BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM

KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN CUỐI KỲ

BỘ MÔN: LẬP TRÌNH R CHO PHÂN TÍCH

MÃ HP: RPAN233577_23_1_01

ĐỀ TÀI: ỨNG DỤNG MACHINE LEARNING TRONG DỰ ĐOÁN LƯỢNG CALO ĐỐT CHÁY BẰNG CÁC PHƯƠNG PHÁP PHÂN TÍCH TRÊN R

GVHD: Trần Trọng Bình

HỌC KỲ I – NĂM HỌC 2023-2024

SVTH:

Nguyễn Trung Đức	**21142261**
Nguyễn Đình Phúc	**21133069**
Hồ Minh Trí	**21133110**
Đinh Đức Nguyên Vũ	**20128171**

TP. Hồ Chí Minh, tháng 12 năm 2023

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Họ và tên sinh viên thực hiện

- Nguyễn Trung Đức - 21142261
- Nguyễn Đình Phúc - 21133069
- Hồ Minh Trí - 21133110
- Đinh Đức Nguyên Vũ (Nhóm trưởng) - 20128171

Bộ Môn: Lập Trình R cho Phân tích

MÃ HP: RPAN233577_23_1_01

Nhận xét của giảng viên:

.....
.....
.....
.....
.....
.....
.....

Tp. HCM, tháng 12 năm 2023
Giảng viên hướng dẫn
(Tên và chữ ký)

Trần Trọng Bình

LỜI CẢM ƠN

Chúng em xin gửi lời biết ơn chân thành nhất đến thầy Trần Trọng Bình - giảng viên bộ môn Lập Trình R cho Phân tích, vì tấm lòng hướng dẫn và sự định hình tận tâm mà thầy đã dành cho chúng em trong suốt quá trình thực hiện đồ án cuối kỳ môn Lập Trình R cho Phân tích.

Đồ án này là thành quả của một quá trình học tập và làm việc nhóm đầy khó khăn. Để hoàn thành dự án này, chúng em đã nhận được sự hỗ trợ đặc biệt quan trọng từ thầy. Thầy không chỉ là người chỉ dẫn kiến thức chuyên sâu mà còn là nguồn động viên lớn, giúp chúng em vượt qua những thách thức trong quá trình nghiên cứu và phát triển dự án. Sự tận tâm và kiên nhẫn của thầy đã tạo ra một môi trường học tập tích cực, giúp chúng em hiểu rõ hơn về kiến thức quan trọng và phát triển kỹ năng làm việc nhóm cũng như giải quyết vấn đề.

Chúng em xin bày tỏ lòng biết ơn sâu sắc đối với những lời hướng dẫn, góp ý, cũng như sự hỗ trợ nhiệt tình của thầy trong suốt thời gian qua. Thực sự, nếu không có sự hỗ trợ từ thầy, chúng em không thể hoàn thành dự án một cách xuất sắc như hiện nay.

Bên cạnh đó, chúng em cũng xin chân thành cảm ơn trường Đại học Sư phạm Kỹ Thuật TP. Hồ Chí Minh đã tạo điều kiện môi trường học tập cũng như nguồn kiến thức phong phú để hoàn thành đồ án này một cách tốt nhất.

Một lần nữa, chúng em xin chân thành cảm ơn thầy Trần Trọng Bình vì những đóng góp quý báu đã làm phong phú kiến thức và kinh nghiệm của chúng em.

Trân trọng,

Nhóm 16

BẢNG PHÂN CÔNG CÔNG VIỆC

Tên	MSSV	Nhiệm vụ		Đánh giá
Nguyễn Trung Đức	21142261	Tìm hiểu về đề tài, ứng dụng của nó trong thực tiễn	Tìm hiểu, xây dựng mô hình Random Forest	100%
Nguyễn Đình Phúc	21133069		Tiền xử lý dữ liệu	100%
Hồ Minh Trí	21133110		Tìm hiểu, xây dựng mô hình Hồi quy tuyến tính	100%
Đinh Đức Nguyên Vũ	20128171		Tìm hiểu, xây dựng mô hình Decision Tree	100%

DANH MỤC HÌNH ẢNH

MỤC LỤC

PHẦN MỞ ĐẦU

1. Lý do chọn đề tài.

Trong thời đại hiện đại, khi công nghệ đang bước vào một giai đoạn phát triển mạnh mẽ, việc quan tâm đến vẻ đẹp và sức khỏe của bản thân cũng như gia đình trở nên ngày càng quan trọng. Tuy nhiên, áp lực cuộc sống và thiếu thời gian khiến nhiều người dễ dàng rơi vào thói quen ăn uống không lành mạnh, với đồ ăn nhanh và đồ ăn vặt chiếm lĩnh thế thượng phong.

Nhược điểm của thói quen này không chỉ làm tăng lượng calo hấp thụ mà còn góp phần vào vấn đề béo phì ngày càng gia tăng. Trong bối cảnh này, việc duy trì sự cân bằng giữa lượng calo tiêu thụ và calo hấp thụ trở nên quan trọng hơn bao giờ hết.

Lúc này, Machine Learning đóng vai trò như một công cụ có khả năng xử lý và phân tích dữ liệu đầu vào phức tạp bao gồm yếu tố ảnh hưởng đến việc đốt cháy calo hàng ngày không nằm trong tầm kiểm soát của con người để tính toán lượng calories đốt cháy. Sử dụng thuật toán trên ngôn ngữ lập trình R, chúng tôi có thể hiệu quả hóa quá trình xử lý và phân tích các yếu tố ảnh hưởng đến việc đốt cháy calo, như tuổi tác, giới tính, cường độ hoạt động, và kích thước cơ thể. Ứng dụng Machine Learning với thuật toán hồi quy vào việc dự đoán lượng calories bị đốt cháy của cơ thể con người mang lại tính cấp thiết cao, có khả năng thuận tiện và tối ưu chi phí hơn việc ước lượng truyền thống. Khi ứng dụng thành công lâu dài có thể mở rộng mô hình dự đoán trong nhiều lĩnh vực khác nhau như thiết kế chương trình tập luyện và ăn kiêng, đánh giá tình trạng sức khỏe và phát triển thiết bị theo dõi sức khỏe thông minh.

Ứng dụng này không chỉ giúp theo dõi hiệu quả của quá trình tập luyện và chế độ dinh dưỡng mà còn mở ra cánh cửa cho việc phát triển các ứng dụng thông minh hơn trong lĩnh vực thiết kế chương trình tập luyện, đánh giá tình trạng sức khỏe và phát triển thiết bị theo dõi sức khỏe.

Dự án "*Ứng dụng Machine Learning trong việc dự đoán lượng calo đốt cháy bằng các phương pháp phân tích trên R*" không chỉ đặt ra mục tiêu giảm cân mà còn mong muốn thúc đẩy nhận thức về sức khỏe và tạo động lực cho việc duy trì một lối sống lành mạnh. Chúng tôi hy vọng rằng sự kết hợp giữa công nghệ và sức khỏe có thể mang lại lợi ích to lớn cho mọi người trong thời đại ngày nay.

2. Mục tiêu và nội dung nghiên cứu.

2.1. Mục tiêu.

Nhóm chúng tôi đặt mục tiêu xây dựng một hệ thống có khả năng dự đoán lượng calo được đốt cháy trong quá trình hoạt động thường ngày và tập luyện của cơ thể con người. Mục đích chính là hỗ trợ người dùng trong việc theo dõi và quản lý hiệu quả lượng calo tiêu thụ từ hoạt động hàng ngày và quá trình tập luyện của họ.

2.2. Nội dung nghiên cứu.

- Tìm hiểu và xác định đề tài nghiên cứu, khám phá các tài liệu nghiên cứu khoa học và báo cáo có liên quan đến đề tài đã chọn. Hiểu rõ ý nghĩa của đề tài và tìm hiểu về các phương pháp thực hiện đã được áp dụng trong nghiên cứu trước.
- Đánh giá ý nghĩa của đề tài và xem xét các ưu và nhược điểm của các phương pháp đã được đề xuất trong các báo cáo liên quan. Dựa trên cơ sở lý thuyết, đề xuất một phương pháp thực hiện hiệu quả để có thể áp dụng trong dự án.
- Tiến hành tìm kiếm tập dữ liệu từ các nguồn dữ liệu khác nhau. Thực hiện quá trình huấn luyện mô hình bằng ba thuật toán khác nhau: Random Forest, Decision Tree và Hồi quy tuyến tính.
- Tổng hợp kết quả và đưa ra nhận xét, cùng với đề xuất hướng phát triển tiếp theo cho đề tài.

PHẦN NỘI DUNG

CHƯƠNG 1: GIỚI THIỆU CHUNG VỀ CÁC THUẬT TOÁN.

1.1. Decision Tree.

Thuật toán Decision Tree (Cây Quyết định) là một phương pháp máy học được sử dụng trong việc xây dựng mô hình dự đoán hoặc phân loại dựa trên dữ liệu đào tạo. Thuật toán này tạo ra một cây quyết định bằng cách phân chia tập dữ liệu thành các phần nhỏ hơn và xác định quy luật quyết định tại mỗi nút của cây.

Quyết định tại mỗi nút được thực hiện dựa trên một thuộc tính của dữ liệu, và việc phân chia dữ liệu tiếp tục cho đến khi đạt được điều kiện dừng nào đó. Các điều kiện dừng có thể là kích thước tối thiểu của một nút, số lượng tầng cây, hoặc một điều kiện dừng cụ thể khác.

Ví dụ, giả sử dựa theo thời tiết mà các bạn nam sẽ quyết định đi đá bóng hay không? Những đặc điểm ban đầu là: Thời tiết, Độ ẩm, Gió. Dựa vào những thông tin trên, bạn có thể xây dựng được mô hình như sau:

Thời tiết

├── *Nắng*

│ ├── *Độ ẩm*

│ │ ├── *Bình thường: Đi chơi bóng*

│ │ └── *Cao: Không đi chơi bóng*

│ └── ...

└── ...

Dựa theo mô hình trên, ta thấy: Nếu trời nắng, độ ẩm bình thường thì khả năng các bạn nam đi chơi bóng sẽ cao. Còn nếu trời nắng, độ ẩm cao thì khả năng các bạn nam sẽ không đi chơi bóng.

Cây quyết định có thể được sử dụng cho cả bài toán phân loại và dự đoán. Khi áp dụng cho phân loại, mỗi lá của cây biểu diễn một nhãn hoặc lớp khác nhau. Trong khi đó, khi sử dụng cho dự đoán, mỗi lá biểu diễn một giá trị số liên tục.

Thuật toán Decision Tree được ưa chuộng do tính đơn giản, dễ hiểu và khả năng tạo ra mô hình linh hoạt cho nhiều loại dữ liệu. Tuy nhiên, cũng cần chú ý để tránh tình trạng quá mức phức tạp (overfitting), và có những biến thước quan trọng để điều chỉnh quá trình xây dựng cây.

1.2. Random Forest.

Random Forest là một mô hình máy học được xây dựng trên cơ sở của nhiều cây quyết định (Decision Trees). Điểm độc đáo của Random Forest là sự kết hợp của

nhiều cây quyết định khác nhau để tạo ra một mô hình mạnh mẽ hơn và giảm nguy cơ overfitting (quá mức phức tạp).

Thuật toán Random Forest hoạt động bằng cách chọn ngẫu nhiên một phần của dữ liệu đào tạo và một số thuộc tính ngẫu nhiên để xây dựng từng cây quyết định. Sau đó, khi có dự đoán cần thực hiện, mỗi cây trong Random Forest đưa ra dự đoán của mình, và kết quả cuối cùng được quyết định dựa trên đa số phiếu bầu.

Giả sử: Bạn có một công ty bán hàng trực tuyến và bạn muốn dự đoán khách hàng nào sẽ mua hàng trong tháng tới dựa trên các thông tin như: số lượng mua sắm trong tháng trước, thời gian truy cập trang web, số lượng sản phẩm đã xem, v.v.

Ta có thể sử dụng thuật toán RandomForest để giải quyết vấn đề này như sau:

- Chọn ngẫu nhiên một số lượng nhỏ khách hàng từ danh sách của bạn (ví dụ: 1000 khách hàng).
- Chọn ngẫu nhiên một số lượng nhỏ các thông tin (ví dụ: 3 thông tin là số lượng mua sắm trong tháng trước, thời gian truy cập trang web, số lượng sản phẩm đã xem).
- Xây dựng một cây quyết định dựa trên 1000 khách hàng và 3 thông tin đã chọn.
- Lặp lại các bước 1-3 nhiều lần (ví dụ: 100 lần) để tạo ra 100 cây quyết định khác nhau.
- Khi bạn muốn dự đoán một khách hàng cụ thể sẽ mua hàng trong tháng tới hay không, hãy đưa thông tin của khách hàng đó vào mỗi cây quyết định. Mỗi cây quyết định sẽ đưa ra một dự đoán riêng, và dự đoán cuối cùng sẽ được quyết định bằng cách lấy số phiếu bầu nhiều nhất từ tất cả các cây.

Với cách làm này, thuật toán RandomForest giúp bạn tận dụng được sức mạnh của nhiều cây quyết định, giảm thiểu hiện tượng overfitting (quá khớp), và tăng độ chính xác của dự đoán.

Random Forest có những ưu điểm như khả năng xử lý cả dữ liệu lớn, khả năng ổn định mô hình và giảm thiểu nguy cơ overfitting. Nó cũng có khả năng xác định độ quan trọng của các thuộc tính trong quá trình dự đoán, giúp hiểu rõ hơn về tác động của các yếu tố vào kết quả.

Tính linh hoạt và hiệu suất của Random Forest đã làm cho nó trở thành một trong những lựa chọn phổ biến trong nhiều bài toán máy học, đặc biệt là khi có nhiều biến giải thích và dữ liệu có độ phức tạp cao.

1.3. Hồi quy tuyến tính.

Hồi quy tuyến tính (Linear Regression) là một kỹ thuật phân tích dữ liệu được sử dụng để dự đoán giá trị của một biến dựa trên giá trị của một biến khác1. Có hai loại biến trong hồi quy tuyến tính:

Biến phụ thuộc (Dependent Variable): Biến mà chúng ta muốn dự đoán.

Biến độc lập (Independent Variable): Biến được sử dụng để dự đoán giá trị của biến khác.

Ví dụ, giả sử bạn có dữ liệu về chi phí và thu nhập của bạn trong năm ngoái. Kỹ thuật hồi quy tuyến tính phân tích dữ liệu này và xác định rằng chi phí của bạn là một nửa thu nhập của bạn.

Hồi quy tuyến tính hoạt động như thế nào? Về bản chất, một kỹ thuật hồi quy tuyến tính đơn giản cố gắng vẽ một đồ thị đường giữa hai biến dữ liệu, x và y.

Các bước trong hồi quy tuyến tính

Để có cái nhìn tổng quan, hãy xem xét dạng đơn giản nhất của phương trình đồ thị đường giữa y và x; y=c*x+m, trong đó c và m là hằng số cho tất cả các giá trị có thể có của x và y. Vì vậy, chẳng hạn giả sử rằng tập dữ liệu đầu vào cho (x,y) là (1,5), (2,8), và (3,11). Để xác định phương pháp hồi quy tuyến tính, bạn sẽ thực hiện các bước sau:

- Vẽ một đường thẳng và đo lường mối tương quan giữa 1 và 5.
- Tiếp tục thay đổi hướng của đường thẳng cho các giá trị mới (2,8) và (3,11) cho đến khi tất cả các giá trị đều phù hợp.
- Xác định phương trình hồi quy tuyến tính là y=3*x+2.

Hồi quy tuyến tính thích hợp cho những bài toán trong đó có mối quan hệ tuyến tính giữa biến độc lập và biến phụ thuộc, và nó thường được sử dụng để dự đoán giá trị của biến phụ thuộc dựa trên giá trị của các biến độc lập. Mặc dù đơn giản, nhưng hồi quy tuyến tính vẫn là một công cụ quan trọng và mạnh mẽ trong nghiên cứu và ứng dụng thống kê và máy học.

1.4. Phương pháp đánh giá hiệu suất của các mô hình.

Để có thể đánh giá được hiệu suất của từng mô hình học máy sử dụng các thuật toán Decision Tree, Random Forest và Hồi quy tuyến tính, ta sử dụng các tham số sau:

- MAE (Mean Absolute Error):
  - + Mô tả: MAE đo lường trung bình giá trị tuyệt đối của sự chênh lệch giữa giá trị dự đoán và giá trị thực tế.
  - + Công thức: $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}|$
- MSE (Mean Squared Error):
  - + Mô tả: MSE là trung bình của bình phương sự chênh lệch giữa giá trị dự đoán và giá trị thực tế.
  - + Công thức: $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2$
- RMSE (Root Mean Squared Error):
  - + Mô tả: RMSE là căn bậc hai của MSE, giúp đưa về cùng đơn vị với giá trị thực tế.
  - + Công thức: $RMSE = \sqrt{RSE}$
- R-Squared (Hệ số xác định):

+ Mô tả: R-Squared đo lường tỷ lệ phương sai giữa giá trị dự đoán và giá trị trung bình của biến phụ thuộc so với tỷ lệ phương sai giữa giá trị thực tế và giá trị trung bình của biến phụ thuộc.
+ Công thức:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}$$

Trong đó:

n là số lượng quan sát.

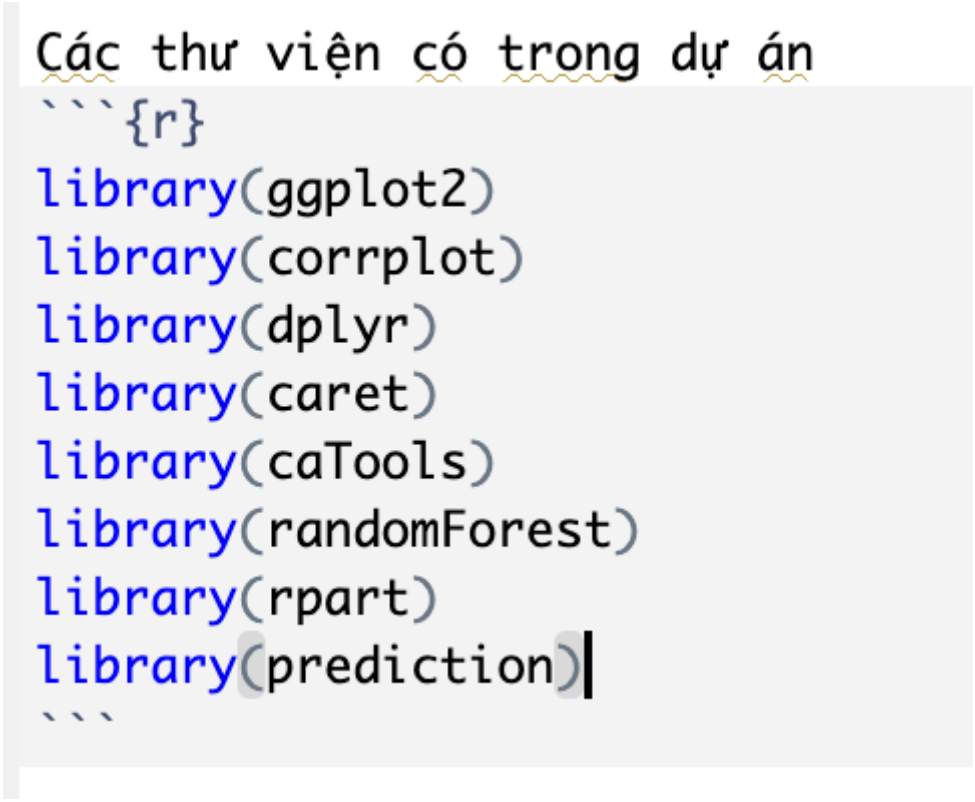
Y_i là giá trị thực tế của quan sát thứ i.

$\hat{Y}_i$ là giá trị dự đoán của quan sát thứ i.

$\bar{Y}$ là giá trị trung bình của biến phụ thuộc.

CHƯƠNG 2: TIỀN XỬ LÝ DỮ LIỆU.

2.1. Chuẩn bị các thư viện.



Hình 1: Các thư viện có trong dự án

Trong đó:

- Thư viện ggplot2: Tạo biểu đồ thống kê và trực quan hóa dữ liệu.
- Thư viện corrplot: Vẽ biểu đồ tương quan như Heatmap, để hiểu thêm về mối quan hệ giữa các biến trong tập dữ liệu
- Thư viên dplyr: Cung cấp các hàm để thực hiện các thao tác xử lý dữ liệu dễ dàng và linh hoạt. Giúp lọc, sắp xếp, tổng hợp, và chuyển đổi dữ liệu một cách hiệu quả.
- Thư viện caTools: Cung cấp các công cụ cho việc chia tách tập dữ liệu thành tập train và tập test. Hỗ trợ quá trình chuẩn bị dữ liệu cho việc đào tạo và đánh giá mô hình.
- Thư viện randomForest: Cung cấp triển khai của thuật toán Random Forest.
- Thư viện rpart: Cung cấp triển khai của thuật toán Decision Tree (Recursive Partitioning).
- Thư viện prediction: dùng để xây dựng mô hình dự đoán Calories.

2.2. Đọc file và in ra các dữ liệu.

In ra các dòng dữ liệu từ CSV

```{r}
path_calories <-"calories.csv"
path_excercise <- "exercise.csv"
data_calories <- read.csv(path_calories)
data_exercise <- read.csv(path_excercise)
print(head(data_calories))
print(head(data_exercise))
```

Hình 2: Đọc 2 file csv calories và exercise

Description: df [6 × 2]

	User_ID <int>	Calories <dbl>
1	14733363	231
2	14861698	66
3	11179863	26
4	16180408	71
5	17771927	35
6	15130815	123

6 rows

Hình 3: Dữ liệu từ file calories.csv

Description: df [6 × 8]

	User_ID <int>	Gender <chr>	Age <int>	Height <dbl>	Weight <dbl>	Duration <dbl>	Heart_Rate <dbl>	Body_Temp <dbl>
1	14733363	male	68	190	94	29	105	40.8
2	14861698	female	20	166	60	14	94	40.3
3	11179863	male	69	179	79	5	88	38.7
4	16180408	female	34	179	71	13	100	40.5
5	17771927	female	27	154	58	10	81	39.8
6	15130815	female	36	151	50	23	96	40.7

6 rows

Hình 4: Dữ liệu từ file exercise.csv

Ở đây, ta có 2 tệp dữ liệu là: data_calories(bao gồm ID người dùng và Calories người đó đã tiêu thụ) và data_exercise(bao gồm các thông tin cụ thể hơn về người đó).

2.3. Gộp 2 tệp dữ liệu thông qua User_ID.

Gộp 2 tệp dữ liệu

```{r}
data_full = merge(data_calories, data_exercise, by = "User_ID")
print(head(data_full))
```

Hình 5: Gộp 2 tập dữ liệu data_calories và data_exercise

Ở đây, ta có thể thông qua cột User_ID để gộp 2 tệp dữ liệu, để có dữ liệu mới data_full như sau

Description: df [6 × 9]

	User_ID <int>	Calories <dbl>	Gender <chr>	Age <int>	Height <dbl>	Weight <dbl>	Duration <dbl>	Heart_Rate <dbl>	Body_Temp <dbl>
1	10001159	76	female	67	176	74	12	103	39.6
2	10001607	93	female	34	178	79	19	96	40.6
3	10005485	49	female	38	178	77	14	82	40.5
4	10005630	36	female	39	169	66	8	90	39.6
5	10006441	122	male	23	169	73	25	102	40.7
6	10006606	130	male	50	183	89	23	96	40.4

6 rows

Hình 6: Tập dữ liệu gộp từ 2 tập

Ở đây, ta có thể thông qua cột User_ID để gộp 2 tệp dữ liệu, để có dữ liệu mới data_full như trên.

2.4. Kiểm tra dữ liệu đầu vào và trực quan hóa dữ liệu.

2.4.1. Kiểm tra dữ liệu.

In số lượng dòng và cột

```{r}
data_size = dim(data_full)
cat("Số dòng:", data_size[1], "\n")
cat("Số cột:", data_size[2], "\n")
str(data_full)
```

```
Số dòng: 15000
Số cột: 9
'data.frame':   15000 obs. of  9 variables:
 $ User_ID   : int  10001159 10001607 10005485 10005630 10006441 10006606 10007368 10007686 10008086 10008486 ...
 $ Calories  : num  76 93 49 36 122 130 65 30 129 55 ...
 $ Gender    : chr  "female" "female" "female" "female" ...
 $ Age       : int  67 34 38 39 23 50 21 47 56 53 ...
 $ Height    : num  176 178 178 169 169 183 185 145 165 157 ...
 $ Weight    : num  74 79 77 66 73 89 80 47 74 65 ...
 $ Duration  : num  12 19 14 8 25 23 12 7 25 10 ...
 $ Heart_Rate: num  103 96 82 90 102 96 103 84 93 97 ...
 $ Body_Temp : num  39.6 40.6 40.5 39.6 40.7 40.4 39.9 39.6 40.8 39.9 ...
```

Hình 7: Kiểm tra số dòng, cột và kiểu dữ liệu từng cột

Kiểm tra số dòng, cột trong tệp dữ liệu và kiểu dữ liệu của từng cột có trong dữ

Kiểm tra dữ liệu đầu vào

```{r}
cat("Số các giá trị trùng lặp: ", sum(duplicated(data_full)), "\n")
cat("Số dòng chứa giá trị null: ", sum(!complete.cases(data_full)), "\n")
summary(data_full)
```

```
Số các giá trị trùng lặp:  0
Số dòng chứa giá trị null:  0
    User_ID            Calories         Gender              Age            Height         Weight         Duration     
 Min.   :10001159   Min.   :  1.00   Length:15000       Min.   :20.00   Min.   :123.0   Min.   : 36.00   Min.   : 1.00  
 1st Qu.:12474191   1st Qu.: 35.00   Class :character   1st Qu.:28.00   1st Qu.:164.0   1st Qu.: 63.00   1st Qu.: 8.00  
 Median :14997285   Median : 79.00   Mode  :character   Median :39.00   Median :175.0   Median : 74.00   Median :16.00  
 Mean   :14977359   Mean   : 89.54                      Mean   :42.79   Mean   :174.5   Mean   : 74.97   Mean   :15.53  
 3rd Qu.:17449279   3rd Qu.:138.00                      3rd Qu.:56.00   3rd Qu.:185.0   3rd Qu.: 87.00   3rd Qu.:23.00  
 Max.   :19999647   Max.   :314.00                      Max.   :79.00   Max.   :222.0   Max.   :132.00   Max.   :30.00  
   Heart_Rate       Body_Temp    
 Min.   : 67.00   Min.   :37.10  
 1st Qu.: 88.00   1st Qu.:39.60  
 Median : 96.00   Median :40.20  
 Mean   : 95.52   Mean   :40.03  
 3rd Qu.:103.00   3rd Qu.:40.60  
 Max.   :128.00   Max.   :41.50  
```

Hình 8: Kiểm tra dòng null, lặp và thuộc tính từng cột

Ta tiến hành kiểm tra các dòng trong dữ liệu, để không có dòng chứa giá trị null và trùng lặp. Khi dữ liệu bị 2 tình trạng trên mà không xử lý thì sẽ ảnh hưởng đến dữ liệu dự đoán sau này.

Thay đổi giá trị Gender

```{r}
# Thay đổi giá trị trong cột 'Gender'
data_full$Gender <- ifelse(data_full$Gender == 'male', 0, 1)
# In ra một số dòng đầu của dataframe
print(head(data_full))
```

Description: df [6 × 9]

	User_ID <int>	Calories <dbl>	Gender <dbl>	Age <int>	Height <dbl>	Weight <dbl>	Duration <dbl>	Heart_Rate <dbl>	Body_Temp <dbl>
1	10001159	76	1	67	176	74	12	103	39.6
2	10001607	93	1	34	178	79	19	96	40.6
3	10005485	49	1	38	178	77	14	82	40.5
4	10005630	36	1	39	169	66	8	90	39.6
5	10006441	122	0	23	169	73	25	102	40.7
6	10006606	130	0	50	183	89	23	96	40.4

6 rows

Hình 9: Thay đổi giá trị trong cột Gender

Tiếp theo, ta xử lý cột Gender có trong dữ liệu, ta thay đổi male = 0, female = 1.

2.4.2. Trực quan hóa dữ liệu bằng các biểu đồ.

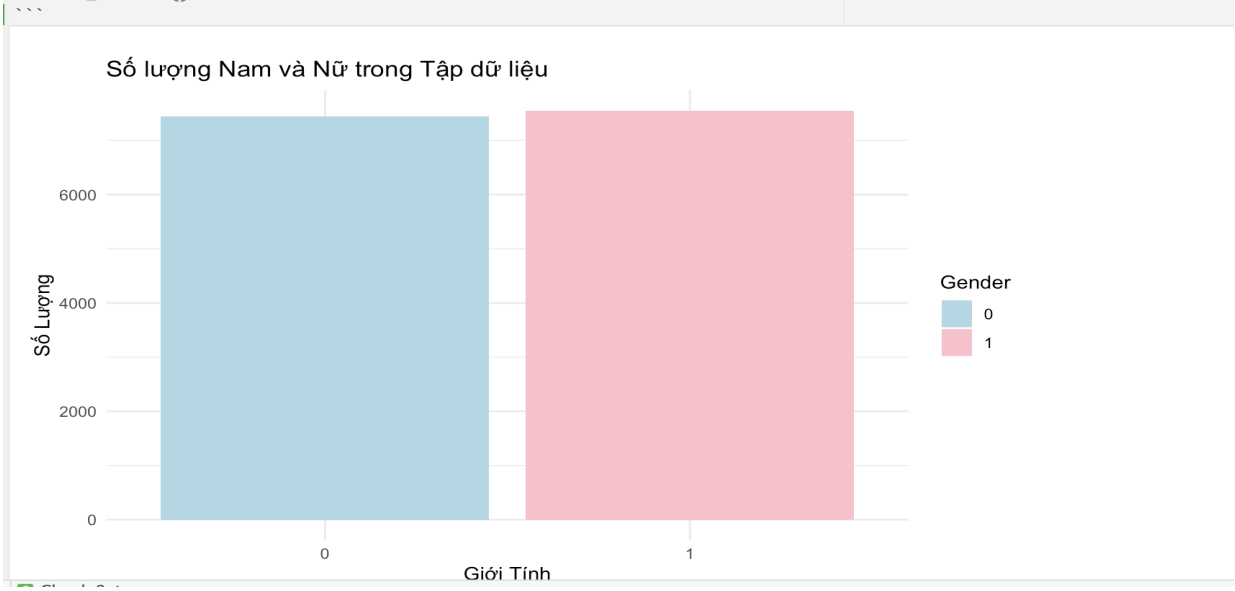
Biểu đồ phân tán Quan hệ giữa Calories và Duration

```{r}
ggplot(data_full, aes(x = Duration, y = Calories,)) +
  geom_point(color = "blue") +
  labs(x = "Duration of exercise in terms of minutes",
       y = "Calories Burned",
       title = "Relationship between the duration of an activity and the calories burned") +
  theme_minimal()
```

Hình 10: Biểu đồ giữa Thời gian tập luyện và Calories tiêu thụ

Biểu đồ này thể hiện mối quan hệ giữa thời gian luyện tập và lượng Calories tiêu thụ.

```
Biểu đồ cột số lượng nam và nữ trong dataset
```{r}
gender_counts <- table(data_full$Gender)
ggplot(data = data.frame(Gender = names(gender_counts), Count = as.numeric(gender_counts)),
       aes(x = Gender, y = Count, fill = Gender)) +
  geom_bar(stat = "identity") +
  labs(x = "Giới Tính",
       y = "Số Lượng",
       title = "Số lượng Nam và Nữ trong Tập dữ liệu") +
  scale_fill_manual(values = c("0" = "lightblue", "1" = "pink")) +
  theme_minimal()
```
```

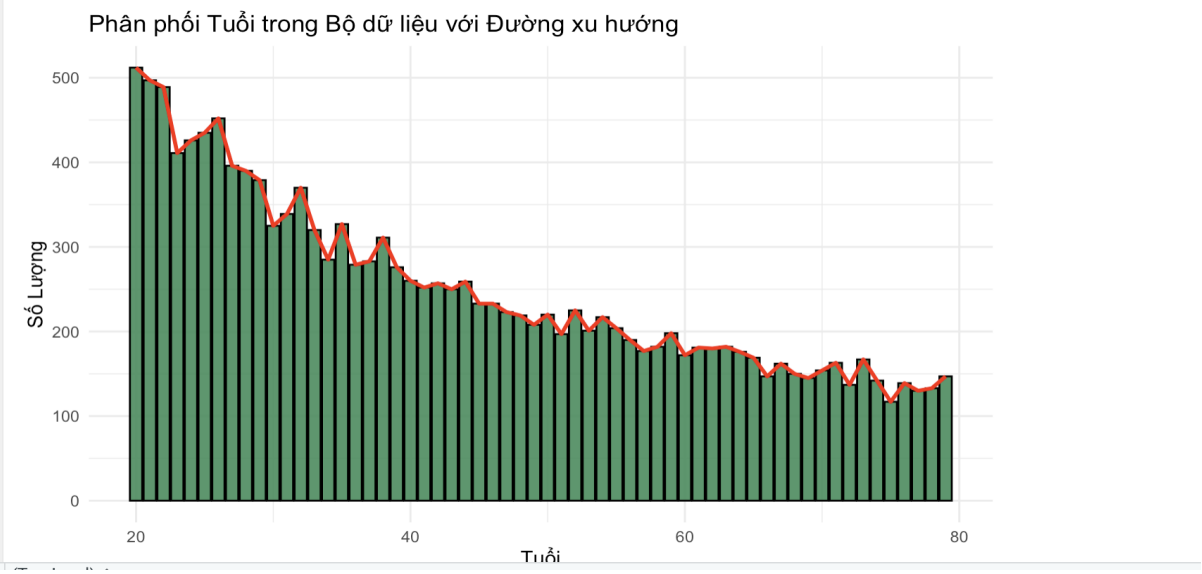


Hình 11: Biểu đồ phân phối số lượng Nam và Nữ trong tập dữ liệu

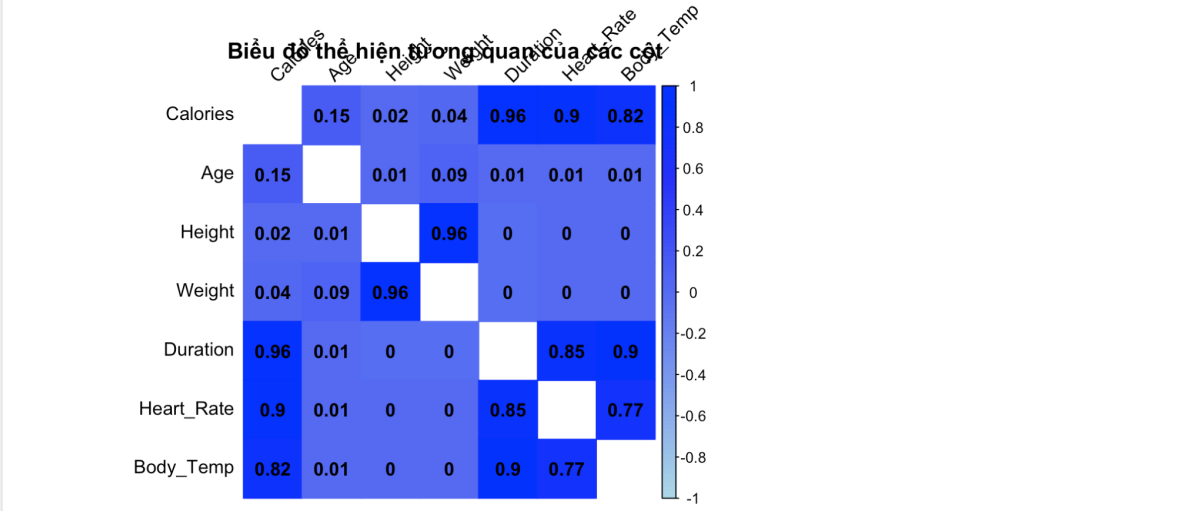
Biểu đồ trên cho thấy số lượng nam và nữ, trong đó 0 là nam và 1 là nữ. Ở đây ta thấy 2 cột giá trị đều cân bằng, chứng tỏ tệp dữ liệu này không bị mất cân bằng dữ liệu

Biểu đồ cột và đường thể hiện độ tuổi trong dataset

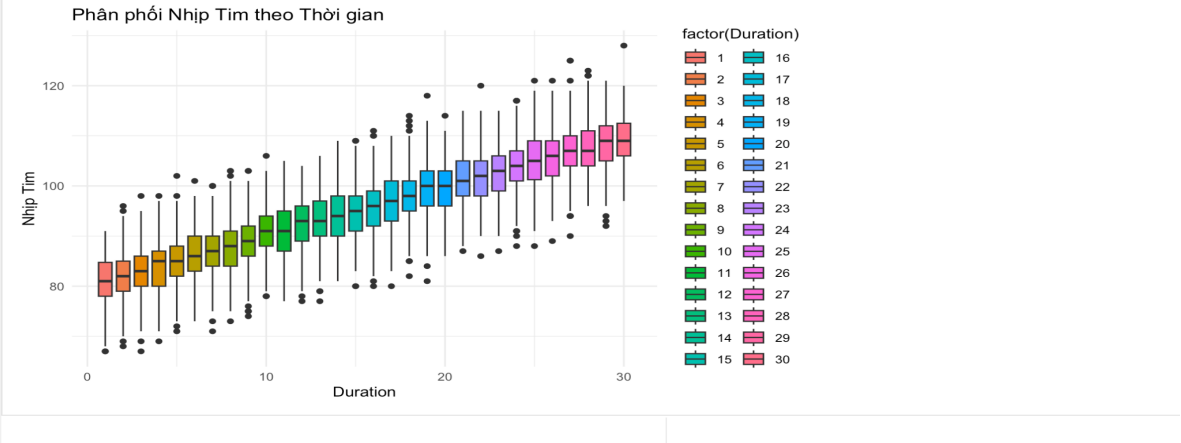
```{r}
age_counts <- table(data_full$Age)
ggplot(data = data.frame(Age = as.numeric(names(age_counts)), Count = as.numeric(age_counts)),
       aes(x = Age, y = Count)) +
  geom_bar(stat = "identity", fill = "seagreen", color = "black", alpha = 0.9) +
  geom_line(stat = "identity", color = "red", size = 1) +
  labs(x = "Tuổi",
       y = "Số Lượng",
       title = "Phân phối Tuổi trong Bộ dữ liệu với Đường xu hướng") +
  theme_minimal()
```



Hình 12: Biểu đồ phân phối độ tuổi trong tập dữ liệu

```
Heat map thể hiện mức độ tương quan giữa các thuộc tính trong dataset
```{r}
# Lấy dữ liệu từ các cột số
numeric_columns <- c("Calories", "Age", "Height", "Weight", "Duration", "Heart_Rate", "Body_Temp")
corr_df <- data_full[numeric_columns]
# Tính ma trận tương quan
correlation_matrix <- cor(corr_df)
# Thiết lập các tùy chọn cho biểu đồ
heatmap_options <- list(annot = TRUE, col = colorRampPalette(c("lightblue", "blue"))(100), center = 0)
# Vẽ biểu đồ tương quan
corrplot(correlation_matrix, method = "color", addCoef.col = "black", tl.col = "black", tl.srt = 45, diag = FALSE, col = heatmap_options$col)
# Đặt tiêu đề cho biểu đồ
title("Biểu đồ thể hiện tương quan của các cột")
```
```



Hình 13: Biểu đồ nhiệt(Headmap) thể hiện mức độ tương quan giữa các thuộc tính.

Ở đây ta có 1 Heatmap, thể hiện các mức độ tương quan giữa các thuộc tính có trong tập dữ liệu theo mức độ từ -1 đến 1, với -1 là tương quan âm(Positive correlation) và 1 là tương quan dương (Negative Correlation).

```
BoxPlot hiển thị mối quan hệ giữa nhịp tim (Heart Rate) và nhiệt độ cơ thể (Body Temperature) dựa trên giá trị "Duration" (thời gian)
```{r}
ggplot(data_full, aes(x = Duration, y = Heart_Rate, fill = factor(Duration))) +
  geom_boxplot() +
  labs(title = 'Phân phối Nhịp Tim theo Thời gian',
       x = 'Duration',
       y = 'Nhịp Tim') +
  theme_minimal()

```
```

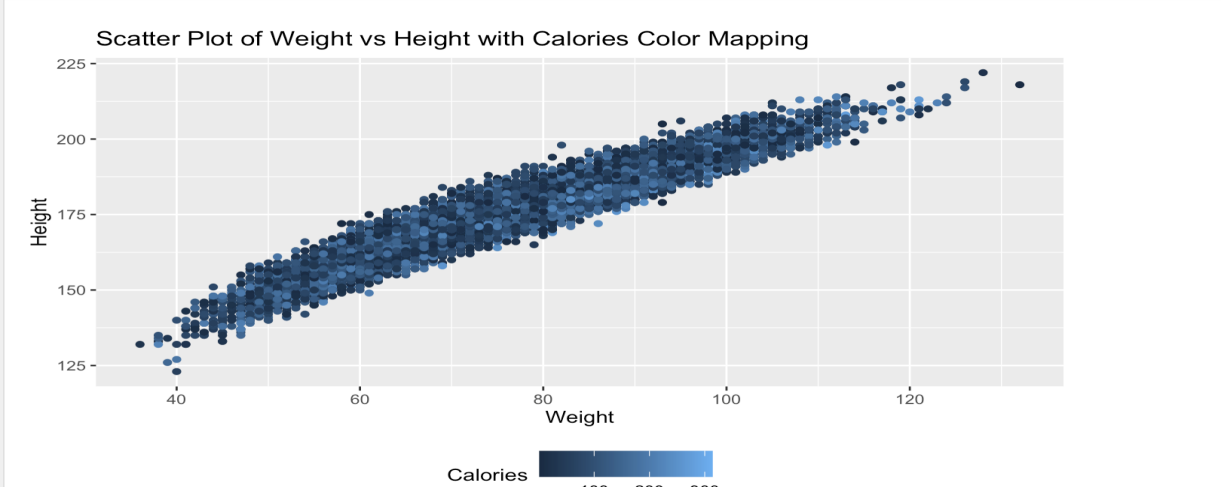


Hình 14: Biểu đồ phân phối nhịp tim theo thời gian luyện tập.

Ở đây, ta dùng Boxplot để thể hiện mối quan hệ giữa nhịp tim và thời gian luyện tập.

ScatterPlot hiển thị cân nặng và chiều cao với Calories

```{r}
# Calculate calorie counts
calorie_counts <- table(data_full$Calories)

# Create labels for the legend
labels <- paste("Calories: ", names(calorie_counts), " (", calorie_counts, ")", sep="")
# Create the scatter plot
ggplot(data_full, aes(x=Weight, y=Height, color=Calories)) +
  geom_point() +
  labs(title="Scatter Plot of Weight vs Height with Calories Color Mapping",
       x="Weight",
       y="Height") +
  theme(legend.position="bottom") +
  scale_color_gradient()
```



Hình 15: Biểu đồ phân tán mối quan hệ giữa cân nặng, chiều cao và Calories

Ta dùng biểu đồ phân tán thể hiện mối quan hệ giữa cân nặng và chiều cao với Calories.

2.5. Phân tách dữ liệu thành các tập.

```{r}
# Tách X và y từ data_full
X <- data_full[, !(names(data_full) %in% c("Calories"))]
y <- data_full$Calories
print(head(X))
print(head(y))
```

data.frame
6 x 8

R Console

Description: df [6 × 8]

	User_ID <int>	Gender <dbl>	Age <int>	Height <dbl>	Weight <dbl>	Duration <dbl>	Heart_Rate <dbl>	Body_Temp <dbl>
1	10001159	1	67	176	74	12	103	39.6
2	10001607	1	34	178	79	19	96	40.6
3	10005485	1	38	178	77	14	82	40.5
4	10005630	1	39	169	66	8	90	39.6
5	10006441	0	23	169	73	25	102	40.7
6	10006606	0	50	183	89	23	96	40.4

6 rows

Hình 16: Tách dữ liệu thành 2 tập X và y

Tách dữ liệu thành 2 tập X và y, trong đó: X gồm tất cả dữ liệu cần thiết cho việc dự đoán Calories, và y là tập dữ liệu chứa Calories.

```r
X <- X[, !(names(X) %in% c("User_ID"))]
print(head(X))
```

Description: df [6 × 7]

	Gender <dbl>	Age <int>	Height <dbl>	Weight <dbl>	Duration <dbl>	Heart_Rate <dbl>	Body_Temp <dbl>
1	1	67	176	74	12	103	39.6
2	1	34	178	79	19	96	40.6
3	1	38	178	77	14	82	40.5
4	1	39	169	66	8	90	39.6
5	0	23	169	73	25	102	40.7
6	0	50	183	89	23	96	40.4

6 rows

Hình 17: Xóa cột User_ID ở tập X

Ta tiến hành drop cột User_ID ở tập X, vì nó không cần thiết cho mô hình học máy.

Chia các dữ liệu thành X_train, X_test, y_train, y_test

```r
# In ra kích thước của X và y
cat("Kích thước của X:", dim(X), "\n")
cat("Kích thước của y:", length(y), "\n")
# Chia tập dữ liệu
set.seed(123)
# Chia dữ liệu thành tập huấn luyện (training_set) và tập kiểm thử (test_set)
split <- sample.split(data_full$Calories, SplitRatio = 0.8)
# Chia dữ liệu X, y thành tập huấn luyện và tập kiểm thử
X_train <- X[split, ]
y_train <- y[split]
X_test <- X[!split, ]
y_test <- y[!split]

# In ra kích thước của các tập dữ liệu
cat("Kích thước tập huấn luyện (X_train):", dim(X_train), "\n")
cat("Kích thước tập kiểm thử (X_test):", dim(X_test), "\n")
cat("Kích thước nhãn tập huấn luyện (y_train):", length(y_train), "\n")
cat("Kích thước nhãn tập kiểm thử (y_test):", length(y_test), "\n")
```

```
Kích thước của X: 15000 7
Kích thước của y: 15000
Kích thước tập huấn luyện (X_train): 12006 7
Kích thước tập kiểm thử (X_test): 2994 7
Kích thước nhãn tập huấn luyện (y_train): 12006
Kích thước nhãn tập kiểm thử (y_test): 2994
```

Hình 18: Chia tập X,y thành X_train, X_test, y_train, y_test

Cuối cùng, ta tiến hành chia các tập dữ liệu ra thành các tập nhỏ hơn bao gồm: X_train, X_test, y_train, y_test để tiến hành thực nghiệm.

CHƯƠNG 3: THỰC NGHIỆM, KẾT QUẢ VÀ THẢO LUẬN.

3.1. Thực nghiệm và kết quả bằng mô hình Decision Tree.

3.1.1. Thực nghiệm.

Decision Tree

```
```{r}
# Huấn luyện mô hình Decision Tree
dt_model <- rpart(y_train ~ ., data = data.frame(y_train, X_train))

# Dự đoán trên dữ liệu kiểm tra
dt_predictions <- predict(dt_model, newdata = data.frame(X_test))

# Tính Các sai số
dt_mae <- MAE(y_test, dt_predictions)
dt_mse <- mean((y_test - dt_predictions)^2)
dt_rmse <- sqrt(dt_mse)
dt_r2 <- 1 - dt_mse / var(y_test)

# In các thông số đánh giá
cat("\nDecision Tree Metrics:\n")
cat("Mean Absolute Error:", dt_mae, "\n")
cat("Mean Squared Error:", dt_mse, "\n")
cat("Root Mean Squared Error:", dt_rmse, "\n")
cat("R2 Score:", dt_r2, "\n")
```
```

Hình 19: Huấn luyện mô hình học máy với Decision Tree

Ở đây, Sử dụng hàm rpart để huấn luyện một mô hình Decision Tree. Biến phụ thuộc y_train là cột đầu tiên trong dataframe, và các biến độc lập X_train là các cột còn lại. Mô hình được lưu vào biến dt_model.

Tiếp theo, sử dụng hàm predict để dự đoán trên dữ liệu kiểm tra (X_test). Kết quả dự đoán được lưu vào biến dt_predictions.

Tiếp theo, tính các sai số đánh giá hiệu suất của mô hình trên dữ liệu kiểm tra:

- Mean Absolute Error (dt_mae).
- Mean Squared Error (dt_mse).
- Root Mean Squared Error (dt_rmse).
- R2 Score (dt_r2).

```r
# In kết quả dự đoán
dt_df <- data.frame(dt_predictions)
result <- data.frame(y_test)
result <- result[order(row.names(result)), ]
y_both <- cbind(dt_df, result)
colnames(y_both) <- c('Decision Tree', 'Thuc Te')
print(y_both)
```

Hình 20: In ra kết quả của mô hình Decision Tree

Ta in ra kết quả dự đoán và so sánh với dữ liệu thực tế có trong tập y_test.

3.1.2. Kết quả.

```
Decision Tree Metrics:
Mean Absolute Error: 14.50349
Mean Squared Error: 378.2745
Root Mean Squared Error: 19.44928
R2 Score: 0.9022665
```

Hình 21: Các thông số đánh giá của mô hình Decision Tree

Decision Tree <dbl>	Thuc Te <dbl>
198.72309	157
198.72309	184
162.64835	193
70.48930	77
14.51471	65
162.64835	23
70.48930	153
14.51471	32
14.51471	169
162.64835	175

Hình 22: Số liệu dự đoán của Decision Tree so với thực tế

Kết quả các thông số đánh giá và số liệu dự đoán của Decision Tree so với thực tế.

3.2. Thực nghiệm và kết quả bằng mô hình Random Forest.

3.2.1. Thực nghiệm.

RANDOM FOREST

```r
# Huấn luyện mô hình Random Forest
rf_model <- randomForest(x = X_train, y = y_train, ntree = 500)

# Dự đoán trên tập kiểm tra
rf_predictions <- predict(rf_model, newdata = X_test)

# Tính Các sai số
rf_mae <- MAE(y_test, rf_predictions)
cat("Mean Absolute Error:", rf_mae, "\n")


rf_mse <- mean((y_test - rf_predictions)^2)
cat("Mean Squared Error:", rf_mse, "\n")

rf_rmse <- sqrt(rf_mse)
cat("Root Mean Squared Error:", rf_rmse, "\n")

rf_r2 <- 1 - sum((y_test - rf_predictions)^2) / sum((y_test - mean(y_test))^2)
cat("R-squared (R2):", rf_r2, "\n")
```

Hình 23: Huấn luyện mô hình học máy với Random Forest

Ở đây, Sử dụng hàm randomForest để huấn luyện một mô hình Random Forest. Biến phụ thuộc y_train là cột đầu tiên trong dataframe, và các biến độc lập X_train là các cột còn lại. Mô hình được lưu vào biến rf_model.

Tiếp theo, sử dụng hàm predict để dự đoán trên dữ liệu kiểm tra (X_test). Kết quả dự đoán được lưu vào biến rf_predictions.

Tiếp theo, tính các sai số đánh giá hiệu suất của mô hình trên dữ liệu kiểm tra:

- Mean Absolute Error (rf_mae).
- Mean Squared Error (rf_mse).
- Root Mean Squared Error (rf_rmse).
- R2 Score (rf_r2).

```{r}
# Create a data frame for predicted and original calories
calories_df <- data.frame(RandomForest = rf_predictions, ThucTe = y_test)
print(calories_df)
```

Hình 24: In ra kết quả của mô hình Random Forest

Ta in ra kết quả dự đoán và so sánh với dữ liệu thực tế có trong tập y_test.

3.2.2. Kết quả.

```
Random Forest Metrics:
Mean Absolute Error: 2.397624
Mean Squared Error: 14.41156
Root Mean Squared Error: 3.796257
R-squared (R2): 0.9962753
```

Hình 25: Các thông số đánh giá của mô hình Random Forest

RandomForest <dbl>	ThucTe <dbl>
156.936695	157
201.907812	202
181.159711	189
52.956931	43
10.774221	10
157.917739	156
101.417693	104
8.361109	8
19.706387	20
186.632397	184

Hình 26: Số liệu dự đoán của Random Forest so với thực tế

Kết quả các thông số đánh giá và số liệu dự đoán của Random Forest so với thực tế.

3.3. Thực nghiệm và kết quả bằng mô hình Hồi quy tuyến tính.

3.3.1. Thực nghiệm.

````
Hồi quy tuyến tính
```{r}
# Huấn luyện mô hình Linear Regression
model <- lm(y_train ~ ., data = cbind(y_train, X_train))

# Dự đoán trên tập kiểm thử
lm_predictions <- predict(model, newdata = data.frame(X_test))

#Tính các sai số
lm_mae <- MAE(y_test, lm_predictions)
lm_mse <- mean((y_test - lm_predictions)^2)
lm_rmse <- sqrt(lm_mse)
lm_r2 <- 1 - lm_mse / var(y_test)

# In thông số đánh giá
cat("\nHồi Quy Tuyến Tính Metrics:\n")
cat("Mean Absolute Error:", lm_mae, "\n")
cat("Mean Squared Error:", lm_mse, "\n")
cat("Root Mean Squared Error:", lm_rmse, "\n")
cat("R2 Score:", lm_r2, "\n")

```
````

Hình 27: Huấn luyện mô hình bằng Hồi quy tuyến tính

Ở đây, Sử dụng hàm lm để huấn luyện một mô hình Hồi quy tuyến tính. Biến phụ thuộc y_train là cột đầu tiên trong dataframe, và các biến độc lập X_train là các cột còn lại. Mô hình được lưu vào biến lm_model.

Tiếp theo, sử dụng hàm predict để dự đoán trên dữ liệu kiểm tra (X_test). Kết quả dự đoán được lưu vào biến lm_predictions.

Tiếp theo, tính các sai số đánh giá hiệu suất của mô hình trên dữ liệu kiểm tra:

- Mean Absolute Error (lm_mae).
- Mean Squared Error (lm_mse).
- Root Mean Squared Error (lm_rmse).
- R2 Score (lm_r2).

```r
# Dự đoán và so sánh với kết quả thực tế
lin_reg_df <- data.frame(lm_preditions, y_test)
colnames(lin_reg_df) <- c('Hoi Quy Tuyen Tinh', 'Thuc Te')
print(lin_reg_df)
```

Hình 28: In ra kết quả mô hình Hồi quy tuyến tính

Ta in ra kết quả dự đoán và so sánh với dữ liệu thực tế có trong tập y_test.

3.3.2. Kết quả.

```
Hồi Quy Tuyến Tính Metrics:
Mean Absolute Error: 8.415269
Mean Squared Error: 126.6106
Root Mean Squared Error: 11.25214
R2 Score: 0.9672881
```

Hình 29: Các thông số đánh giá của Hồi quy tuyến tính

Hoi Quy Tuyen Tinh <dbl>	Thuc Te <dbl>
151.8274947	157
195.1682498	202
199.2472568	189
47.6788088	43
12.7970722	10
152.5842067	156
108.3583396	104
14.2385453	8
30.5024793	20
189.9471148	184

Hình 30: Số liệu dự đoán của Hồi quy tuyến tính so với thực tế

Kết quả các thông số đánh giá và số liệu dự đoán của Hồi quy tuyến tính so với thực tế.

3.4. So sánh kết quả của ba mô hình.

```r
# In kết quả dự đoán
sosanh <- data.frame( DecisionTree = dt_predictions, RandomForest = rf_predictions, HoiQuyTuyenTinh = lm_preditions, ThucTe = y_test)
print(sosanh)
```

Description: df [2,994 × 4]

	DecisionTree <dbl>	RandomForest <dbl>	HoiQuyTuyenTinh <dbl>	ThucTe <dbl>
27	198.72309	157.017744	151.8274947	157
31	198.72309	201.803849	195.1682498	202
32	162.64835	182.620065	199.2472568	189
38	70.48930	52.730599	47.6788088	43
49	14.51471	11.020517	12.7970722	10
51	162.64835	156.784982	152.5842067	156
58	70.48930	102.025788	108.3583396	104
60	14.51471	8.274340	14.2385453	8
61	14.51471	19.202899	30.5024793	20
63	162.64835	186.005694	189.9471148	184

1–10 of 2,994 rows

Previous 1 2 3 4 5 6 ... 100 Next

Hình 31: So sánh kết quả dự đoán của 3 mô hình so với thực tế.

```r
# In các thông số đánh giá cho từng phương pháp
cat("\nCác thông số đánh giá cho từng phương pháp\n")

cat("\nDecision Tree Metrics:\n")
cat("Mean Absolute Error:", dt_mae, "\n")
cat("Mean Squared Error:", dt_mse, "\n")
cat("Root Mean Squared Error:", dt_rmse, "\n")
cat("R2 Score:", dt_r2, "\n")

cat("\nRandomForest Metrics:\n")
cat("Mean Absolute Error:", rf_mae, "\n")
cat("Mean Squared Error:", rf_mse, "\n")
cat("Root Mean Squared Error:", rf_rmse, "\n")
cat("R2 Score:", rf_r2, "\n")

cat("\nHồi Quy Tuyến Tính Metrics:\n")
cat("Mean Absolute Error:", lm_mae, "\n")
cat("Mean Squared Error:", lm_mse, "\n")
cat("Root Mean Squared Error:", lm_rmse, "\n")
cat("R2 Score:", lm_r2, "\n")
```

Hình 32: In ra các thông số đánh giá của từng mô hình

```
Các thông số đánh giá cho từng phương pháp

Decision Tree Metrics:
Mean Absolute Error: 14.50349
Mean Squared Error: 378.2745
Root Mean Squared Error: 19.44928
R2 Score: 0.9022665

RandomForest Metrics:
Mean Absolute Error: 2.397624
Mean Squared Error: 14.41156
Root Mean Squared Error: 3.796257
R2 Score: 0.9962753

Hồi Quy Tuyến Tính Metrics:
Mean Absolute Error: 8.415269
Mean Squared Error: 126.6106
Root Mean Squared Error: 11.25214
R2 Score: 0.9672881
```

Hình 33: Kết quả thông số đánh giá của từng mô hình.

Qua các thông số cũng như số liệu dự đoán so với thực tế, ta có thể thấy trong 3 mô hình học máy.

3.5. Kết luận chung về ba mô hình.

3.5.1. Decision Tree.

Mô hình Decision Tree là một phương pháp học máy mạnh mẽ trong việc phân loại và dự đoán. Nó dựa trên cấu trúc cây quyết định, trong đó mỗi nút trong cây đại diện cho một quy tắc hoặc quyết định dựa trên các đặc trưng của dữ liệu.
Dựa vào các chỉ số của mô hình Decision Tree, ta có thể kết luận rằng mô hình này cho kết quả dự đoán không chính xác các mục tiêu.
Giá trị MAE khá cao (14,50349) cho thấy sai số trung bình khá cao.
Giá trị MSE cao của 378.2745 cho thấy mức độ lỗi trung bình của mô hình là khá cao, và điều này chỉ ra rằng mô hình không dự đoán chính xác các giá trị mục tiêu.
Root Mean Squared Error (RMSE) có giá trị 19.44928, chỉ ra rằng sai số trung bình của mô hình trong việc dự đoán giá trị là khá nhỏ. Điều này cho thấy mô hình có khả năng dự đoán chính xác giá trị mục tiêu.
R2 Score có giá trị 0.9022665, gần với 1, cho thấy mô hình giải thích được phần lớn sự biến thiên của biến mục tiêu. Điều này chứng tỏ mô hình Decision Tree có khả năng giải thích mối quan hệ giữa biến đầu vào và biến mục tiêu.
Tuy nhiên, để cải thiện mô hình Decision Tree và giảm giá trị MSE, có thể xem xét các phương pháp như tối ưu hóa siêu tham số, cắt tỉa cây, sử dụng Ensemble Learning như Random Forest hoặc Gradient Boosting. Điều này có thể giúp tăng độ chính xác và giảm thiểu lỗi của mô hình.
Tóm lại, mô hình Decision Tree hiện tại không cho kết quả dự đoán chính xác các mục tiêu, dù RMSE thấp và R2 Score cao. Cần xem xét các phương pháp cải thiện mô hình và cân nhắc sự phù hợp của nó trong bối cảnh bài toán cụ thể.

3.5.2. Random Forest.

Random Forest là một phương pháp học tập thể (ensemble learning) được sử dụng cho việc phân loại, hồi quy. Phương pháp này hoạt động bằng cách xây dựng một loạt cây quyết định (decision trees) trong quá trình huấn luyện. Khi tổng hợp kết quả, Random Forest dùng phương pháp bỏ phiếu (voting) cho bài toán phân loại và lấy giá trị trung bình (average) cho bài toán hồi quy.

Qua kết quả từ mô hình Random Forest, ta thấy được mô hình này đã khắc phục được vấn đề overfitting mà gặp trong Decision Tree.

Giá trị MAE đã giảm đáng kể so với Decision Tree(2,39 so với 14,5), cho thấy độ hiệu quả của mô hình.

Giá trị MSE (Mean Squared Error) cáo là 14.4156 cho thấy mức độ lỗi trung bình của mô hình là khá thấp. Điều này chỉ ra rằng mô hình dự đoán khá chính xác các giá trị mục tiêu.

RMSE (Root Mean Squared Error) có giá trị là 3.796257, chỉ ra rằng sai số trung bình của mô hình trong việc dự đoán giá trị là khá nhỏ. Nó cung cấp một độ lệch chuẩn của các lỗi, giúp ta hiểu rõ hơn về biến động của các lỗi.

R2 Score có giá trị 0.9962753, gần với 1, cho thấy mô hình giải thích được phần lớn sự biến thiên của biến mục tiêu. Điều này chứng tỏ mô hình Random Forest có khả năng giải thích mối quan hệ giữa biến đầu vào và biến mục tiêu.

Tuy nhiên, để cải thiện mô hình Random Forest và giảm giá trị MSE, có thể xem xét các phương pháp như tối ưu hóa siêu tham số, cắt tỉa cây, sử dụng Ensemble Learning như Gradient Boosting. Điều này có thể giúp tăng độ chính xác và giảm thiểu lỗi của mô hình.

3.5.3. Hồi quy tuyến tính.

Dựa trên bảng số liệu và các chỉ số đánh giá, chúng ta có thể nhận xét như sau:

- Mean Absolute Error(MAE): Đã giảm đáng kể so với Decision Tree, tuy nhiên vẫn khá cao(8,41 so với 14,5 của Decision Tree và 2,39 của Random Forest).
- Mean Squared Error (MSE): Ở đây, MSE = 126.5882 cho thấy mức độ sai số bình phương trung bình giữa giá trị dự đoán và giá trị thực tế là khá cao thì giá trị này có thể là dấu hiệu của một mức độ sai số đáng kể.
- Root Mean Squared Error (RMSE):Với RMSE = 11.25114, là một số dương thể hiện độ lớn trung bình của sai số dự đoán, RMSE càng thấp, mô hình càng chính xác. Trong trường hợp này, giá trị RMSE khá thấp, điều này có vẻ là tích cực về mặt độ chính xác của mô hình.
- R-squared (R2): R2 là một chỉ số thống kê mô tả khả năng giải thích của mô hình đối với biến phụ thuộc.Với R2 = 0.9672829, là một giá trị rất cao, gần với 1. Điều này cho thấy mô hình giải thích được khoảng 96.73% sự biến thiên của biến phụ thuộc, là một dấu hiệu tích cực về khả năng mô hình hóa.

Nhìn chung, mặc dù MSE có giá trị khá lớn, nhưng khi xem xét RMSE và R2, có vẻ như mô hình hồi quy tuyến tính đang có tính chính xác khá tốt trong việc dự đoán giá trị thực tế. Tuy nhiên, cần lưu ý rằng việc đánh giá mô hình không chỉ dựa trên các số liệu này mà còn phụ thuộc vào bối cảnh và yêu cầu cụ thể của ứng dụng hay vấn đề nghiên cứu.

Và để khắc phục thêm về MSE đang có giá trị khá lớn ta có thể dùng thêm các phương pháp như :

- Loại Bỏ Các Quan Sát Ngoại lai (Outliers) kiểm tra xem có sự xuất hiện của các quan sát ngoại lai (outliers) không. Nếu có, xem xét loại bỏ chúng để giảm thiểu ảnh hưởng của chúng.
- Biến Đổi Dữ Liệu: thực hiện các biến đổi dữ liệu như logarit hoặc căn bậc hai để làm giảm độ lớn của giá trị và tối ưu hóa mô hình hóa mối quan hệ.
- Kiểm Soát Overfitting: áp dụng các biện pháp kiểm soát overfitting như giảm số lượng biến độc lập, sử dụng regularization, hoặc thử nghiệm các mô hình đơn giản hơn.
- Kiểm Soát Việc Chọn Biến (Variable Selection): kiểm soát quá trình chọn biến để đảm bảo chỉ những biến quan trọng được giữ lại, từ đó giảm thiểu ảnh hưởng của các biến không quan trọng.

- Thực Hiện Cross-Validation: là một kỹ thuật đánh giá hiệu suất của mô hình máy học trên dữ liệu huấn luyện bằng cách chia dữ liệu thành các tập dữ liệu con (folds) và thực hiện quá trình đào tạo và kiểm thử trên các tập con này. Mục tiêu là đánh giá khả năng tổng quát hóa của mô hình trên dữ liệu mới mà nó chưa từng thấy. Sử dụng kỹ thuật cross-validation để kiểm tra và đánh giá hiệu suất của mô hình trên các tập dữ liệu kiểm thử, đồng thời giảm nguy cơ overfitting và cải thiện RMSE.

Từ đó giúp tối ưu hóa và cải thiện độ chính xác của mô hình hồi quy tuyến tính, từ đó giảm giá trị MAE, MSE và làm cho dự đoán trở nên chính xác hơn. Với những phương pháp trên ta có thể hoàn toàn ứng dụng nó cho cả giá trị RMSE, R2 để tối ưu hóa tính chính xác của mô hình.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.

Quá trình xây dựng và phát triển mô hình dự đoán lượng calo tiêu thụ từ hoạt động vận động là một hành trình công phu, đòi hỏi sự tỉ mỉ và sự chú ý đặc biệt cho từng bước để đạt được mục tiêu ban đầu. Bước đầu tiên là tiền xử lý và kiểm tra tập dữ liệu để đảm bảo tính chính xác và đầy đủ, một cơ sở vững chắc cho quá trình phân tích tiếp theo.

Thông qua việc tính toán các thông số thống kê như MAE, MSE, RMSE và R-squared, chúng ta có cái nhìn tổng quan về phân phối của các biến quan trọng trong dữ liệu. Việc sử dụng biểu đồ giúp hiển thị mức độ phân phối và phát hiện các điểm ngoại lệ, có thể ảnh hưởng đến sự chính xác của mô hình. Cũng qua việc kiểm tra mối quan hệ giữa các biến thông qua biểu đồ trong R, chúng ta có cái nhìn tổng quan về sự tương quan giữa Độ Tuổi, Cân Nặng, Thời Gian Tập Luyện,... và Lượng Calories.

Nghiên cứu này tập trung vào việc nhận biết lượng calo mà cơ thể đốt cháy, dựa vào các yếu tố như tuổi, giới tính, cân nặng, chiều cao, nhiệt độ cơ thể, thời gian và nhịp tim. Quan trọng nhất là hiểu rõ lượng calo chúng ta tiêu thụ và sản xuất để duy trì sức khỏe. Việc tính toán lượng calo bị đốt cháy từ các thuật toán xây dựng trở thành một bài toán toàn diện, mang lại giá trị Mean Absolute Error (MAE) ở thuật toán Random Forest là 2.39, đạt được độ chính xác cao và lỗi thấp.

Tóm lại, không chỉ cung cấp cái nhìn tổng quan về dữ liệu, quá trình này còn kết hợp xây dựng và đánh giá các mô hình để dự đoán calo tiêu thụ từ hoạt động vận động. Kết quả này đặt ra những ước lượng chính xác về nhu cầu calo cho các hoạt động cụ thể, hỗ trợ quyết định về lối sống và dinh dưỡng.

Để phát triển mô hình trong tương lai, đề xuất tăng cường thu thập dữ liệu bằng cách tích hợp và xây dựng ứng dụng dự đoán lượng calo trên các nền tảng di động. Điều này sẽ mang lại trải nghiệm cá nhân hóa và thuận tiện cho người dùng, giúp họ theo dõi hoạt động tập luyện và duy trì chế độ dinh dưỡng hiệu quả. Hơn nữa, có thể tích hợp hệ thống nhắc nhở, ứng dụng thông báo về mục tiêu tập luyện và lời khuyên dinh dưỡng để tạo động lực cho người dùng.

Đề xuất mở rộng dữ liệu đa dạng hơn và sử dụng phương pháp tăng cường dữ liệu để mô phỏng các tình huống đa dạng. Điều này giúp mô hình trở nên linh hoạt và có khả năng xử lý các trường hợp ngoại lệ một cách hiệu quả.

TÀI LIỆU THAM KHẢO.

Nguồn dữ liệu: fmendes-DAT263x-demos, FERNANDO FERNANDEZ

Link: https://www.kaggle.com/datasets/fmendes/fmendesdat263xdemos

[1] Random Forest algorithm, Tuấn Nguyễn,

Link: https://machinelearningcoban.com/tabml_book/ch_model/random_forest.html

[2] Decision Tree algorithm, Tuấn Nguyễn,

Link: https://machinelearningcoban.com/tabml_book/ch_model/decision_tree.html

[3] Hồi quy tuyến tính là gì? Phân loại, Phương trình, Ví dụ và Các giả định

Link: https://meeyland.com/tin-tuc/hoi-quy-tuyen-tinh-la-gi-phan-loai-phuong-trinh-vi-du-va-cac-gia-dinh-378180470

[4] Hồi quy tuyến tính là gì?, Daniel Nelson,

Link: https://www.unite.ai/vi/what-is-linear-regression/

[4] Calorie Pred, MUBASSHIRA QURAISHI
Link: https://www.kaggle.com/code/mubasshiraquraishi/calorie-pred-svr-rf-ada-lin-reg
[5] Đánh giá model trong Machine Learing, Nguyen Toan Thinh,
Link: https://viblo.asia/p/danh-gia-model-trong-machine-learing-RnB5pAq7KPG?fbclid=IwAR1QAQ60fSd3F5SYFpz_bYFwYOU8h-PUGVjgGlcFvhRmCAu8fNKdATfTv3c