, `Heart_Rate`, `Body_Temp`.
- Tập dữ liệu này được hợp nhất từ hai file `calories.csv` và `exercise.csv` dựa trên `User_ID`.

3.2. Quy trình thu thập và xử lý dữ liệu (Data Preprocessing)

1. Đọc và hợp nhất dữ liệu: Dữ liệu từ `calories.csv` và `exercise.csv` được đọc và hợp nhất thành một `DataFrame` duy nhất (`data`) dựa trên cột `User_ID`.
2. Kiểm tra và xử lý Missing Values: Mã đã kiểm tra các giá trị null (`data.isnull().sum()`) và kết quả cho thấy không có giá trị null trong tập dữ liệu.
3. Kiểm tra và xử lý trùng lặp: Hàm `check_duplicates(data)` được sử dụng để kiểm tra các hàng bị trùng lặp.
4. Phân tách đặc trưng và nhãn: Cột `Calories` được xác định là biến mục tiêu (`y`), các cột còn lại là đặc trưng (`X`). Cột `User_ID` được loại bỏ khỏi tập đặc trưng (`X`) vì nó không mang ý nghĩa dự đoán.
5. Mã hóa và Chuẩn hóa (Column Transformer và Pipeline):
 - Mã hóa Categorical Features: Cột `Gender` (giới tính) là một biến phân loại (object) và được mã hóa bằng `OrdinalEncoder()`.
 - Chuẩn hóa Numerical Features: Các cột số như `Age`, `Height`, `Weight`, `Duration`, `Heart_Rate`, `Body_Temp` được chuẩn hóa bằng `StandardScaler()`. Điều này giúp đưa các đặc trưng về cùng một tỷ lệ, tránh việc các đặc trưng có giá trị lớn hơn chi phối mô hình.

- ColumnTransformer được sử dụng để áp dụng các phép biến đổi này một cách đồng thời, với remainder='passthrough' để giữ lại các cột không được biến đổi (nếu có).

4. Mô hình học máy (ML Models)

Trong dự án này, ba mô hình hồi quy có độ phức tạp khác nhau đã được sử dụng để dự đoán lượng calo tiêu thụ:

- **Hồi quy tuyến tính (Linear Regression):**

Đây là một mô hình cơ bản và thường là điểm khởi đầu tốt để thiết lập một baseline. Nó đơn giản, dễ hiểu và nhanh chóng để huấn luyện. Nó giả định mối quan hệ tuyến tính giữa các đặc trưng và biến mục tiêu.

- **Rừng ngẫu nhiên (Random Forest Regressor):**

Random Forest là một mô hình học ensemble mạnh mẽ, kết hợp nhiều cây quyết định để đưa ra dự đoán. Nó nổi tiếng với khả năng xử lý tốt các mối quan hệ phi tuyến tính và ít bị overfitting hơn so với một cây quyết định đơn lẻ. Nó cũng ít nhạy cảm với outlier và không cần feature scaling quá chặt chẽ (mặc dù đã được thực hiện trong pipeline).

- **XGBoost (Extreme Gradient Boosting Regressor):**

XGBoost là một thuật toán gradient boosting rất hiệu quả và phổ biến, thường đạt được hiệu suất cao nhất trong các bài toán hồi quy và phân loại. Nó tối ưu hóa hiệu suất bằng cách xây dựng các cây quyết định tuần tự, mỗi cây cố gắng sửa lỗi của cây trước đó. XGBoost cũng có khả năng xử lý missing values và tích hợp nhiều kỹ thuật regularization để tránh overfitting.

5. Kỹ thuật huấn luyện

5.1. Quy trình huấn luyện và kiểm thử

1. Tách dữ liệu huấn luyện và kiểm thử (Train-Test Split):

Dữ liệu được chia thành tập huấn luyện (80%) và tập kiểm thử (20%) bằng cách sử dụng train_test_split với random_state=42 để đảm bảo tính tái lập. Việc này giúp đánh giá khả năng tổng quát hóa của mô hình trên dữ liệu mới.

2. Xây dựng Pipeline:

Một Pipeline được xây dựng để kết hợp các bước tiền xử lý dữ liệu (ColumnTransformer) và mô hình học máy. Điều này đảm bảo rằng các

phép biến đổi dữ liệu được áp dụng nhất quán cho cả tập huấn luyện và kiểm thử, đồng thời giúp quy trình huấn luyện gọn gàng hơn.

3. Huấn luyện mô hình:

Mỗi pipeline (với từng mô hình hồi quy) được huấn luyện trên tập dữ liệu huấn luyện (X_{train} , y_{train}).

4. Đánh giá trên tập kiểm thử:

Sau khi huấn luyện, mô hình được sử dụng để dự đoán trên tập kiểm thử (X_{test}), và các chỉ số đánh giá được tính toán.

5. Kiểm định chéo (Cross - Validation):

KFold với $n_splits=5$ (5-fold cross-validation) và $shuffle=True$, $random_state=42$ được sử dụng để đánh giá độ bền vững của mô hình. $cross_val_score$ tính toán điểm R2 trên 5 tập con khác nhau của dữ liệu, và giá trị trung bình được báo cáo. Kỹ thuật này giúp có được ước tính đáng tin cậy hơn về hiệu suất mô hình bằng cách giảm thiểu sự phụ thuộc vào một lần chia dữ liệu cụ thể.

5.2. Các chỉ số đánh giá mô hình

Các chỉ số sau đây đã được sử dụng để đánh giá hiệu suất của các mô hình hồi quy:

- R-squared (R2):

Ý nghĩa: Đại diện cho tỷ lệ phương sai của biến phụ thuộc có thể được giải thích bởi các biến độc lập trong mô hình. Giá trị R2 càng gần 1, mô hình càng phù hợp với dữ liệu.

Cách sử dụng: Được sử dụng để so sánh mức độ phù hợp của các mô hình khác nhau. `cv_results.mean()` cung cấp giá trị R2 trung bình từ cross-validation.

- Mean Absolute Error (MAE):

Ý nghĩa: Đo lường độ lớn trung bình của các lỗi (sự khác biệt tuyệt đối giữa các dự đoán và giá trị thực). MAE cung cấp một thước đo trực quan về mức độ chính xác của các dự đoán.

Cách sử dụng: Giá trị MAE càng nhỏ, mô hình càng chính xác.

6. Đánh giá kết quả mô hình

6.1. Kết quả mô hình

Model	R2	MAE	CV_R2
Linear Regression	0.967294	8.441514	0.967140
Random Forest	0.998262	1.694910	0.997935
XGBoost	0.998868	1.498120	0.998851

Chỉ số R2, MAE và CV_R2 của từng mô hình

6.2. Đánh giá mô hình

- Linear Regression:

Đạt được một hiệu suất khá tốt với R2 là 0.967. Điều này cho thấy mối quan hệ tuyến tính giữa các đặc trưng và lượng calo tiêu thụ là đáng kể. Tuy nhiên, MAE là 8.44 cho thấy dự đoán trung bình có thể sai lệch khoảng 8.44 đơn vị calo.

- Random Forest Regressor:

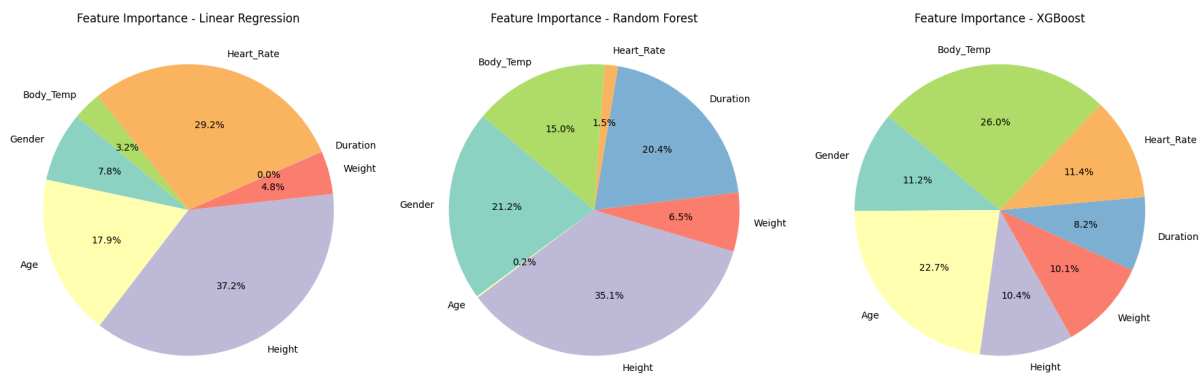
Hiệu suất của Random Forest vượt trội đáng kể so với Linear Regression, với R2 đạt 0.998262 và MAE giảm xuống chỉ còn 1.69491. Điều này chứng tỏ Random Forest có khả năng nắm bắt tốt hơn các mối quan hệ phi tuyến tính phức tạp trong dữ liệu.

- XGBoost Regressor:

XGBoost là mô hình đạt hiệu suất tốt nhất, với R2 cao nhất (0.998868) và MAE thấp nhất (1.498120). Giá trị CV_R2 (0.998851) cũng rất gần với R2 trên tập kiểm thử, cho thấy mô hình có tính ổn định và khả năng tổng quát hóa tốt.

6.3. Phân tích tầm quan trọng của đặc trưng (Feature Importance)

Sau khi huấn luyện, chúng tôi đã tiến hành phân tích tầm quan trọng của các đặc trưng đối với mỗi mô hình đã huấn luyện. Kết quả được thể hiện ở biểu đồ tròn sau:



Biểu đồ Feature Importance của từng mô hình

- Đối với Linear Regression:

Mô hình hồi quy tuyến tính cho thấy Duration (Thời gian tập luyện) là đặc trưng có ảnh hưởng lớn nhất, chiếm 50.4% tầm quan trọng. Heart_Rate (Nhịp tim) đứng thứ hai với 26.3%, theo sau là Weight (Cân nặng) với 9.6% và Height (Chiều cao) với 6.8%. Age (Tuổi) và Body_Temp (Nhiệt độ cơ thể) có tầm quan trọng thấp hơn, lần lượt là 4.5% và 2.3%. Điều này cho thấy mối quan hệ tuyến tính mạnh mẽ giữa thời gian tập luyện và nhịp tim với lượng calo đốt cháy.

- Đối với Random Forest:

Mô hình Random Forest cũng xác nhận Duration là đặc trưng quan trọng nhất, chiếm 46.5% tầm quan trọng. Heart_Rate tiếp tục là yếu tố ảnh hưởng đáng kể với 31.7%. Các đặc trưng khác như Body_Temp (7.6%), Age (6.2%), Height (4.4%) và Weight (3.7%) cũng đóng góp vào dự đoán. Random Forest, với khả năng xử lý các mối quan hệ phi tuyến tính, vẫn đặt trọng tâm vào các yếu tố liên quan trực tiếp đến hoạt động thể chất.

- Đối với XGBoost:

Tương tự như hai mô hình trên, Duration là đặc trưng nổi bật nhất với 45.4% tầm quan trọng, và Heart_Rate là đặc trưng quan trọng thứ hai với 33.7%. Body_Temp (8.0%), Age (6.3%), Height (3.6%) và Weight (2.9%) cũng có vai trò trong mô hình. XGBoost, mô hình mạnh mẽ nhất, cũng khẳng định tầm ảnh hưởng vượt trội của thời gian và nhịp tim.

Nhận xét tổng quát:

Qua phân tích tầm quan trọng của đặc trưng từ cả ba mô hình, một điểm chung nổi bật là Duration (Thời gian tập luyện) và Heart_Rate (Nhịp tim) là hai đặc trưng có ảnh hưởng quan trọng nhất trong việc dự đoán lượng calo tiêu thụ. Điều này là hoàn toàn hợp lý về mặt sinh học và thể thao, bởi vì thời gian và cường độ của một buổi tập (thể hiện qua nhịp tim) trực tiếp quyết định lượng năng lượng mà cơ thể đốt cháy.

Các đặc trưng khác như Body_Temp (Nhiệt độ cơ thể), Age (Tuổi), Height (Chiều cao) và Weight (Cân nặng) cũng có đóng góp nhất định vào mô hình, tuy nhiên với mức độ ảnh hưởng thấp hơn so với Duration và Heart_Rate. Body_Temp có thể phản ánh mức độ nỗ lực và quá trình trao đổi chất, trong khi Age, Height, Weight là các yếu tố nền tảng ảnh hưởng đến chỉ số BMI và mức năng lượng tiêu hao cơ bản của mỗi người.

Việc nhận diện rõ ràng các đặc trưng quan trọng này không chỉ giúp chúng tôi hiểu được các yếu tố chính ảnh hưởng đến lượng calo đốt cháy mà còn có ý nghĩa quan trọng trong việc thu thập và chuẩn bị dữ liệu trong tương lai. Để tối ưu hóa hiệu suất dự đoán, cần đảm bảo rằng dữ liệu cho Duration và Heart_Rate được thu thập chính xác và đầy đủ nhất. Phân tích này cũng tăng cường khả năng giải thích của mô hình, giúp người dùng hiểu rõ hơn về cách các hoạt động của họ ảnh hưởng đến lượng calo tiêu hao.

6.4. Chọn mô hình tối ưu

Dựa trên các chỉ số đánh giá, XGBoost Regressor là mô hình tối ưu cho tác vụ dự đoán lượng calo tiêu thụ. Nó không chỉ đạt R2 cao nhất mà còn có MAE thấp nhất, cho thấy độ chính xác vượt trội và sai số dự đoán nhỏ nhất. Sự ổn định của kết quả cross-validation cũng củng cố cho lựa chọn này, đảm bảo rằng mô hình không chỉ hoạt động tốt trên một tập dữ liệu cụ thể mà còn có khả năng tổng quát hóa tốt trên các dữ liệu chưa từng thấy.

7. Thách thức và điều chỉnh

7.1. Các thách thức trong quá trình huấn luyện mô hình

- Định lượng cường độ tập luyện:

Heart_Rate và Body_Temp là các đặc trưng quan trọng để đánh giá cường độ, nhưng việc thu thập dữ liệu chính xác và nhất quán cho các yếu tố này có thể là một thách thức trong môi trường thực tế.

- Tương tác giữa các đặc trưng:
Các mô hình Linear Regression có thể gặp khó khăn với các tương tác phức tạp giữa các đặc trưng. Random Forest và XGBoost đã giải quyết tốt vấn đề này.

7.2. Các điều chỉnh đã thực hiện

- Sử dụng Pipeline và ColumnTransformer:
Việc kết hợp tiền xử lý và mô hình thành một pipeline đã giúp quy trình huấn luyện hiệu quả, sạch sẽ và đảm bảo tính nhất quán trong các bước biến đổi dữ liệu.
- So sánh đa dạng các mô hình:
Việc thử nghiệm Linear Regression, Random Forest và XGBoost đã giúp xác định mô hình phù hợp nhất với dữ liệu và bài toán cụ thể này.
- Đánh giá bằng Cross-Validation:
Sử dụng cross-validation là một cải tiến quan trọng, giúp đánh giá hiệu suất mô hình một cách robust hơn, giảm thiểu rủi ro overfitting và cung cấp cái nhìn chân thực hơn về khả năng tổng quát hóa của mô hình.

B. Mô hình gợi ý thực đơn dinh dưỡng sau buổi tập

1. Mục tiêu mô hình

Mục tiêu của mô hình gợi ý thực đơn dinh dưỡng là cung cấp các gợi ý thực đơn cá nhân hóa và khoa học cho người dùng sau khi họ hoàn thành buổi tập. Dựa trên lượng năng lượng tiêu thụ được dự đoán từ mô hình đầu tiên và các thông tin cá nhân cơ bản (như mục tiêu dinh dưỡng cá nhân đã được xác định trước hoặc thông qua phân tích BMI và hoạt động), mô hình này sẽ phân loại và đề xuất các thực đơn phù hợp để hỗ trợ người dùng đạt được mục tiêu sức khỏe cụ thể của họ. Các mục tiêu dinh dưỡng được hỗ trợ bao gồm: giảm cân, tăng cơ, giữ cân, hoặc giảm mỡ, đảm bảo người dùng có một chế độ ăn uống cân bằng và hiệu quả sau khi vận động.

2. Dữ liệu huấn luyện

2.1. Mô tả dữ liệu

- Dữ liệu được sử dụng để huấn luyện và phát triển mô hình gợi ý thực đơn dinh dưỡng là tập tin menu.csv, chứa thông tin chi tiết về các bữa ăn mẫu. Đây là một tập dữ liệu quan trọng, cho phép chúng tôi phân tích thành

phần dinh dưỡng của từng thực đơn để phân loại chúng theo các mục tiêu sức khỏe khác nhau.

- Đặc trưng (Features):
 - + menu: Mô tả chi tiết của từng thực đơn (ví dụ: 'Trái chuối (82g) + Salad rau củ (151g) + Sữa tươi không đường (200ml)').
 - + total_calories: Tổng lượng calo của thực đơn.
 - + total_protein: Tổng lượng tinh bột (gram) trong thực đơn.
 - + total_fat: Tổng lượng chất béo (gram) trong thực đơn.
 - + total_carb: Tổng lượng chất đạm (gram) trong thực đơn.
 - + protein_ratio, fat_ratio, carb_ratio: Các tỷ lệ năng lượng mà protein, chất béo và carbohydrate đóng góp vào tổng calo của thực đơn. Đây là các đặc trưng phái sinh quan trọng, được tính toán từ các đặc trưng thô, giúp định lượng tỷ lệ macro dinh dưỡng theo một cách chuẩn hóa, làm cơ sở cho việc phân loại.
- Nhãn (Labels):

goal: Nhãn mục tiêu dinh dưỡng mà thực đơn đó được phân loại vào, bao gồm các loại như giảm_mỡ, tăng_cơ, giảm_cân, tăng_cân, giữ_cân, và không_phù_hợp. Nhãn này là yếu tố then chốt giúp hệ thống xác định thực đơn nào phù hợp với mục tiêu của người dùng.

2.2. Quy trình thu thập và xử lý dữ liệu (Data Preprocessing)

1. Đọc dữ liệu: File menu.csv được đọc vào DataFrame menu_df.
2. Tính toán Macro Ratios: Một hàm compute_macro_ratio được định nghĩa để tính toán tỷ lệ năng lượng từ protein, chất béo và carbohydrate (protein_ratio, fat_ratio, carb_ratio) dựa trên tổng calo và lượng của từng macro. Điều này giúp định lượng thành phần dinh dưỡng của mỗi thực đơn.
3. Gán nhãn mục tiêu: Một hàm classify_goal được tạo để gán nhãn goal (như giảm_mỡ, tăng_cơ, giảm_cân, tăng_cân, giữ_cân, không_phù_hợp) cho mỗi thực đơn dựa trên các tỷ lệ macro này. Các ngưỡng được định nghĩa rõ ràng trong hàm:
 - giảm_mỡ: protein ≥ 0.24 , fat ≤ 0.24 , carb ≤ 0.50
 - tăng_cơ: protein ≥ 0.24 , fat ≤ 0.30 , carb ≥ 0.38
 - giảm_cân: protein ≥ 0.22 , fat ≤ 0.30 , carb < 0.52
 - tăng_cân: protein ≥ 0.14 , fat ≤ 0.35 , carb ≥ 0.50
 - giữ_cân: protein ≥ 0.18 , fat ≤ 0.40
 - không_phù_hợp: nếu không thuộc các trường hợp trên.

Việc gán nhãn này chuyển đổi dữ liệu thực đơn thành một tập dữ liệu có nhãn, sẵn sàng cho việc phân loại. Đây là một phương pháp tiếp cận dựa trên luật (rule-based) để phân loại thực đơn, phù hợp cho giai đoạn phát triển ban đầu của hệ thống.

4. Lưu hoặc xem trước dữ liệu đã xử lý: Sau khi hoàn thành các bước xử lý, DataFrame `menu_df` được lưu lại dưới dạng `menu_labeled.csv` (nếu cần) hoặc xem trước để kiểm tra chất lượng dữ liệu và phân phối nhãn.

3. Mô hình gợi ý thực đơn (Rule-Based Recommendation System)

3.1. Cách hoạt động

Hệ thống hoạt động dựa trên một hệ thống phân loại dựa trên luật (rule-based classification system) đã được định nghĩa và áp dụng trong bước tiền xử lý dữ liệu. Các luật này được xây dựng dựa trên tỷ lệ năng lượng từ protein, chất béo và carbohydrate (`protein_ratio`, `fat_ratio`, `carb_ratio`) của từng thực đơn.

3.2. Lý do lựa chọn phương pháp

- Tính minh bạch và dễ giải thích:
Trong lĩnh vực dinh dưỡng và sức khỏe, sự minh bạch về cách thức gợi ý được tạo ra là cực kỳ quan trọng. Các luật rõ ràng giúp người dùng và các chuyên gia dễ dàng hiểu tại sao một thực đơn cụ thể lại được đề xuất cho một mục tiêu dinh dưỡng nhất định. Điều này tăng cường sự tin cậy và khả năng giải thích của hệ thống.
- Phù hợp với mục tiêu ban đầu:
Mục tiêu của giai đoạn phát triển hiện tại là xây dựng một nền tảng gợi ý dựa trên các nguyên tắc dinh dưỡng cơ bản và các tiêu chí phân loại đã được xác định rõ ràng. Việc áp dụng trực tiếp các luật đã biết là cách tiếp cận hiệu quả nhất cho giai đoạn này.
- Hiệu quả và khả năng kiểm soát:
Khi các tiêu chí phân loại đã được biết và có thể định lượng, việc sử dụng hệ thống luật giúp kiểm soát chính xác đầu ra, đảm bảo rằng mọi gợi ý đều tuân thủ các nguyên tắc dinh dưỡng mà chúng tôi muốn truyền tải. Điều này giúp giảm thiểu sự phức tạp không cần thiết của mô hình học máy.

3.3. Quy trình gợi ý thực đơn

Quy trình tập trung vào việc áp dụng các luật đã định nghĩa để gán nhãn cho từng thực đơn và sau đó là quy trình gợi ý dựa trên nhu cầu người dùng:

1. Phân loại và Gán nhãn Thực đơn: Toàn bộ tập dữ liệu thực đơn (menu_df) được xử lý thông qua hàm `classify_goal`, nơi mỗi thực đơn được gán một nhãn mục tiêu dinh dưỡng (goal) phù hợp dựa trên các tỷ lệ macro dinh dưỡng của nó.
2. Lọc và Đề xuất: Khi người dùng cung cấp thông tin về mục tiêu dinh dưỡng cá nhân của họ (ví dụ: giảm cân, tăng cơ), hệ thống sẽ truy vấn và lọc ra các thực đơn trong tập dữ liệu đã được gán nhãn có goal tương ứng. Các thực đơn này sau đó có thể được sắp xếp hoặc chọn lọc thêm (ví dụ: dựa trên tổng calo phù hợp với lượng calo tiêu thụ của người dùng) để đưa ra gợi ý phù hợp nhất.

4. Đánh giá kết quả

Khác với các mô hình học máy cần chỉ số hiệu suất như Accuracy hay F1-score, chất lượng của hệ thống gợi ý dựa trên luật được đánh giá chủ yếu qua sự hợp lý, khoa học và minh bạch của các luật đã định nghĩa. Hệ thống này đảm bảo rằng mọi gợi ý đều tuân thủ chặt chẽ các nguyên tắc dinh dưỡng đã được thiết lập.

Tính chính xác của luật: Các ngưỡng tỷ lệ macro dinh dưỡng được sử dụng để phân loại được xây dựng dựa trên các khuyến nghị phổ biến trong ngành dinh dưỡng, đảm bảo tính hợp lệ về mặt khoa học của các gợi ý.

Khả năng mở rộng: Hệ thống luật có thể dễ dàng được điều chỉnh hoặc mở rộng bằng cách thêm các luật mới hoặc thay đổi các ngưỡng hiện có, cho phép hệ thống thích nghi linh hoạt với các yêu cầu dinh dưỡng phức tạp hơn trong tương lai hoặc khi có thêm thông tin về các khuyến nghị chuyên sâu.

Đánh giá hiệu quả trong thực tế: Trong các giai đoạn phát triển tiếp theo, chất lượng và hiệu quả của hệ thống gợi ý có thể được đánh giá sâu hơn thông qua phản hồi trực tiếp từ người dùng về mức độ hài lòng, tính phù hợp với nhu cầu cá nhân, và khả năng hỗ trợ họ đạt được mục tiêu sức khỏe. Đây là một bước quan trọng để tinh chỉnh và tối ưu hóa hệ thống dựa trên dữ liệu thực tế.

III. Kiến trúc hệ thống (Software Engineering)

1. Tổng quan kiến trúc hệ thống

- Mục tiêu hệ thống SmartFit là một hệ thống tích hợp trí tuệ nhân tạo giúp người dùng:
 - Dự đoán lượng calo tiêu thụ sau tập luyện.
 - Gợi ý thực đơn dinh dưỡng phù hợp với mục tiêu cá nhân (giảm cân, tăng cơ, giữ cân,...).
 - Theo dõi lịch sử sức khỏe, luyện tập và dinh dưỡng.
- Hệ thống SmartFit được thiết kế theo mô hình 3 lớp (Three-tier Architecture):

Frontend (Giao diện người dùng)

- Công nghệ sử dụng: HTML, CSS, JavaScript, Chart.js
- Chức năng:
 - Cho phép người dùng nhập thông tin cá nhân và chỉ số tập luyện.
 - Gửi yêu cầu đến backend để nhận kết quả dự đoán calo.
 - Hiển thị biểu đồ dinh dưỡng (Chart.js).
 - Giao diện thân thiện, tương thích đa nền tảng.

Backend (Xử lý và kết nối)

- Ngôn ngữ sử dụng: PHP + Python
- Chức năng:
 - PHP: Quản lý người dùng, session, lưu dữ liệu vào MySQL, xử lý logic ứng dụng.
 - Python (Flask API): Thực hiện dự đoán calo và gợi ý thực đơn dinh dưỡng thông qua mô hình máy học.
 - Cung cấp các RESTful API:
 - /predict: Nhận dữ liệu đầu vào, trả về calo tiêu thụ.
 - /recommend_menu: Gợi ý thực đơn theo mục tiêu.
 - /health: Kiểm tra trạng thái hệ thống.

AI Service (Dịch vụ dự đoán bằng mô hình học máy)

- Ngôn ngữ: Python + scikit-learn
- Thành phần chính:
 - pipeline.pkl: Mô hình AI được huấn luyện sẵn dùng để dự đoán calo.
 - Flask App (app.py): Giao tiếp giữa backend PHP và mô hình AI.
- Chức năng:
 - Xử lý dữ liệu đầu vào.
 - Dự đoán lượng calo dựa trên 6 chỉ số: tuổi, chiều cao, cân nặng, thời gian tập, nhịp tim, nhiệt độ cơ thể.
 - Gợi ý thực đơn phù hợp với mục tiêu người dùng.

- Kiến trúc cơ sở dữ liệu

- Cơ sở dữ liệu sử dụng: MySQL
- Các bảng chính:
 - users: Quản lý thông tin người dùng.
 - user_profile: Hồ sơ sức khỏe cá nhân.
 - workout_history: Lưu lịch sử tập luyện, calo đốt, thực đơn đề xuất.
 - menus: Lưu trữ danh sách thực đơn mẫu với thành phần dinh dưỡng và mục tiêu tương ứng.

2. Kiến trúc Phần mềm (Software Architecture)

Kiến trúc Phần mềm – Hệ thống SmartFit

- Mô hình kiến trúc tổng thể

Hệ thống SmartFit sử dụng mô hình kiến trúc phần mềm multi-layered architecture (kiến trúc phân lớp) kết hợp microservice cho AI module. Cụ thể:

Các lớp chính:

1. Presentation Layer (Giao diện người dùng)
2. Application Layer (Controller + Xử lý yêu cầu)
3. Data Layer (Cơ sở dữ liệu)
4. AI Microservice Layer (Flask API + Model ML)

- Mô tả các thành phần kiến trúc

1. Presentation Layer (Frontend)

- Công nghệ: HTML, CSS, JavaScript, Chart.js
- Chức năng:
 - Giao tiếp trực tiếp với người dùng.
 - Nhập thông tin: tuổi, cân nặng, chiều cao, v.v.
 - Gửi yêu cầu tính toán đến backend.
 - Hiển thị kết quả và biểu đồ dinh dưỡng (Chart.js).

2. Application Layer (PHP Controller)

- Công nghệ: PHP
- Vai trò:
 - Nhận dữ liệu từ frontend.
 - Xác thực session đăng nhập.
 - Gọi các API (Flask) để dự đoán calo hoặc gợi ý thực đơn.
 - Lưu kết quả vào cơ sở dữ liệu nếu người dùng đã đăng nhập.

3. Business Logic Layer

- Ngôn ngữ: PHP + Python
- Vai trò:
 - PHP xử lý logic như kiểm tra mục tiêu, lọc thực đơn theo calo.
 - Python xử lý logic phức tạp về mô hình học máy và phân tích dinh dưỡng.

4. Data Layer

- Công nghệ: MySQL
- Thành phần chính:
 - users: Quản lý tài khoản người dùng.
 - user_profile: Hồ sơ sức khỏe cá nhân.
 - workout_history: Lưu các buổi tập luyện + thực đơn gợi ý.
 - menus: Dữ liệu thực đơn + thành phần dinh dưỡng.

5. AI Microservice Layer

- Ngôn ngữ: Python (Flask)
- Thành phần:
 - app.py: Xây dựng API /predict, /recommend_menu, /health

- pipeline.pkl: Mô hình học máy dự đoán calo (đã huấn luyện bằng scikit-learn)
- Chức năng:
 - Nhận dữ liệu JSON từ PHP controller.
 - Dự đoán lượng calo tiêu thụ dựa trên 6 chỉ số sinh học.
 - Trả lại kết quả dưới dạng JSON.
 - Gợi ý thực đơn phù hợp theo mục tiêu và lượng calo đã đốt.

2.1 Kiến trúc tổng thể

Về mặt triển khai tổng thể, hệ thống SmartFit được tổ chức theo hướng tách biệt các thành phần, đảm bảo tính module và dễ dàng phát triển độc lập. Sơ đồ tổng thể có thể mô tả như sau:

- Frontend:

- Là phần giao diện tĩnh chạy trên trình duyệt, cho phép người dùng nhập dữ liệu và nhận phản hồi.
- Giao tiếp với backend thông qua các lệnh AJAX hoặc fetch request.

- Backend (PHP):

- Xử lý yêu cầu từ giao diện, kiểm tra phiên người dùng, kết nối với cơ sở dữ liệu và gửi yêu cầu đến AI.
- Là điểm kết nối trung tâm giữa người dùng, cơ sở dữ liệu và dịch vụ trí tuệ nhân tạo.

- AI Microservice (Python Flask):

- Chạy độc lập ở cổng riêng (mặc định là 5000).
- Nhận dữ liệu JSON từ backend và trả về kết quả dự đoán dưới dạng JSON.
- Sử dụng mô hình pipeline.pkl đã được huấn luyện với scikit-learn.

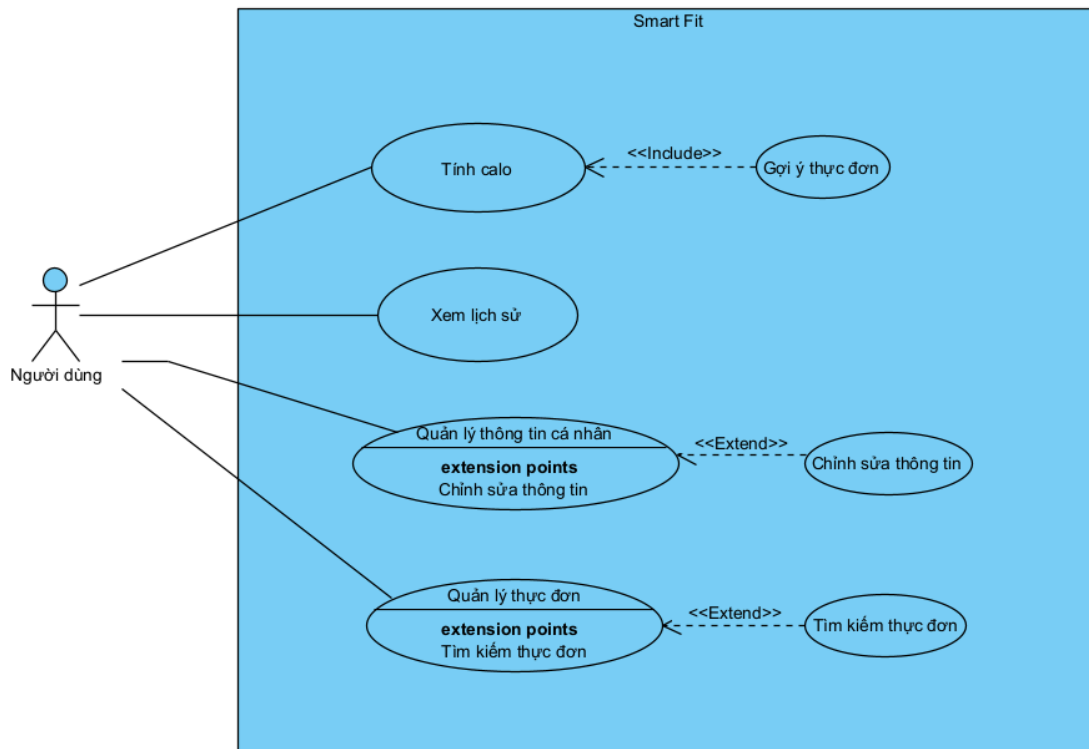
- Cơ sở dữ liệu (MySQL):

- Lưu trữ tập trung toàn bộ dữ liệu người dùng, bài tập, lịch sử dự đoán và thực đơn mẫu.
- Các bảng chính: users, user_profile, workout_history, menus.

- Giao tiếp và bảo mật:

- Backend PHP kiểm soát truy cập bằng session (\$_SESSION).
- AI API chỉ giao tiếp nội bộ (không trực tiếp mở ra ngoài).
- Giao tiếp dữ liệu qua JSON (RESTful API), thuận tiện cho tích hợp về sau.

2.2. Biểu đồ Use Case



Biểu đồ Use Case Diagram

- Use Case Diagram: Mô tả các hành động chính của người dùng trong hệ thống, bao gồm tính calo, xem lịch sử, quản lý thông tin cá nhân, quản lý thực đơn.

- Mô tả các use case:

Tính lượng năng lượng tiêu thụ:

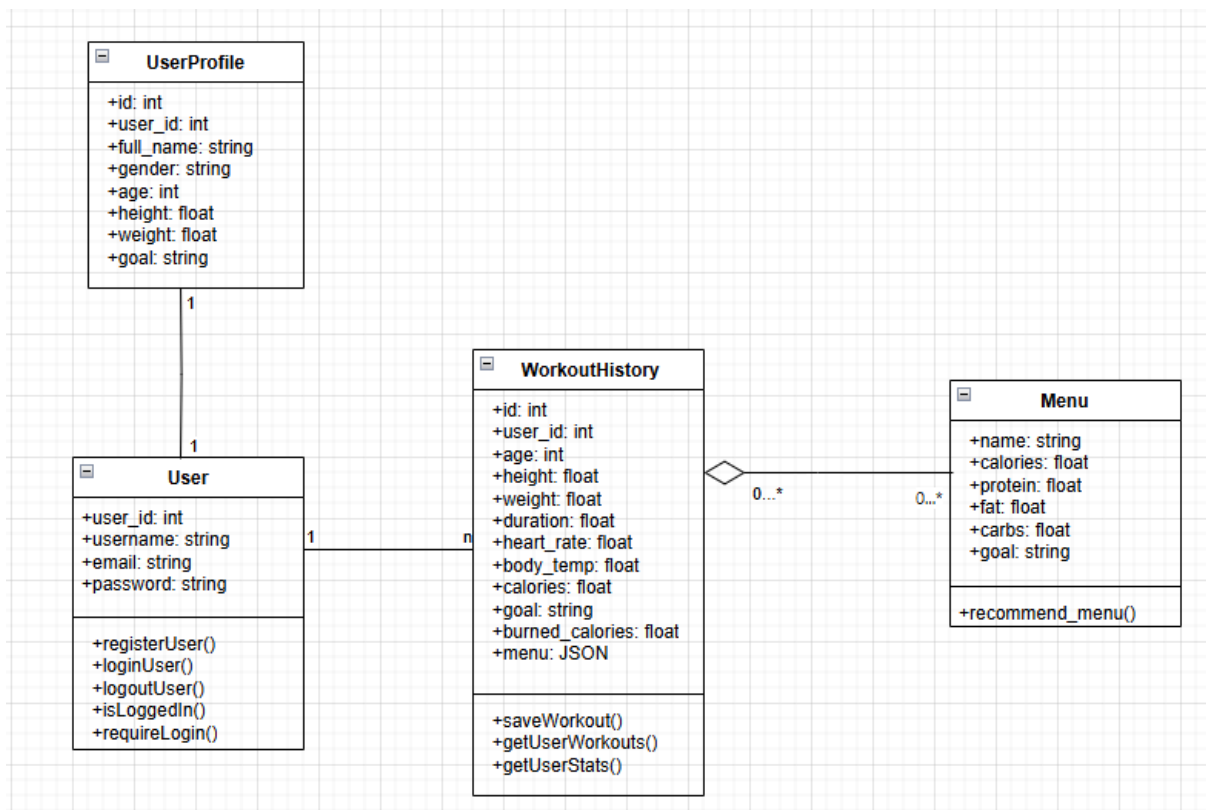
- Người dùng có thể sử dụng chức năng tính calo để xác định lượng calo đã tiêu thụ hoặc cần tiêu thụ trong các buổi tập.
- Include: Kết quả tính calo sẽ luôn kèm theo việc Gợi ý thực đơn, giúp người dùng có thực đơn dinh dưỡng phù hợp.

○ Xem lịch sử:

- Người dùng có thể xem lại các thông tin lịch sử lượng năng lượng tiêu hao và theo dõi quá trình tiến triển mục tiêu sức khỏe của mình.

- Quản lý thông tin cá nhân:
 - Người dùng có thể quản lý thông tin cá nhân của mình, bao gồm thay đổi các thông tin như chiều cao, cân nặng, tuổi, mục tiêu.
 - Extend: Người dùng có thể chỉnh sửa thông tin trong trường hợp cần thay đổi thông tin cá nhân.
- Quản lý thực đơn:
 - Người dùng có thể xem và quản lý các thực đơn dinh dưỡng trong hệ thống.
 - Extend: Người dùng có thể Tìm kiếm thực đơn phù hợp với mục tiêu sức khỏe của mình.

2.3 Biểu đồ lớp (Class Diagram)



Biểu đồ lớp

● Thuyết minh

1. User

- Thuộc tính:
 - `user_id`: Mã định danh người dùng (int).

- username: Tên đăng nhập (string).
- email: Địa chỉ email (string).
- password: Mật khẩu (string).
- Phương thức:
 - registerUser(): Đăng ký tài khoản mới.
 - loginUser(): Đăng nhập hệ thống.
 - logoutUser(): Đăng xuất khỏi hệ thống.
 - isLoggedIn(): Kiểm tra trạng thái đăng nhập.
 - requireLogin(): Yêu cầu đăng nhập để truy cập chức năng.
- Mối quan hệ:
 - Một User có một UserProfile (1-1).
 - Một User có nhiều WorkoutHistory (1-n).

2. UserProfile

- Thuộc tính:
 - id: Mã định danh hồ sơ (int).
 - user_id: Liên kết đến User (int).
 - full_name: Họ tên đầy đủ (string).
 - gender: Giới tính (string).
 - age: Tuổi (int).
 - height: Chiều cao (float).
 - weight: Cân nặng (float).
 - goal: Mục tiêu tập luyện (string).
- Mối quan hệ:
 - Mỗi UserProfile thuộc về một User.

3. WorkoutHistory

- Thuộc tính:
 - id: Mã định danh lịch sử (int).
 - user_id: Liên kết đến User (int).
 - age: Tuổi tại thời điểm tập (int).
 - height: Chiều cao (float).
 - weight: Cân nặng (float).
 - duration: Thời lượng tập (float).
 - heart_rate: Nhịp tim (float).
 - body_temp: Nhiệt độ cơ thể (float).
 - calories: Lượng calo tiêu thụ (float).

- goal: Mục tiêu tập luyện (string).
- burned_calories: Lượng calo đã đốt cháy (float).
- menu: Thực đơn gợi ý (JSON).
- Phương thức:
 - saveWorkout(): Lưu lịch sử tập luyện.
 - getUserWorkouts(): Lấy danh sách lịch sử tập luyện của user.
 - getUserStats(): Lấy thống kê tập luyện của user.
- Mỗi quan hệ:
 - Nhiều WorkoutHistory thuộc về một User (n-1).
 - WorkoutHistory có thể chứa nhiều Menu (aggregation).

4. Menu

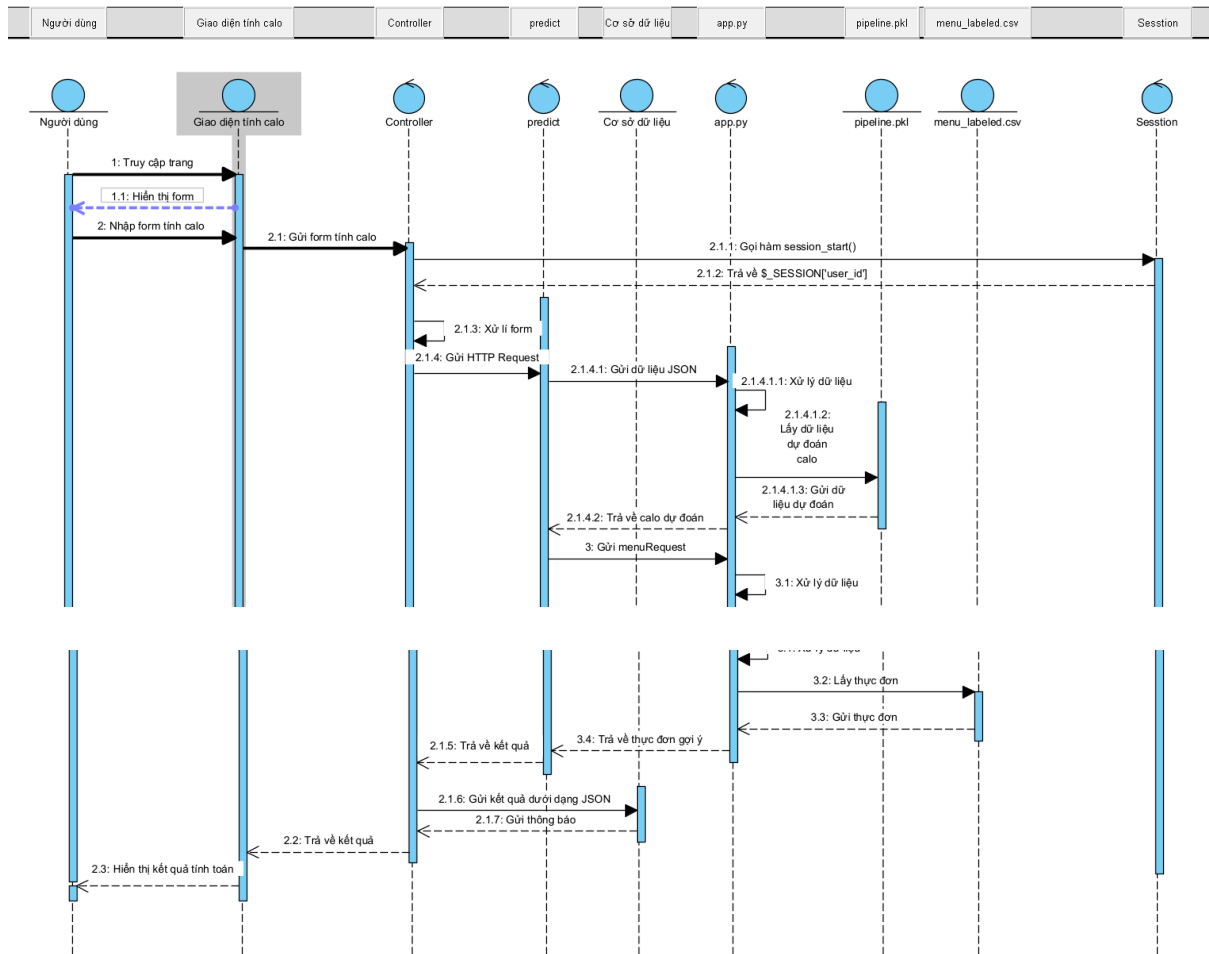
- Thuộc tính:
 - name: Tên món ăn (string).
 - calories: Lượng calo (float).
 - protein: Lượng protein (float).
 - fat: Lượng chất béo (float).
 - carbs: Lượng carbohydrate (float).
 - goal: Mục tiêu phù hợp (string).
- Phương thức:
 - recommend_menu(): Gợi ý thực đơn phù hợp với mục tiêu và lượng calo tiêu thụ.
- Mỗi quan hệ:
 - Menu được liên kết với WorkoutHistory thông qua thuộc tính menu (aggregation, không phải quan hệ trực tiếp).

2.4 Biểu đồ tuần tự (Sequence Diagram)

2.4.1 Tính lượng năng lượng tiêu thụ

- Thuyết minh:
 1. Người dùng truy cập trang web và yêu cầu giao diện tính calo.
 2. Giao diện hiển thị form tính calo để người dùng nhập thông tin.
 3. Người dùng nhập các thông tin cần thiết vào form như tuổi, chiều cao, cân nặng, mục tiêu, thời gian tập, nhiệt độ cơ thể.
 4. Sau khi người dùng nhấn nút “Tính”, một HTTP request được gửi đến controller.

5. Controller nhận dữ liệu từ form và gọi hàm dự đoán trong bước tiếp theo.
6. Controller gọi hàm `session_start()` gửi yêu cầu đến Session.
7. Session phản hồi trả về `user_id` của người dùng.
8. Controller gửi yêu cầu dự đoán tới mô hình (predict), gọi API và nhận dữ liệu dự đoán về từ model.
9. Ứng dụng gửi dữ liệu vào [app.py](#) để thực hiện các tính toán dựa trên mô hình đã huấn luyện.
10. Dữ liệu sau khi được truyền qua `pipeline.pkl` sẽ được xử lý để tạo ra kết quả cuối cùng.
11. Các giá trị và kết quả dự đoán được lưu vào cơ sở dữ liệu để sử dụng lại trong tương lai.
12. Sau khi nhận được kết quả từ model, giao diện hiển thị kết quả tính toán cho người dùng.
13. Kết quả tính toán (calo đã tiêu thụ, thời gian tập, nhịp tim) được gửi về phía người dùng.
14. Giao diện gửi yêu cầu để lấy thực đơn từ API.
15. API trả về thực đơn phù hợp với người dùng dựa trên các thông tin đã nhập.
16. Giao diện nhận và hiển thị thực đơn cho người dùng, bao gồm các món ăn và lượng calo của từng món.



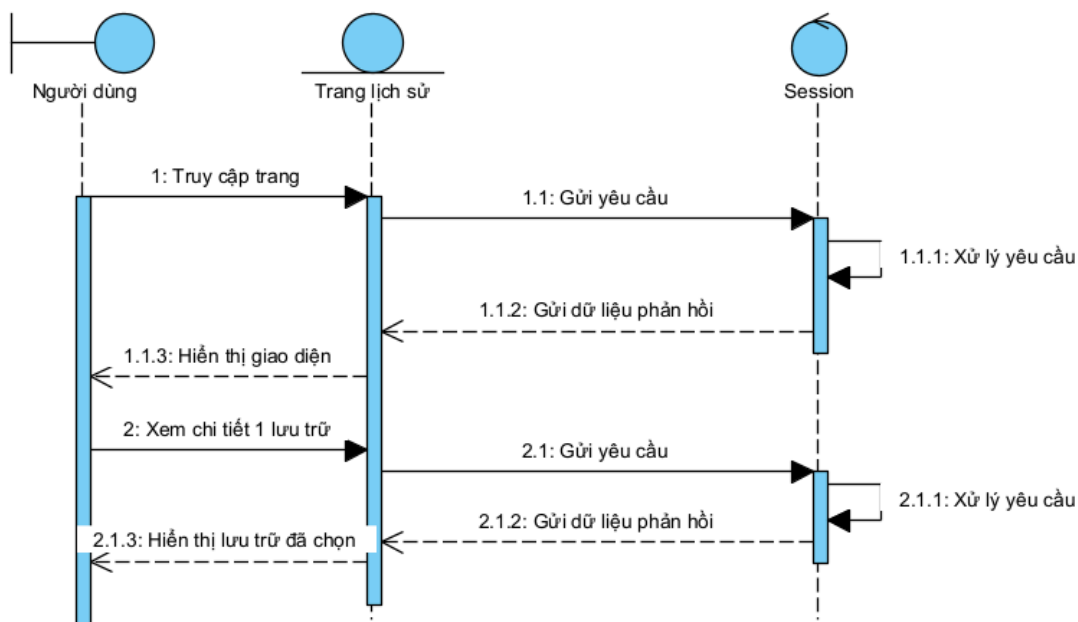
Biểu đồ tuần tự tính năng lượng tiêu thụ

2.4.2 Xem lịch sử

- Thuyết minh:

1. Người dùng truy cập vào trang lịch sử.
2. Trang lịch sử gửi yêu cầu đến Session để lấy dữ liệu liên quan đến lịch sử.
3. Session xử lý yêu cầu) và gửi lại dữ liệu.
4. Trang lịch sử hiển thị giao diện có dữ liệu.
5. Người dùng chọn xem chi tiết một mục đã lưu.
6. Trang lịch sử gửi yêu cầu chi tiết đến.
7. Session Session tiếp nhận và xử lý , sau đó gửi lại dữ liệu.
8. Trang lịch sử hiển thị thông tin chi tiết cho người dùng.

Người dùng		Trang lịch sử		Session	
------------	--	---------------	--	---------	--



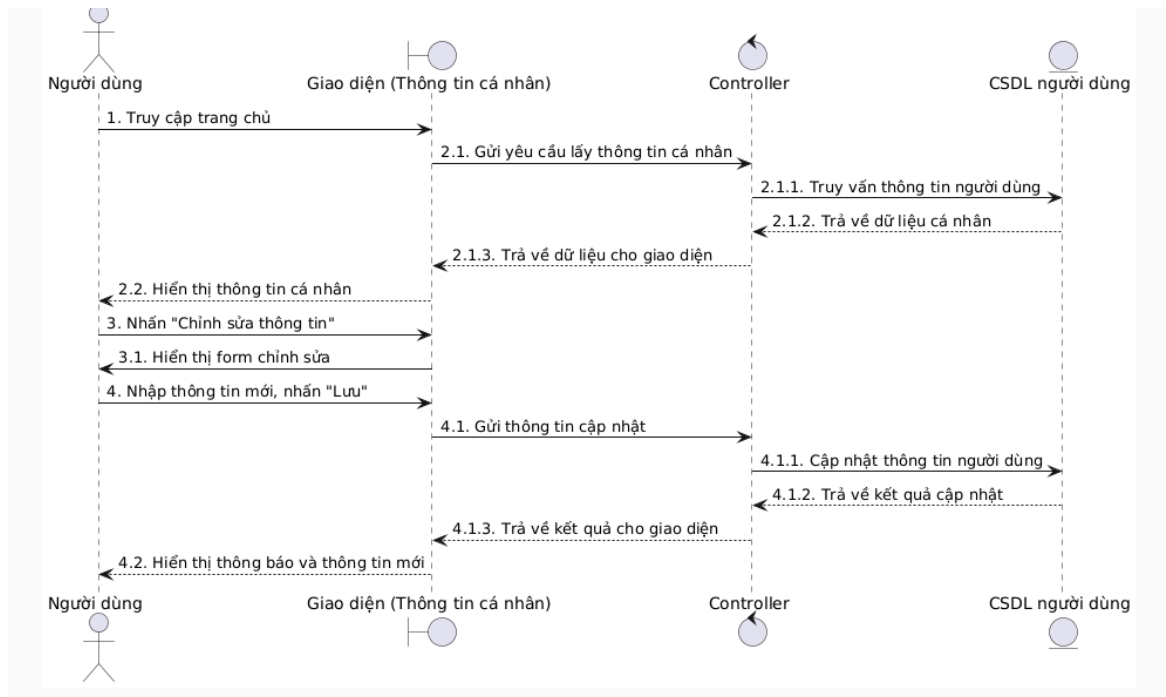
Biểu đồ tuần tự xem lịch sử

2.4.3 Quản lý thông tin cá nhân

- Thuyết minh:

1. Người dùng truy cập trang web và yêu cầu giao diện hiển thị thông tin cá nhân.
2. Giao diện lấy user_id từ session để xác định người dùng hiện tại.
3. Giao diện gửi yêu cầu truy vấn thông tin cá nhân lên controller.
4. Controller nhận user_id và truy vấn cơ sở dữ liệu để lấy thông tin cá nhân (họ tên, chiều cao, cân nặng, mục tiêu, v.v.).
5. Cơ sở dữ liệu trả về thông tin cá nhân cho controller.
6. Controller trả dữ liệu cá nhân về cho giao diện.
7. Giao diện hiển thị thông tin cá nhân của người dùng.
8. Khi người dùng muốn thay đổi thông tin, nhấn nút "Chỉnh sửa thông tin".
9. Giao diện hiển thị form cho phép người dùng nhập thông tin mới (họ tên, chiều cao, cân nặng, mục tiêu, v.v.).
10. Người dùng nhập thông tin mới và nhấn "Lưu".
11. Giao diện gửi dữ liệu cập nhật lên controller.

12. Controller kiểm tra tính hợp lệ và cập nhật thông tin vào cơ sở dữ liệu.
13. Sau khi cập nhật thành công, controller trả về kết quả cho giao diện.
14. Giao diện hiển thị thông báo thành công và cập nhật lại thông tin cá nhân mới cho người dùng.



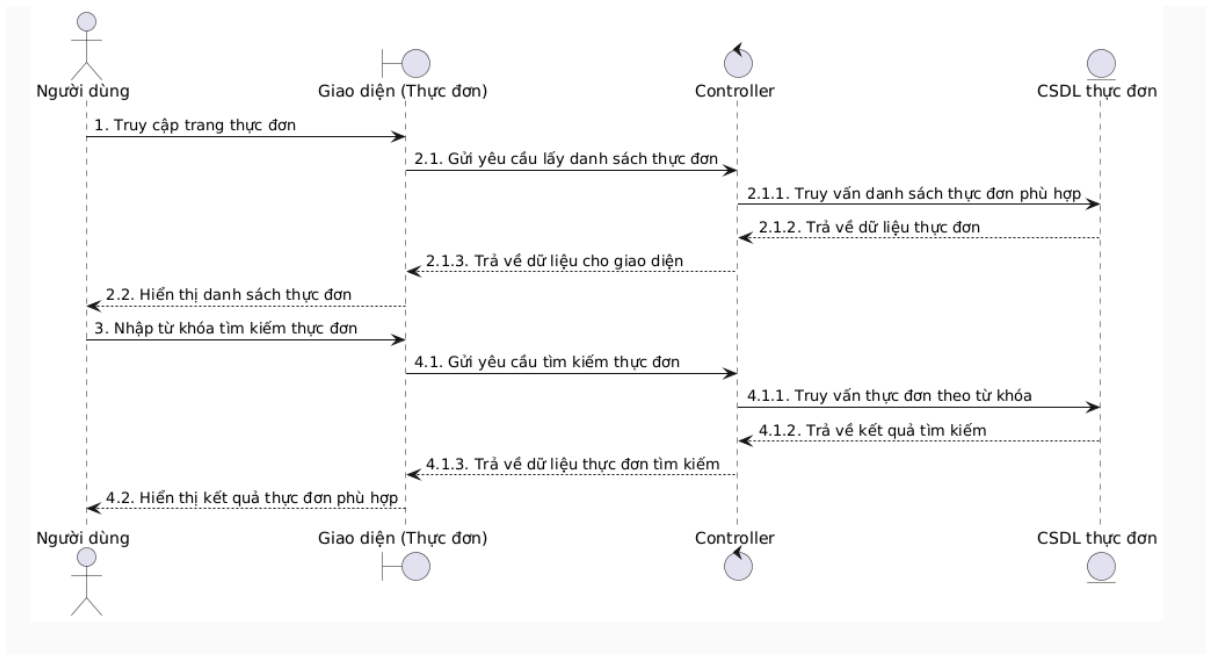
Biểu đồ tuần tự quản lý thông tin cá nhân

2.4.4 Quản lý thực đơn

- Thuyết minh

1. Người dùng truy cập trang web và chọn chức năng quản lý thực đơn.
2. Giao diện gửi yêu cầu lấy danh sách thực đơn lên controller.
3. Controller truy vấn cơ sở dữ liệu để lấy các thực đơn phù hợp với người dùng.
4. Cơ sở dữ liệu trả về danh sách thực đơn cho controller.
5. Controller gửi dữ liệu thực đơn về giao diện.
6. Giao diện hiển thị danh sách thực đơn cho người dùng.
7. Người dùng có thể nhập từ khóa để tìm kiếm thực đơn phù hợp mục tiêu.
8. Giao diện gửi yêu cầu tìm kiếm thực đơn lên controller.
9. Controller truy vấn cơ sở dữ liệu với điều kiện tìm kiếm.

10. Cơ sở dữ liệu trả về kết quả tìm kiếm cho controller.
11. Controller gửi danh sách thực đơn tìm kiếm về giao diện.
12. Giao diện hiển thị kết quả thực đơn phù hợp cho người dùng.



Biểu đồ tuần tự quản lý thực đơn

3. Các thành phần hệ thống

3.1 Front-end

- Mục tiêu giao diện: Giao diện người dùng (UI) cần phải dễ dàng sử dụng, trực quan và cung cấp trải nghiệm mượt mà cho người dùng. Các tính năng chính bao gồm:
 - Biểu đồ tương tác: Sử dụng Chart.js để hiển thị các biểu đồ về lượng calo tiêu thụ, tỷ lệ dinh dưỡng, và tiến trình mục tiêu của người dùng.
 - Dự đoán calo tiêu thụ: Người dùng có thể nhập thông tin cá nhân (tuổi, chiều cao, cân nặng, nhịp tim, nhiệt độ cơ thể) và nhận dự đoán về số calo họ tiêu thụ trong một buổi tập.
- Đề xuất thực đơn: Dựa trên mục tiêu sức khỏe của người dùng (giảm cân, tăng cân, giữ cân), hệ thống sẽ gợi ý thực đơn với các giá trị dinh dưỡng được tính toán rõ ràng (calo, protein, chất béo, carb).

- Lịch sử bài tập và theo dõi: Người dùng có thể theo dõi lịch sử các bài tập, xem các số liệu về mục tiêu đã đạt được, số calo đã đốt, và sự tiến bộ theo thời gian.
- Các công cụ frontend sử dụng như HTML, CSS, JS, và Chart.js cho biểu đồ tương tác.
 - HTML: Cung cấp cấu trúc cơ bản cho các trang web, đảm bảo các phần tử của trang (form nhập liệu, nút bấm, bảng biểu đồ) được hiển thị đúng đắn.
 - CSS: Thiết kế giao diện với các lớp màu sắc, kiểu chữ, và bố cục. Sử dụng CSS Flexbox và Grid để tạo bố cục đáp ứng, giúp giao diện hiển thị tốt trên mọi thiết bị.
 - JavaScript (JS): JavaScript sẽ đảm nhận các tính năng động của giao diện như gửi dữ liệu đến API backend và nhận kết quả phản hồi mà không cần phải tải lại trang. Sử dụng JavaScript thuần hoặc jQuery để xử lý các sự kiện và AJAX request.
 - Chart.js: Một thư viện JavaScript mạnh mẽ để vẽ các biểu đồ trực quan và tương tác. Biểu đồ sẽ được sử dụng để hiển thị các số liệu về lượng calo, phân tích dinh dưỡng, và mục tiêu luyện tập.

3.2 Back-end

- PHP:
 - Xử lý dữ liệu và kết nối cơ sở dữ liệu: PHP sẽ kết nối với cơ sở dữ liệu MySQL, nơi lưu trữ thông tin người dùng, lịch sử tập luyện và các thực đơn gợi ý.
 - Quản lý người dùng và đăng nhập: PHP chịu trách nhiệm về các tác vụ liên quan đến đăng ký, đăng nhập, và bảo mật người dùng.
 - Xử lý logic nghiệp vụ: PHP xử lý các phép toán đơn giản như tính toán calo tiêu thụ và lọc thực đơn, sau đó trả về kết quả cho người dùng.
- Python:
 - Xử lý AI và phép toán tính toán: Python đảm nhiệm các phép toán học phức tạp liên quan đến mô hình học máy. Các phép toán này bao gồm dự đoán calo tiêu thụ, phân tích dinh dưỡng, và gợi ý thực đơn.
 - Machine Learning: Python sử dụng mô hình đã huấn luyện (lưu trữ dưới dạng file pipeline.pkl) để thực hiện các dự đoán về calo và giúp đề xuất các thực đơn.

- API Restful: Các API RESTful sẽ đảm nhận việc kết nối giữa frontend và backend, cũng như giữa backend và mô hình AI. Các API bao gồm:
 - /predict: Nhận dữ liệu từ frontend (như tuổi, chiều cao, cân nặng, nhịp tim) và sử dụng mô hình AI để dự đoán số calo tiêu thụ.
 - Method: POST
 - Request: Dữ liệu JSON chứa các tham số như tuổi, chiều cao, cân nặng, nhịp tim.
 - Response: Kết quả dự đoán calo tiêu thụ.
 - /recommend_menu: Nhận thông tin về mục tiêu sức khỏe (giảm cân, tăng cân) và lượng calo đã đốt, sau đó gợi ý các thực đơn phù hợp từ cơ sở dữ liệu.
 - Method: POST
 - Request: Dữ liệu JSON chứa mục tiêu và lượng calo đã đốt.
 - Response: Danh sách các thực đơn phù hợp với mục tiêu và lượng calo đã đốt.

3.3 Tích hợp mô hình AI

- Cách AI model được tích hợp vào backend để nhận đầu vào từ người dùng, thực hiện các dự đoán và gửi kết quả trở lại cho frontend.
 - Method: GET
 - Response: Trạng thái hệ thống (healthy hoặc thông báo lỗi).

4. Tích hợp và Triển khai

4.1 Quy trình triển khai

- Hệ thống SmartFit được xây dựng theo mô hình Client-Server, bao gồm ba thành phần chính:

- Backend: Sử dụng PHP và MySQL để xử lý logic nghiệp vụ và lưu trữ dữ liệu.
- AI Model: Xây dựng bằng Python và Flask, phục vụ các chức năng dự đoán và gợi ý.
- Frontend: Phát triển bằng HTML, CSS, JavaScript và Bootstrap để cung cấp giao diện người dùng.

- Quy trình triển khai tổng thể gồm các bước như sau:

1. Triển khai Backend (PHP & MySQL)

- Cài đặt môi trường:
 - Sử dụng XAMPP để thiết lập môi trường phát triển gồm Web Server (Apache) và hệ quản trị cơ sở dữ liệu MySQL trên máy tính cá nhân hoặc máy chủ.
 - Toàn bộ mã nguồn PHP được đặt trong thư mục htdocs của XAMPP, ví dụ: htdocs/Smart_Fitness/.
- Cấu hình cơ sở dữ liệu:
 - Tạo cơ sở dữ liệu có tên smartfit_db trong MySQL.
 - Chạy tập tin setup.sql để khởi tạo các bảng cần thiết như: users, user_profile, workout_history, menus,...
 - Kết nối tới cơ sở dữ liệu được thiết lập trong file cấu hình db_config.php.
- Triển khai API backend:
 - Các file PHP như index.php, controller.php, auth.php, menu.php thực hiện các chức năng: xác thực người dùng, quản lý lịch sử tập luyện, truy xuất thực đơn,...
 - Một số API nội bộ như predict.php đóng vai trò trung gian kết nối giữa backend PHP và AI Service (Flask), thực hiện gửi – nhận dữ liệu dự đoán.

2. Triển khai AI Model (Flask Microservice)

- Cài đặt môi trường Python:
 - Hệ thống yêu cầu Python phiên bản 3.7 trở lên.
 - Các thư viện cần thiết (Flask, scikit-learn, pandas, v.v.) được cài đặt bằng lệnh:
 - `pip install -r requirements.txt`
 - (File requirements.txt được đặt trong thư mục ml_service).
- Triển khai mô hình AI:
 - Đặt mô hình đã huấn luyện (pipeline.pkl) và dữ liệu thực đơn (menu_labeled.csv) vào thư mục ml_service.
 - Khởi động Flask server bằng lệnh: `python app.py`
 - Flask chạy tại địa chỉ `http://localhost:5000` và cung cấp các endpoint chính:
 - `/predict`: Dự đoán lượng calories tiêu thụ.
 - `/recommend_menu`: Gợi ý thực đơn phù hợp.

3. Tích hợp Backend với AI Service

- Giao tiếp PHP – Flask:
 - Khi người dùng gửi dữ liệu qua frontend (ví dụ: thông tin buổi tập), backend PHP sử dụng cURL để gửi HTTP request đến Flask API.
 - Cụ thể, file `api/predict.php` sẽ thực hiện các bước sau:
 - Nhận dữ liệu đầu vào từ người dùng.
 - Gửi request tới endpoint `/predict` của Flask để nhận kết quả dự đoán lượng calories tiêu thụ.
 - Dựa trên kết quả calories vừa nhận, tiếp tục gửi request tới endpoint `/recommend_menu` để lấy thực đơn phù hợp.
- Xử lý dữ liệu trả về:
 - Kết quả trả về từ Flask dưới dạng JSON.
 - Backend PHP sẽ giải mã (decode) JSON này, sau đó sử dụng dữ liệu để hiển thị lên giao diện người dùng hoặc lưu vào cơ sở dữ liệu phục vụ cho thống kê và lịch sử tập luyện.

4. Triển khai Frontend

- Giao diện người dùng:
 - Được xây dựng bằng HTML, CSS (sử dụng Bootstrap) và JavaScript.
 - Các trang chức năng được tổ chức qua các file như:
 - `register.php`, `login.php`: Đăng ký, đăng nhập người dùng.
 - `index.php`: Dashboard hiển thị thông tin tổng quan.
 - `menu.php`: Gợi ý thực đơn.
- Tích hợp với Backend:
 - Frontend gửi dữ liệu tới backend thông qua các biểu mẫu HTML sử dụng phương thức POST/GET hoặc qua AJAX tùy từng tính năng.
 - Backend xử lý và trả kết quả (dự đoán calories, gợi ý thực đơn...), sau đó frontend sẽ hiển thị dữ liệu phù hợp cho người dùng.

4.2 Quy trình kiểm thử

- Kiểm thử chức năng tính toán calories và gợi ý thực đơn
 - Mục tiêu: Đảm bảo hệ thống tính toán lượng calories tiêu hao chính xác dựa trên dữ liệu người dùng nhập và trả về gợi ý thực đơn phù hợp.
 - Thực hiện:
 - Nhập các giá trị hợp lệ cho giới tính, tuổi, chiều cao, cân nặng, thời gian tập, nhịp tim, nhiệt độ cơ thể.
 - Nhấn nút "Tính toán với AI".
 - Kiểm tra kết quả calories trả về và danh sách thực đơn gợi ý.
 - Kết quả mong đợi: Calories được tính toán hợp lý, thực đơn gợi ý đúng mục tiêu (giảm cân, tăng cơ, v.v.), không có lỗi kết nối AI.

Giao diện trang tính toán

- Kiểm thử lưu và hiển thị lịch sử
 - Mục tiêu: Đảm bảo lịch sử tính toán được lưu trữ và hiển thị đúng cho từng người dùng.
 - Thực hiện:


- Thực hiện nhiều lần tính toán với các thông số khác nhau.
- Đăng nhập/đăng xuất để kiểm tra lịch sử cá nhân.
- Đóng/mở lại trình duyệt để kiểm tra lưu trữ session khi chưa đăng nhập.
- Kết quả mong đợi: Lịch sử hiển thị đúng, không mất dữ liệu khi đã đăng nhập, cảnh báo mất lịch sử khi chưa đăng nhập.

🕒 Lịch sử hoạt động									
📄 Xuất dữ liệu									
Thời gian	Tuổi	Chiều cao (cm)	Cân nặng (kg)	Thời gian tập (phút)	Nhịp tim	Nhiệt độ (°C)	Calories	Mục tiêu	Chi tiết
2025-06-07 17:40:49	19	178	90	26	100	36.5	129	tăng_cân	👁
2025-06-07 17:39:51	19	178	90	26	100	36.5	129	giảm_cân	👁
2025-05-29 23:48:03	19	189	78	30	100	36.5	143	giảm_mỡ	👁
2025-05-29 23:48:19	19	189	78	30	100	36.5	143	giảm_cân	👁

Giao diện trang lịch sử

- Kiểm thử cập nhật thông tin cá nhân và prefill form
 - Mục tiêu: Đảm bảo thông tin cá nhân được cập nhật và tự động điền vào form tính toán.
 - Thực hiện:
 - Cập nhật thông tin cá nhân (giới tính, tuổi, chiều cao, cân nặng).
 - Chuyển sang tab tính toán, kiểm tra các trường đã được điền sẵn.
 - Kết quả mong đợi: Thông tin được lưu và prefill chính xác.

Thông tin cá nhân



Họ tên: Nguyễn Đình Anh

Giới tính: Nam

Tuổi: 19

Chiều cao: 178 cm

Cân nặng: 90 kg

Mục tiêu: Giảm cân

✎ [Chỉnh sửa thông tin](#)

Giới tính

Nam

▼

Mục tiêu cá nhân

Chọn mục tiêu

Tuổi

19

Chiều cao (cm)

178

Cân nặng (kg)

90

Thời gian tập (phút)

Nhịp tim (nhịp/phút)

Nhiệt độ cơ thể (°C)

Giao diện thông tin cá nhân

- Kiểm thử giao tiếp với AI backend
 - Mục tiêu: Đảm bảo hệ thống kết nối đúng với Flask API, xử lý lỗi khi backend không phản hồi.
 - Thực hiện:
 - Tắt Flask backend, thử tính toán để kiểm tra thông báo lỗi.
 - Bật lại backend, kiểm tra lại chức năng.
 - Kết quả mong đợi: Hiện thị thông báo lỗi rõ ràng khi backend không hoạt động, hoạt động bình thường khi backend chạy.

Giới tính Nam	Mục tiêu cá nhân Giảm cân
Tuổi 19	Chiều cao (cm) 178
Cân nặng (kg) 90	Thời gian tập (phút) 26
Nhịp tim (nhịp/phút) 106	Nhiệt độ cơ thể (°C) 36,6

Tính toán với AI

Lỗi kết nối AI
Không thể kết nối tới AI backend hoặc dữ liệu không hợp lệ.

Công cụ tính Calories & Gợi ý bữa ăn


Giới tính Nam	Mục tiêu cá nhân Tăng cân
Tuổi 19	Chiều cao (cm) 178
Cân nặng (kg) 90	Thời gian tập (phút) 26
Nhịp tim (nhịp/phút) 106	Nhiệt độ cơ thể (°C) 36,6

Tính toán với AI

Kết quả

151
calories đã đốt cháy

Gợi ý bữa ăn



• Khoai lang luộc (117g) + Trứng luộc (43g)
167,7 kcal

Giao diện trang tính toán

- Kiểm thử hiệu suất
 - Mục tiêu: Đảm bảo hệ thống phản hồi nhanh với lượng dữ liệu lớn.
 - Thực hiện:
 - Nhập liên tục nhiều lần tính toán calories.
 - Kiểm tra tốc độ phản hồi của hệ thống và API.
 - Kết quả mong đợi: Thời gian phản hồi dưới 2 giây, không bị treo hoặc timeout.

IV. Tổng kết dự án

1. Tổng kết

Trong suốt quá trình phát triển dự án này, chúng tôi đã thành công trong việc xây dựng một hệ thống thông minh, tích hợp sâu rộng Trí tuệ Nhân tạo để hỗ trợ người dùng tối ưu hóa kế hoạch tập luyện và dinh dưỡng cá nhân. Từ việc thiết lập kiến trúc hệ thống vững chắc với sự phân tách rõ ràng giữa các thành phần frontend, backend và lõi AI, đến việc triển khai các mô hình học máy hiệu quả và quy trình gợi ý dinh dưỡng minh bạch, chúng tôi đã tạo ra một công cụ có khả năng cung cấp các giá trị thiết thực.

Các mô hình AI đã chứng minh khả năng dự đoán lượng calo tiêu thụ với độ chính xác cao, đặc biệt là XGBoost Regressor với hiệu suất vượt trội. Đồng thời, hệ thống gợi ý thực đơn dựa trên luật đã đảm bảo tính khoa học và minh bạch trong việc đề xuất các bữa ăn phù hợp với mục tiêu dinh dưỡng cá nhân. Qua các biểu đồ Use Case, chúng tôi đã phác họa rõ ràng các tương tác chính của người dùng với hệ thống, bao gồm tính calo, xem lịch sử, và quản lý thông tin cá nhân, tất cả đều được xây dựng trên một giao diện người dùng thân thiện và dễ sử dụng.

Tuy nhiên, tiềm năng phát triển của hệ thống vẫn còn rất lớn. Những đề xuất cải tiến về việc tích hợp Computer Vision cho nhận diện bữa ăn và phân tích động tác tập luyện, phát triển mô hình học máy cho gợi ý thực đơn, cùng với khả năng cá nhân hóa sâu hơn và theo dõi tiến độ động, sẽ là những bước tiếp theo quan trọng để nâng cao hiệu quả và tính ứng dụng của hệ thống.

Chúng tôi tin tưởng rằng, với nền tảng vững chắc đã được xây dựng và lộ trình phát triển rõ ràng, hệ thống này sẽ tiếp tục được hoàn thiện để trở thành một công cụ không thể thiếu, đồng hành cùng mỗi cá nhân trên hành trình chinh phục sức khỏe và lối sống lành mạnh.

2. Kết luận

Hệ thống thông minh ứng dụng Trí tuệ Nhân tạo trong việc hỗ trợ tập luyện và dinh dưỡng mà chúng tôi đã phát triển đã đạt được những thành tựu đáng kể, đáp ứng mục tiêu đề ra ban đầu:

- Thành tựu về dự đoán năng lượng tiêu thụ: Mô hình dự đoán lượng calo tiêu thụ đã thể hiện hiệu suất vượt trội, đặc biệt là với mô hình XGBoost Regressor, đạt chỉ số R^2 lên đến 0.998868 và MAE chỉ 1.498120. Điều này cho thấy khả năng dự đoán chính xác lượng năng lượng đốt cháy của

người dùng dựa trên các yếu tố tập luyện và thể trạng cá nhân. Phân tích tầm quan trọng của đặc trưng cũng đã chỉ ra rõ ràng rằng Duration (Thời gian tập luyện) và Heart_Rate (Nhịp tim) là hai yếu tố có ảnh hưởng mạnh mẽ nhất đến lượng calo tiêu thụ.

- Thành tựu về gợi ý thực đơn dinh dưỡng: Mặc dù sử dụng phương pháp dựa trên luật (rule-based) thay vì mô hình học máy phức tạp, hệ thống đã thành công trong việc phân loại và gợi ý các thực đơn dinh dưỡng phù hợp với các mục tiêu sức khỏe khác nhau như giảm cân, tăng cơ, giữ cân và giảm mỡ. Quy trình xử lý dữ liệu để tính toán tỷ lệ macro dinh dưỡng và gán nhãn mục tiêu đã tạo ra một cơ sở dữ liệu thực đơn có cấu trúc và dễ quản lý, đảm bảo tính minh bạch và khoa học trong các gợi ý.
- Tổng quan hệ thống: Hệ thống đã kết nối thành công hai chức năng chính (dự đoán calo và gợi ý thực đơn) để tạo ra một giải pháp toàn diện và cá nhân hóa. Giao diện người dùng (UI) đã được xây dựng để cung cấp một cách tương tác trực quan và thân thiện, giúp người dùng dễ dàng nhập thông tin và nhận các gợi ý từ hệ thống.

Nhìn chung, các kết quả đạt được sau khi triển khai hệ thống là rất khả quan, chứng minh tính khả thi và tiềm năng ứng dụng của AI trong việc hỗ trợ người dùng đạt được các mục tiêu sức khỏe cá nhân một cách hiệu quả hơn.

3. Đề xuất cải tiến

Để nâng cao hơn nữa hiệu quả, tính năng và trải nghiệm người dùng của hệ thống, chúng tôi đề xuất một số cải tiến và các bước phát triển tiếp theo như sau:

- Tối ưu hóa siêu tham số (Hyperparameter Tuning) cho mô hình dự đoán Calo:

Mặc dù XGBoost đã đạt hiệu suất cao, việc thực hiện tối ưu hóa siêu tham số sâu hơn bằng các kỹ thuật như GridSearchCV hoặc RandomizedSearchCV có thể giúp tinh chỉnh mô hình và có thể đạt được độ chính xác cao hơn nữa.

- Phát triển mô hình học máy cho gợi ý thực đơn:

Thay vì hoàn toàn dựa trên luật, trong các phiên bản tiếp theo, có thể phát triển một mô hình phân loại sử dụng học máy (ví dụ: XGBoost Classifier hoặc Random Forest Classifier cho phân loại) để dự đoán nhãn mục tiêu của thực đơn. Điều này sẽ cho phép hệ thống học hỏi từ các mẫu dữ liệu phức tạp hơn và có thể tạo ra các gợi ý linh hoạt hơn, đặc biệt khi dữ liệu thực đơn trở nên đa dạng hơn.

- Tích hợp chức năng AI dựa trên Thị giác Máy tính (Computer Vision):
 - + Nhận diện bữa ăn và tính toán calo qua ảnh (CNN):
 Phát triển một mô hình Mạng nơ-ron tích chập (CNN) để nhận diện các món ăn trong bữa ăn từ hình ảnh do người dùng cung cấp. Mô hình này sẽ phân tích các thành phần, ước lượng khối lượng và từ đó tính toán lượng calo, protein, chất béo, carbohydrate một cách tự động. Điều này giúp người dùng dễ dàng theo dõi lượng calo nạp vào mà không cần nhập liệu thủ công.
 - + Phân tích động tác tập luyện và tính toán calo tiêu thụ (Computer Vision cho phân tích chuyển động):
 Áp dụng kỹ thuật Computer Vision để phân tích động tác của người dùng trong quá trình tập luyện. Hệ thống có thể nhận diện loại bài tập (ví dụ: squat, chống đẩy, nhảy dây) và đếm số lần lặp lại (reps). Kết hợp với thông tin về cường độ (ví dụ: tốc độ thực hiện động tác) và dữ liệu sinh trắc học của người dùng, mô hình AI có thể ước tính lượng calo tiêu thụ chính xác hơn nhiều so với việc chỉ dựa vào nhịp tim hay thời gian tập luyện. Điều này cũng mở ra khả năng cung cấp phản hồi về kỹ thuật tập luyện, giúp người dùng tập đúng form và tránh chấn thương.
- Tích hợp thêm yếu tố cá nhân hóa:
 - + Sở thích và Dị ứng: Cho phép người dùng nhập các thông tin về sở thích ẩm thực, dị ứng thực phẩm hoặc các yêu cầu ăn kiêng đặc biệt (ví dụ: ăn chay, không gluten) để lọc và cá nhân hóa thực đơn gợi ý.
 - + Bệnh lý nền/Tình trạng sức khỏe đặc biệt: Nếu có thể thu thập dữ liệu an toàn và hợp pháp, hệ thống có thể cân nhắc các tình trạng sức khỏe đặc biệt (ví dụ: tiểu đường, huyết áp cao) để đưa ra khuyến nghị an toàn và phù hợp hơn.
- Xây dựng hệ thống gợi ý động và theo dõi tiến độ:
 - + Lịch sử và xu hướng: Phát triển khả năng lưu trữ lịch sử tập luyện và dinh dưỡng của người dùng. Từ đó, hệ thống có thể phân tích xu hướng, học hỏi từ phản hồi của người dùng và tự động điều chỉnh kế hoạch theo thời gian (ví dụ: tăng dần cường độ tập luyện, điều chỉnh lượng calo mục tiêu).
 - + Phản hồi của người dùng: Tích hợp cơ chế phản hồi đơn giản (ví dụ: đánh giá mức độ hài lòng với thực đơn, mức độ hiệu quả của

buổi tập) để mô hình có thể học hỏi và cải thiện các gợi ý trong tương lai.

- Mở rộng cơ sở dữ liệu thực đơn: Tăng cường số lượng và sự đa dạng của các thực đơn trong cơ sở dữ liệu, đặc biệt là cho các mục tiêu ít phổ biến hơn, để đảm bảo người dùng có nhiều lựa chọn hơn và tránh sự lặp lại.
- Cải thiện giao diện người dùng (UI) và trải nghiệm người dùng (UX): Mặc dù UI hiện tại đã thân thiện, việc tiếp tục cải thiện thiết kế, thêm các tính năng trực quan hóa dữ liệu (ví dụ: biểu đồ tiến độ tập luyện, phân tích dinh dưỡng hàng tuần) sẽ nâng cao trải nghiệm người dùng tổng thể.

Lời cảm ơn

Chúng tôi xin bày tỏ lòng biết ơn sâu sắc và chân thành nhất đến thầy Đặng Hoàng Long, người đã tận tình hướng dẫn, truyền đạt những kiến thức chuyên sâu và cung cấp những lời khuyên vô cùng quý báu trong suốt quá trình chúng tôi thực hiện và hoàn thiện báo cáo dự án này.

Sự chỉ dẫn, định hướng rõ ràng và những góp ý chi tiết của thầy đã giúp chúng tôi vượt qua không ít thách thức, từ giai đoạn lên ý tưởng, thu thập và xử lý dữ liệu, cho đến việc lựa chọn, huấn luyện và đánh giá các mô hình Trí tuệ Nhân tạo phức tạp. Không chỉ là một người hướng dẫn chuyên môn, thầy còn là nguồn động lực lớn, khơi gợi niềm đam mê và tinh thần học hỏi, nghiên cứu trong chúng tôi.

Những kiến thức mà thầy đã trang bị, đặc biệt là trong lĩnh vực AI và các ứng dụng của nó trong thực tiễn, đã là nền tảng vững chắc để chúng tôi có thể xây dựng và phát triển hệ thống này. Chúng tôi nhận thấy rằng những bài học và kinh nghiệm từ dự án này sẽ là hành trang quý giá cho chặng đường học tập và sự nghiệp tương lai.

Một lần nữa, chúng tôi xin chân thành cảm ơn thầy vì tất cả những nỗ lực, sự kiên nhẫn và sự hỗ trợ không ngừng nghỉ.

Hết