TRƯỜNG ĐẠI HỌC MỞ HÀ NỘI

**KHOA CÔNG NGHỆ THÔNG TIN**

----------------------



**BÁO CÁO BÀI TẬP LỚN**

**MÔN: NHẬP MÔN KHAI PHÁ DỮ LIỆU VÀ MÁY HỌC**

**Nhóm 1**

**Đề tài: Chẩn đoán nguy cơ mắc bệnh béo phì sử dụng**

**học máy có giám sát**

**GVHD: TS.Vũ Xuân Hạnh**

**Sinh viên thực hiện:**

**Nguyễn Đức An – 2210A01**

**Dương Việt Hoàng - 2210A06**

**Vũ Văn Phúc - 2210A06**

**Phùng Thế Thăng - 2210A06**

**Hà Nội, 2024**

**MỤC LỤC**

[MỤC LỤC 2](#_Toc181279101)

[DANH MỤC HÌNH ẢNH 2](#_Toc181279102)

[CHƯƠNG 1: KHẢO SÁT NGHIỆP VỤ VÀ THU THẬP DỮ LIỆU 6](#_Toc181279103)

[1.1 Khảo sát nghiệp vụ 6](#_Toc181279104)

[1.1.1. Xác định đối tượng nghiệp vụ 6](#_Toc181279105)

[1.1.2. Tiếp cận và đánh giá tình huống 6](#_Toc181279106)

[1.1.3. Xác định mục tiêu 6](#_Toc181279108)

[1.1.4. Lập kế hoạch dự án 6](#_Toc181279109)

[1.2 Thu thập dữ liệu 7](#_Toc181279110)

[1.2.1. Thu thập dữ liệu ban đầu 7](#_Toc181279111)

[1.2.2. Mô tả dữ liệu 8](#_Toc181279112)

[1.2.3. Khám phá dữ liệu 9](#_Toc181279113)

[CHƯƠNG 2: TIỀN XỬ LÝ DỮ LIỆU VÀ MÔ HÌNH HÓA 11](#_Toc181279114)

[2.1 Tiền xử lý dữ liệu 11](#_Toc181279115)

[2.1.1. Chọn dữ liệu 11](#_Toc181279116)

[2.1.2. Làm sạch dữ liệu 11](#_Toc181279117)

[2.1.3. Xây dựng dữ liệu 11](#_Toc181279118)

[2.1.4. Tích hợp dữ liệu 12](#_Toc181279119)

[2.2 Mô hình hóa 13](#_Toc181279120)

[2.2.1. Lựa chọn kĩ thuật mô hình hóa 13](#_Toc181279121)

[2.2.2. Tạo thiết kế thử nghiệm (Test) 13](#_Toc181279122)

[2.2.3. Xây dựng mô hình 14](#_Toc181279123)

[2.2.4. Mô hình truy cập 15](#_Toc181279124)

[CHƯƠNG 3: ĐÁNH GIÁ MÔ HÌNH VÀ TRIỂN KHAI ỨNG DỤNG 17](#_Toc181279125)

[3.1 Đánh giá mô hình 17](#_Toc181279126)

[3.1.1. Đánh giá kết quả 17](#_Toc181279127)

[3.1.2. Đánh giá quá trình 18](#_Toc181279128)

[3.1.3. Xác định bước tiếp theo 21](#_Toc181279129)

[3.2. Triển khai ứng dụng 22](#_Toc181279130)

[3.2.1.Kế hoạch triển khai 22](#_Toc181279131)

[3.2.2.Lập kế hoạch giám sát và bảo trì 24](#_Toc181279132)

[3.2.3.Lập báo cáo cuối cùng 24](#_Toc181279133)

[3.2.4.Đánh giá dự án 24](#_Toc181279134)

**DANH MỤC HÌNH ẢNH**

[Hình 1‑1. Dataset lấy từ Kaggle 8](#_Toc180959010)

[Hình 1‑2. Dataset lấy từ UC Irvine Machine Learning Repository 9](#_Toc180959011)

[Hình 1‑3. Khám phá Dataset 1 10](#_Toc180959012)

[Hình 1‑4. Khám phá Dataset 2 11](#_Toc180959013)

[Hình 2‑1.Tích hợp dữ liệu 13](#_Toc180959014)

[Hình 2‑2.Phân bố lớp yes và no của nhãn Obesity sau cân bằng 14](#_Toc180959015)

[Hình 2‑3.Test set có 3541 bản ghi 15](#_Toc180959016)

[Hình 3‑1.Ma trận nhầm lẫn 20](#_Toc180959017)

[Hình 3‑2.Đường cong ROC 21](#_Toc180959018)

**PHẦN MỞ ĐẦU**

**Lý do chọn đề tài**

Theo số liệu thống kê năm 2016, trên toàn thế giới có khoảng 800 triệu người đang sống chung với tình trạng béo phì, bao gồm hơn 670 triệu người lớn và ít nhất 120 triệu trẻ em và thanh thiếu niên. Con số này vẫn đang tăng lên. WHO ước tính rằng đến năm 2025, khoảng 167 triệu người nữa – người lớn và trẻ em – sẽ trở nên kém khỏe mạnh hơn vì họ thừa cân hoặc béo phì. [1]

Bệnh béo phì không chỉ gây ra các vấn đề sức khỏe nghiêm trọng như tim mạch, tiểu đường mà còn tạo ra gánh nặng kinh tế lớn cho xã hội.Điều đó đặt ra yêu cầu cần phải chẩn đoán, phát hiện sớm bệnh béo phì giúp kịp thời phòng ngừa, ngăn chặn nguy cơ xảy ra biến chứng gây nguy hiểm cho người bệnh..

**Tổng quan vấn đề nghiên cứu**

Theo Tổ chức Y tế thế giới, thừa cân béo phì nghĩa là tình trạng tích lũy mỡ quá mức và không bình thường tại một vùng cơ thể hay toàn thân gây ra nhiều nguy hại tới sức khỏe. Nhìn chung, bệnh thừa cân béo phì thể hiện trọng lượng cơ thể cao hơn trọng lượng chuẩn ở một người khỏe mạnh. [Bệnh thừa cân béo phì](https://vinmecdr.com/huong-dan-thuc-hien-quy-trinh-chuyen-mon-danh-gia-va-dieu-tri-benh-thua-can-beo-phi/) là bệnh mãn tính do sự dư thừa quá mức lượng mỡ trong cơ thể.

Bệnh thừa cân, béo phì được phân loại bằng [chỉ số khối cơ thể BMI](https://www.vinmec.com/vie/bai-viet/chi-so-bmi-bao-nhieu-la-binh-thuong-vi). Chỉ số khối cơ thể được tính dựa trên chiều cao và trọng lượng cơ thể. Chỉ số khối cơ thể (BMI) sẽ được tính theo công thức: trọng lượng cơ thể của 1 người (tính bằng kg) chia cho bình phương chiều cao (tính bằng mét). Do chỉ số BMI mô tả mối liên quan giữa trọng lượng cơ thể với chiều cao nên sẽ liên quan chặt chẽ đến tổng số lượng mỡ phân bố trong cơ thể ở người trưởng thành.

Theo phân loại của Tổ chức Y tế thế giới, một người trưởng thành, trừ người có thai nếu có chỉ số BMI trong khoảng 25 – 29,9 được xem là thừa cân, và một người trưởng thành có chỉ số BMI 30 được xem là béo phì. Dấu hiệu dễ nhận thấy nhất của thừa cân béo phì gia tăng trọng lượng cơ thể và khối lượng mỡ tích tụ tại một số phần đặc biệt của cơ thể như: bụng, đùi, eo, ngay cả ở ngực nữa.

**Phân loại mức độ béo phì với người trưởng thành :**

* Gầy : BMI < 18,5
* Bình thường : BMI từ 18,5 đến 24,9
* Thừa cân (không béo phì), nếu BMI từ 25,0 đến 29,9
* Béo phì loại 1 (nguy cơ thấp), nếu BMI từ 30,0 đến 34,9
* Béo phì loại 2 (nguy cơ trung bình), nếu BMI từ 35,0 đến 39,9
* Béo phì loại 3 (nguy cơ cao), nếu BMI bằng hoặc lớn hơn 40,0

**Phân loại béo phì**

**Phân loại theo tuổi:**

- Béo phì bắt đầu ở tuổi trưởng thành .

- Béo phì thiếu niên.

**Phân loại theo sự phân bố mỡ:**

- Béo phì dạng nam: mỡ tập trung chủ yếu ở gáy, cổ, mặt, vai, cánh tay, ngực, bụng trên rốn.

- Béo phì dạng nữ: mỡ tập trung chủ yếu ở đùi, mông, cẳng chân.

- Béo phì hỗn hợp: mỡ phân bố khá đồng đều.

**Nguyên nhân gây ra thừa cân béo phì**:

**Sự mất cân bằng giữa năng lượng ăn vào và năng lượng tiêu hao**. Khẩu phần năng lượng ăn vào vượt quá nhu cầu năng lượng của cơ thể, do đó năng lượng bị dư thừa và được chuyển thành mỡ tích lũy trong các cơ quan

**Yếu tố di truyền**: Có những người mang một số gen trong các nhóm gen như nhóm gen kích thích sự ngon miệng, nhóm gen liên quan đến tiêu hao năng lượng, nhóm gen điều hoà chuyển hoá, nhóm gen liên quan đến sự biệt hoá và phát triển tế bào mỡ. Trong gia đình nếu cha hoặc mẹ hoặc thậm chí là cả hai đều béo phì thì nguy cơ con cái bị thừa cân sẽ rất cao. Những người bị thừa cân béo phì do yếu tố di truyền có tốc độ trao đổi chất chậm chạp, khó cải thiện.

**Chế độ ăn uống không lành mạnh.** Chế độ ăn nhiều calo, thiếu trái cây và rau quả, nhiều thức ăn nhanh, đồ uống có hàm lượng calo cao và khẩu phần ăn quá lớn sẽ dẫn đến tăng cân.

**Lười vận động.** Nếu chúng ta có lối sống thụ động, cơ thể dễ dàng hấp thụ nhiều calo hơn mỗi ngày so với lượng calo đốt cháy thông qua tập thể dục và các hoạt động thường ngày. Nhìn vào màn hình máy tính, máy tính bảng và điện thoại là không hoạt động. Số giờ ngồi trước màn hình có liên quan chặt chẽ đến việc tăng cân.

**Phương pháp, phạm vi nghiên cứu**

**Phương pháp nghiên cứu:**

* Phân tích dữ liệu: Thu thập dữ liệu từ các nguồn đáng tin cậy, bao gồm thông tin về nhân khẩu học (tuổi, giới tính), hành vi (thói quen ăn uống, hoạt động thể chất), và tình trạng sức khỏe (BMI, tiền sử bệnh). Sau đó, xử lý và chuẩn hóa dữ liệu để loại bỏ những điểm không hợp lệ và chuẩn bị cho mô hình dự đoán.
* Xây dựng mô hình: Sử dụng các công cụ như TensorFlow hoặc Scikit-learn để phát triển mô hình dự đoán dựa trên các thuật toán học máy có giám sát như hồi quy logistic, cây quyết định...
* Huấn luyện và đánh giá mô hình: Huấn luyện mô hình với dữ liệu đã thu thập, sau đó đánh giá hiệu quả của nó dựa trên các chỉ số như độ chính xác (Accuracy), F1-Score,.. để đo độ chính xác của dự đoán.

**Phạm vi nghiên cứu:**

* Phạm vi dữ liệu: Nghiên cứu sẽ sử dụng dữ liệu từ các nguồn công khai và đã được kiểm chứng. Dữ liệu bao gồm thông tin về sức khỏe, thói quen sống, và các yếu tố liên quan đến nguy cơ béo phì như BMI, chế độ ăn uống, hoạt động thể chất
* Phạm vi kỹ thuật: Tập trung vào việc sử dụng các mô hình học máy như hồi quy logistic, cây quyết định... Ngoài ra, sẽ thử nghiệm thêm các kỹ thuật khác như Random Forest và tối ưu hóa mô hình để cải thiện độ chính xác dự đoán.

**Nội dung nghiên cứu**

Ứng dụng các kỹ thuật máy học để dự đoán khả năng một người mắc bệnh béo phì dựa trên các thông tin cá nhân và sức khỏe của họ.

# KHẢO SÁT NGHIỆP VỤ VÀ THU THẬP DỮ LIỆU

## Khảo sát nghiệp vụ

### Xác định đối tượng nghiệp vụ

Bệnh béo phì và các đặc trưng liên quan

### Tiếp cận và đánh giá tình huống

Vào năm 2022, cứ 8 người trên thế giới thì có 1 người mắc bệnh béo phì. Tỷ lệ béo phì ở người lớn trên toàn thế giới đã tăng gấp đôi kể từ năm 1990 và tỷ lệ béo phì ở thanh thiếu niên đã tăng gấp bốn lần. Có khoảng 2,5 tỷ người lớn (18 tuổi trở lên) bị thừa cân. Trong số này, 890 triệu người đang sống chung với tình trạng béo phì. [2]

Các phương pháp chẩn đoán béo phì truyền thống chủ yếu dựa vào chỉ số khối cơ thể (BMI) và vòng eo, tuy nhiên những chỉ số này chưa đủ để đánh giá chính xác nguy cơ mắc bệnh ở giai đoạn sớm. Ngược lại, học máy, với khả năng phân tích dữ liệu khổng lồ và xây dựng các mô hình dự đoán phức tạp, có thể phát hiện những dấu hiệu sớm của bệnh béo phì mà các phương pháp truyền thống bỏ sót.

### Xác định mục tiêu

* Phân loại đối tượng: Phân loại các cá nhân thành các nhóm có nguy cơ cao hoặc thấp mắc bệnh béo phì để có thể đưa ra các biện pháp can thiệp phù hợp.
* Dự đoán sớm: Phát hiện sớm những dấu hiệu của bệnh béo phì để có thể can thiệp kịp thời và hiệu quả.

### Lập kế hoạch dự án

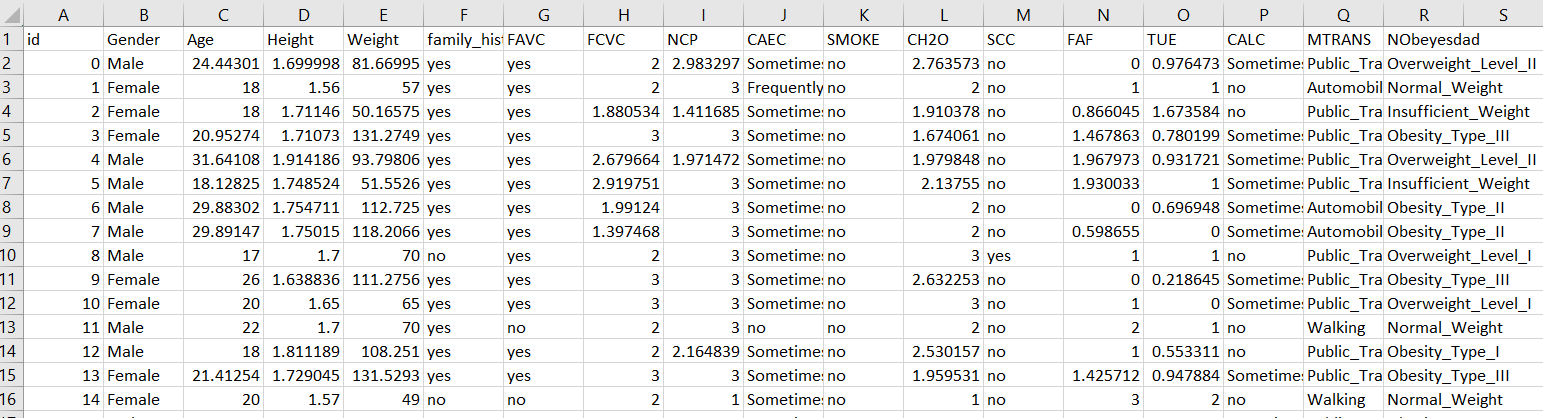
* Khảo sát nghiệp vụ : xác định đối tượng nghiệp vụ , tiếp cận , đánh giá tình huống , xác định mục tiêu , lập kế hoạch
* Thu thập dữ liệu : thu thập dữ liệu từ các trang uy tín, mô tả dữ liệu , khám phá dữ liệu , kiểm tra chất lượng dữ liệu
* Tiền xử lý dữ liệu : chọn dữ liệu, làm sạch dữ liệu,tích hợp dữ liệu, xử lý các giá trị thiếu, sai sót,ngoại lệ, loại bỏ giá trị trùng lặp, chuyển đổi dữ liệu chưa phù hợp
* Mô hình hóa : chọn kỹ thuật mô hình hóa, tạo ra mô hình thử nghiệm , xây dựng mô hình ,đánh giá mô hình
* Đánh giá hiệu suất :
* Đánh giá kết quả : so sánh kết quả dự đoán với kết quả thực tế. ,đánh giá khả năng tổng quát hóa của mô hình.
* Xem xét quá trình
* Xác định bước tiếp theo
* Triển khai ứng dụng :
* Kế hoạch triển khai ứng dụng : thiết kế giao diện đơn giản , thân thiện người dùng
* Thu thập ý kiến phản hồi của người dùng , cập nhật bảo trì ứng dụng khi cần thiết
* Lập báo cáo cuối cùng : tóm tắt quá trình thực hiện, kết quả đạt được , những hạn chế và đề xuất cải tiến
* Đánh giá dự án

## Thu thập dữ liệu

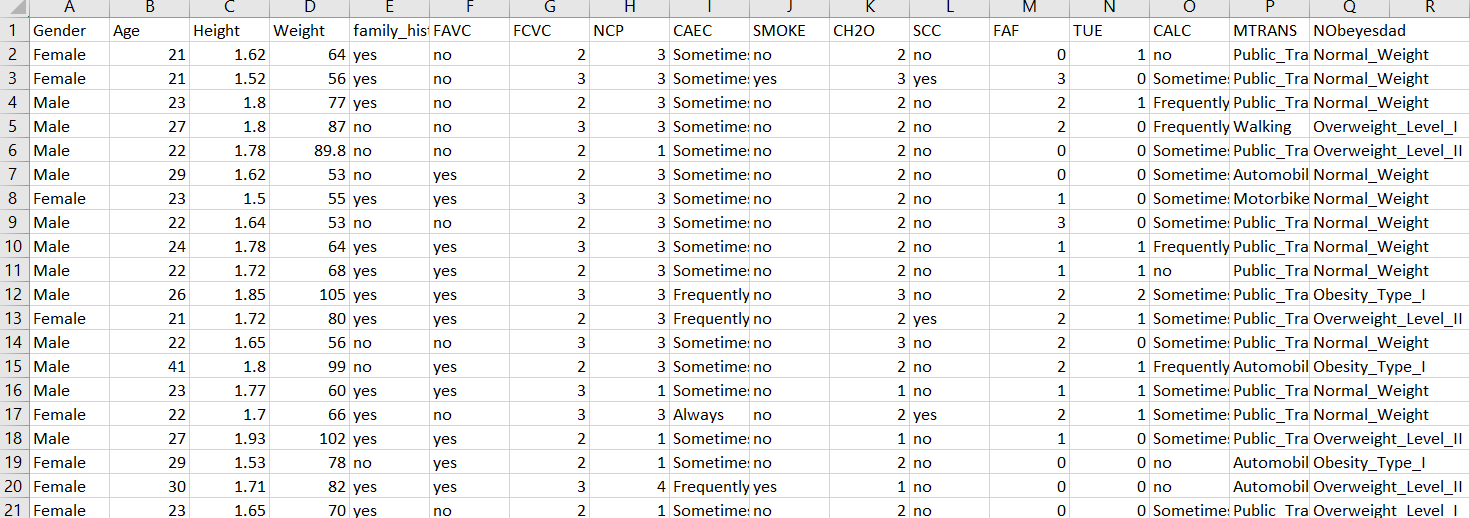
### Thu thập dữ liệu ban đầu

Phương pháp thu thập dữ liệu : từ nguồn hiện có

Dữ liệu được lấy từ 2 trang Kaggle và UC Irvine Machine Learning Repository



Hình 1‑1. Dataset lấy từ Kaggle



Hình 1‑2. Dataset lấy từ UC Irvine Machine Learning Repository

### Mô tả dữ liệu

Dataset 1 (Kaggle)

Dữ liệu ban đầu gồm 20758 bản ghi và 18 trường

* ID : giá trị số , tăng tự động từ 0 đến 20757
* Gender (giới tính) : gồm Male (Nam) và Female (Nữ)
* Age (tuổi) : giá trị số , phạm vi trong tập dữ liệu từ 14 đến 61 tuổi
* Height (chiều cao) : giá trị số (m) , phạm vi trong tập dữ liệu từ 1.45 đến 1.98
* Weight (cân nặng) : giá trị số (kg) ,phạm vi trong tập dữ liệu từ 39 đến 173
* Family\_history\_with\_overweight : gồm 2 giá trị Yes và No
* FAVC (Frequent consumption of high caloric food - Thường xuyên tiêu thụ thực phẩm nhiều calo): gồm 2 giá trị Yes và No
* FCVC(Frequency of consumption of vegetables - Tần suất tiêu thụ rau) : phạm vi trong tập dữ liệu từ 1 (ít) đến 3 (nhiều)
* NCP (Number of main meals - số bữa ăn chính mỗi ngày) : phạm vi trong tập dữ liệu từ 1 đến 3
* CAEC (Consumption of food between meals - Mức tiêu thụ thực phẩm giữa các bữa ăn) : gồm 3 giá trị Sometimes , No, Frequently
* Smoke (hút thuốc): gồm 2 giá trị Yes và No
* CH2O (Consumption of water daily - lượng nước uống hàng ngày) : giá trị số, phạm vi trong tập dữ liệu từ 1 đến 3
* SCC (Calories consumption monitoring - kiểm soát lượng calo hàng ngày): gồm 2 giá trị Yes và No
* FAF (Physical activity frequency - tần suất vận động thể chất) : giá trị số (ngày) phạm vi trong tập dữ liệu từ 0 đến 3
* TUE (Time using technology devices - thời gian sử dụng thiết bị công nghệ) : phạm vi trong tập dữ liệu từ 0 (không) đến 2 (nhiều)
* CALC (Consumption of alcohol - tần suất sử dụng đồ uống có cồn) : gồm 3 giá trị Sometimes, No, Frequently
* MTRANS (phương tiện giao thông thường dùng) : gồm 3 giá trị Automobile , Public\_Transportation, Walking
* Nobeyesdad (Mức độ béo phì) : gồm 8 giá trị Insufficient\_Weight, Normal\_Weight,Obesity\_Type\_I,Obesity\_Type\_II,Obesity\_Type\_III,Overweight\_Level\_I,Overweight\_Level\_II,Overweight\_Level\_III

Dataset 2 (UC Irvine Machine Learning Repository) : dữ liệu ban đầu gồm 2112 bản ghi và 17 trường (không có trường ID , các trường đều giống với Dataset 1)

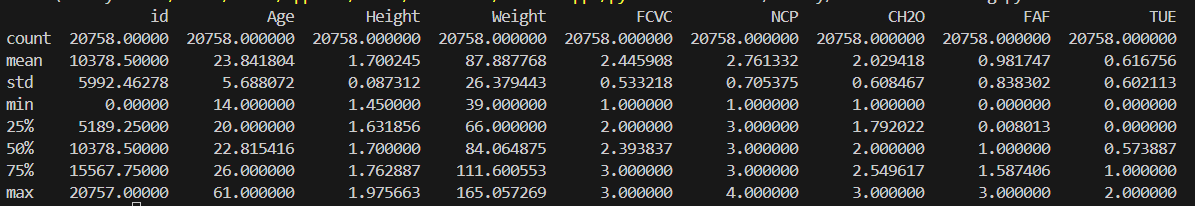
### Khám phá dữ liệu

**Dataset 1:**

import pandas as pd

training\_data = pd.read\_csv('./dataset/Kaggle/obesity\_prediction.csv')

print(training\_data.describe())



Hình 1‑3. Khám phá Dataset 1

* count: số lượng phần tử không bị khuyết,
* mean: giá trị trung bình,
* std: độ lệch chuẩn
* min: giá trị nhỏ nhất
* max: giá trị lớn nhất
* 50%: trung vị.
* 25%: trung vị của các giá trị từ min tới 50%
* 75%: trung vị của các giá trị từ 50% tới max

Nhận xét :

Trường Age (Tuổi):

Giá trị trung bình của độ tuổi là 23,84. Độ tuổi nhỏ nhất là 14, và lớn nhất là 61, cho thấy dữ liệu được thu thập hầu hết ở người trẻ .

Trường Height (Chiều cao):

Chiều cao trung bình của người tham gia là 1,7 mét với độ lệch chuẩn rất thấp (0,0873), cho thấy giá trị chiều cao phân bố khá đồng đều. Chiều cao lớn nhất là 1,98 mét và nhỏ nhất là 1,45 mét.

Trường Weight (Cân nặng):

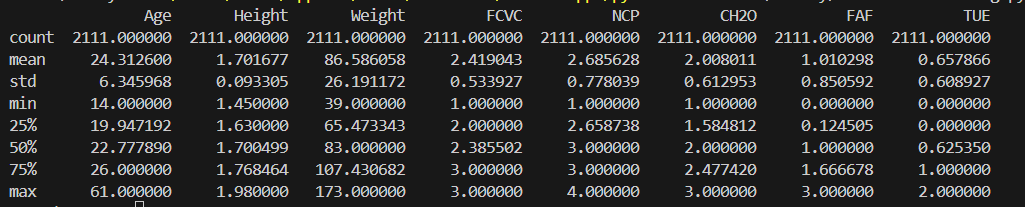
Cân nặng có sự dao động lớn, với giá trị trung bình là 87,89 kg và độ lệch chuẩn 26,38 kg. Trọng lượng lớn nhất là 165 kg, và nhỏ nhất là 39 kg,thể hiện sự đa dạng của dữ liệu.

**Dataset 2:**

import pandas as pd

training\_data=pd.read\_csv('./dataset/dataset2.csv')

print(training\_data.describe())

Hình 1‑4. Khám phá Dataset 2

Nhận xét:

Trường Age (Tuổi):

Giá trị trung bình của độ tuổi là 24,31, và độ tuổi nhỏ nhất là 14 trong khi lớn nhất là 61. Khoảng 50% bản ghi có tuổi dưới 22,78 (giá trị trung vị), cho thấy tập dữ liệu tập trung vào nhóm tuổi trẻ.

Trường Height (Chiều cao):

Chiều cao trung bình là 1,7 mét, với độ lệch chuẩn nhỏ (0,0933), cho thấy chiều cao của các đối tượng khá đồng đều. Chiều cao lớn nhất là 1,98 mét, nhỏ nhất là 1,45 mét.

Trường Weight (Cân nặng):

Cân nặng trung bình là 86,59 kg, với độ lệch chuẩn lớn (26,19 kg), điều này cho thấy sự khác biệt khá lớn trong cân nặng giữa các đối tượng. Cân nặng nhỏ nhất là 39 kg, trong khi cân nặng lớn nhất lên tới 173 kg.

# TIỀN XỬ LÝ DỮ LIỆU VÀ MÔ HÌNH HÓA

## Tiền xử lý dữ liệu

### Chọn dữ liệu

**Dataset 1**

Bỏ các trường FAVC,FCVC,TUE,MTRANS,CALC,ID,CAEC,SCC,Smoke

* gender,age,height,weight,family\_history\_with\_overweight,CH2O,

FAF,NObeyesdad

**Dataset 2**

Bỏ các trường FAVC,FCVC,TUE,MTRANS,CALC,CAEC,SCC,Smoke

* gender,age,height,weight,family\_history\_with\_overweight,CH2O,

FAF,NObeyesdad

### Làm sạch dữ liệu

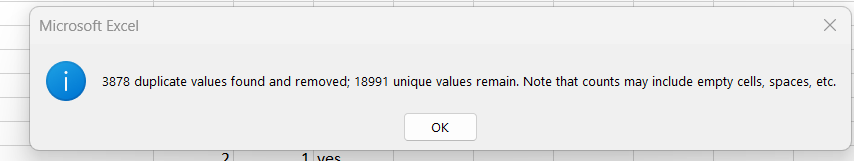
* Loại bỏ dữ liệu không hợp lệ ở trường Age: những bản ghi có trường Age là số thập phân sẽ bị loại bỏ
* Chuyển đổi kiểu dữ liệu số :
* Trường Height,Weight : làm tròn đến chữ số thập phân thứ 2
* Các trường còn lại: làm tròn đến số nguyên gần nhất

### Xây dựng dữ liệu

* Tạo thêm trường BMI từ 2 trường Height và Weight
* Đổi tên trường NObeyesdad thành Obesity
* Biến đổi các thuộc tính danh nghĩa ở các trường gender, obesity, family\_history\_with\_overweight
* Gender: female = 0; male = 1.
* family\_history\_with\_overweight : yes = 1, no = 0.
* Obesity:Insufficeint\_Weight,Normal\_Weight,Overweight\_Level\_I,Overweight\_Level\_II = 0, còn lại : 1

### Tích hợp dữ liệu

Tích hợp 2 dataset , loại bỏ 3878 bản ghi có dữ liệu trùng lặp . Dataset hiện tại có 18891 bản ghi.



Hình 2‑1.Tích hợp dữ liệu

Cân bằng dữ liệu: cân bằng số lượng yes và no ở nhãn Obesity

import pandas as pd

from imblearn.under\_sampling import RandomUnderSampler

data = pd.read\_csv('./dataset/obesity\_prediction\_final\_dataset.csv')

X = data.iloc[:, :-1]  # thuộc tính

y = data.iloc[:, -1]   # nhãn

rus = RandomUnderSampler(random\_state=42)

X\_resampled, y\_resampled = rus.fit\_resample(X, y)

import matplotlib.pyplot as plt

# Đếm số lượng mẫu của mỗi lớp trong dữ liệu mới

class\_count = pd.Series(y\_resampled).value\_counts()

# Vẽ biểu đồ

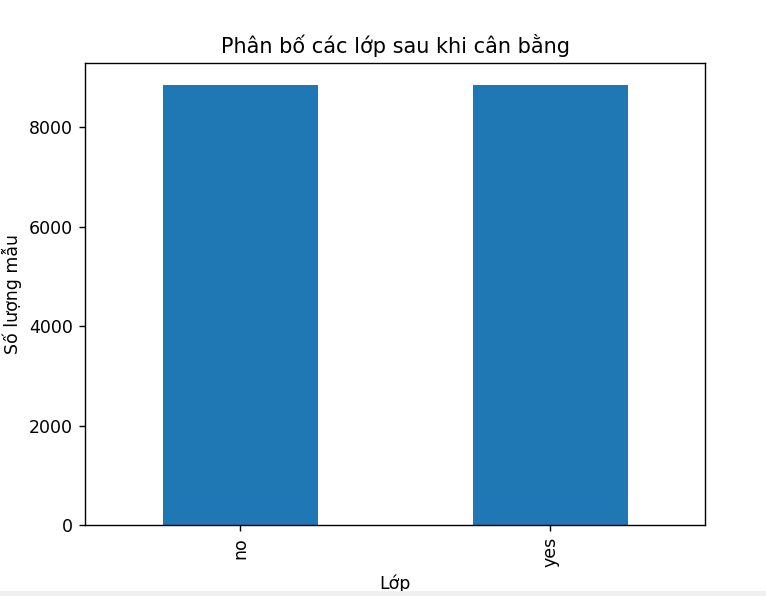
class\_count.plot(kind='bar')

plt.title('Phân bố các lớp sau khi cân bằng')

plt.xlabel('Lớp')

plt.ylabel('Số lượng mẫu')

plt.show()



Hình 2‑2.Phân bố lớp yes và no của nhãn Obesity sau cân bằng

* Dataset còn lại 17698 bản ghi

## Mô hình hóa

### Lựa chọn kĩ thuật mô hình hóa

Sử dụng các thuật toán Naive Bayes,Decision Tree,Random Forest

Sử dụng các thư viện panda,scikitlearn của ngôn ngữ lập trình Python để thực hiện mô hình hóa

### Tạo thiết kế thử nghiệm (Test)

data = pd.read\_csv('./dataset/obesity\_prediction\_dataset\_final.csv')

X=data[['Gender','Age','Height','Weight','BMI','family\_history\_with\_overweight','CH2O','FAF']]

y = data['Obesity']

# Chia dataset thành training set và test set (training 80% test 20%)

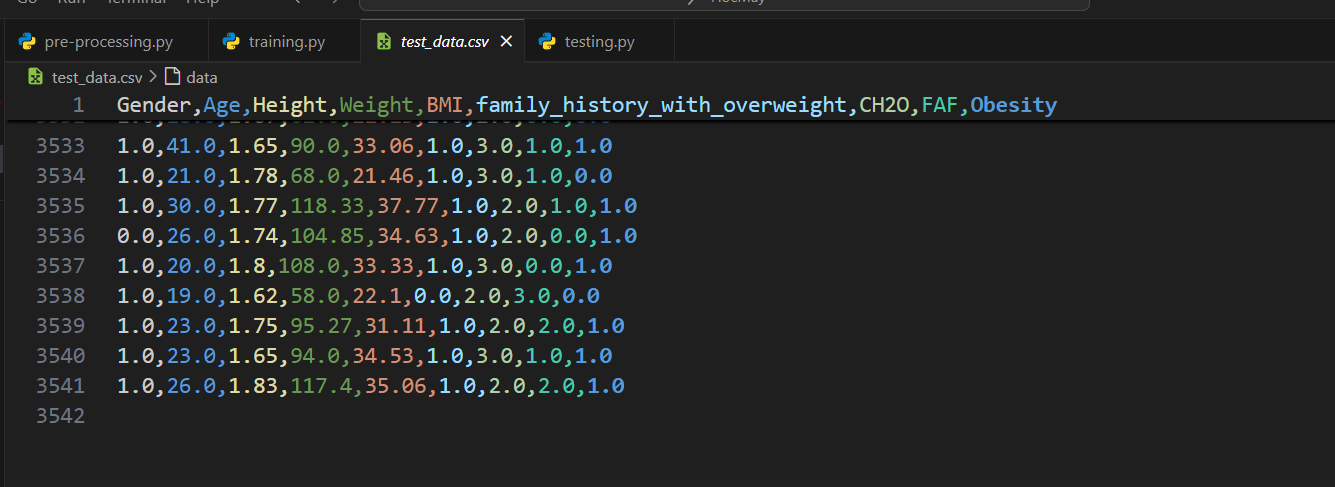
X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.2,

random\_state=42)

# Lưu tập test

test\_data=pd.DataFrame(data=np.column\_stack([X\_test, y\_test]), columns=data.columns)

test\_data.to\_csv("test\_data.csv", index=False)



Hình 2‑3.Test set có 3541 bản ghi

### Xây dựng mô hình

* Naive Bayes

import pandas as pd

import numpy as np

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

import joblib

from sklearn.metrics import \*

from sklearn.naive\_bayes import CategoricalNB

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

# Đọc dataset

data = pd.read\_csv('./dataset/obesity\_prediction\_dataset\_final.csv')

X=data[['Gender','Age','Height','Weight','BMI','family\_history\_with\_overweight','CH2O','FAF']]

y = data['Obesity']

# Chia tập training thành training set và validation set(training 60% validation 20%)

X\_train,X\_val,y\_train,y\_val=train\_test\_split(X\_train, y\_train, test\_size=0.25, random\_state=42)

# NaiveBayes

modelNB = CategoricalNB()

modelNB.fit(X\_train,y\_train)

y\_pred = modelNB.predict(X\_val)

joblib.dump(modelNB,'modelNB.csv')

* Decision Tree

modelDT = DecisionTreeClassifier()

modelDT.fit(X\_train,y\_train)

y\_pred = modelDT.predict(X\_val)

joblib.dump(modelDT,'./model/modelDT.csv')

* Random Forest

modelRF = RandomForestClassifier()

modelRF.fit(X\_train,y\_train)

y\_pred = modelRF.predict(X\_val)

joblib.dump(modelRF,'./model/modelRF.csv')

### Mô hình truy cập

So sánh hiệu suất trên 3 tập dữ liệu

A: Gender,Age,BMI,family\_history\_with\_overweight,CH2O,FAF,Obesity

B:Gender,Age,Height,Weight,family\_history\_with\_overweight,CH2O,FAF,Obesity

C:Gender,Age,Height,Weight,BMI,family\_history\_with\_overweight,CH2O,FAF,Obesity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Dataset | Thuật toán | Accuracy | Precision | Recall | F1-score |
| A | Naive Bayes | 96.04% | 95.97% | 96.29% | 96.13% |
|  | Decision Tree | 94.12% | 94.1% | 94.41% | 94.25% |
|  | Random Forest | 96,07% | 96,13% | 96,18% | 96,15% |
| B | Naive Bayes | 92.97% | 91.83% | 94.58% | 93.18% |
|  | Decision Tree | 95.02% | 95.13% | 95.13% | 95.13% |
|  | Random Forest | 96.92% | 97.43% | 96.51% | 96.96% |
| C | Naive Bayes | 96,98% | 97,21% | 96,73% | 96,93% |
|  | Decision Tree | 95,28% | 95,3% | 95,46% | 95,38% |
|  | Random Forest | 97% | 97,22% | 96,89% | 97,05% |

Tập dữ liệu C có hiệu suất cao nhất và ổn định trên cả ba thuật toán. Các chỉ số hiệu suất chênh lệch rất nhỏ, cho thấy tính ổn định của mô hình trên tập này. Do đó, tập dữ liệu C là lựa chọn phù hợp nhất cho bài toán."

# ĐÁNH GIÁ MÔ HÌNH VÀ TRIỂN KHAI ỨNG DỤNG

## Đánh giá mô hình

### Đánh giá kết quả

import pandas as pd

import joblib

from sklearn.metrics import \*

# Đọc tập test

test\_data = pd.read\_csv('test\_data.csv')

# Đọc model

# Random Forest

model = joblib.load('./model/modelRF.pkl')

# Decision Tree

model = joblib.load('./model/modelDT.pkl')

# Naive Bayes

model = joblib.load('./model/modelNB.pkl')

# Các thuộc tính

new\_X=test\_data[['Gender','Age','Height','Weight','family\_history\_with\_overweight','CH2O','FAF']]

# Nhãn

true\_y = test\_data['Obesity']

# Dự đoán trên tập test

new\_y\_pred = model.predict(new\_X)

accuracy = accuracy\_score(true\_y, new\_y\_pred)

precision = precision\_score(true\_y, new\_y\_pred)

recall = recall\_score(true\_y,new\_y\_pred)

f1 = f1\_score(true\_y,new\_y\_pred)

print(f"Accuracy: {accuracy}")

print(f"Precision: {precision}")

print(f"Recall: {recall}")

print(f"F1-score: {f1}")

**So sánh hiệu suất 3 thuật toán trên tập validation và test**

Naive Bayes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy | Precision | Recall | F1-score |
| Validation | 96,92% | 97,21% | 96,73% | 96,97% |
| Test | 96,47% | 96,61% | 96,45% | 96,53% |

Decision Tree

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy | Precision | Recall | F1-score |
| Validation | 95,28% | 95,3% | 95,46% | 95,38% |
| Test | 94,94% | 95% | 95,01% | 95,04% |

Random Forest

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy | Precision | Recall | F1-score |
| Validation | 97% | 97,22% | 96,89% | 97,05% |
| Test | 96,72% | 96,94% | 96,62% | 96,78% |

**Nhận xét:** Cả ba thuật toán đều có hiệu suất cao với độ chính xác (accuracy) dao động từ 94.94% đến 97%. Trong số đó, **Random Forest** có hiệu suất cao nhất trên cả tập validation và tập test. Độ chính xác, precision, recall, và F1-score của nó đều cao nhất, cho thấy mô hình này có khả năng phân loại tốt và nhất quán. => chọn thuật toán **Random Forest** cho mô hình để phát triển ứng dụng

### Đánh giá quá trình

* Dữ liệu đã được làm sạch , không có dữ liệu bị thiếu, nhãn phân loại đã được cân bằng giữa trường yes , no
* Sử dụng Cross-validation để kiểm tra hiệu suất mô hình trên tập dữ liệu mới

import pandas as pd

import joblib

from sklearn.metrics import \*

from sklearn.model\_selection import cross\_val\_score

test\_data = pd.read\_csv('test\_data.csv')

model = joblib.load('./model/modelRF.pkl')

new\_X=test\_data[['Gender','Age','Height','Weight','BMI','family\_history\_with\_overweight','CH2O','FAF']]

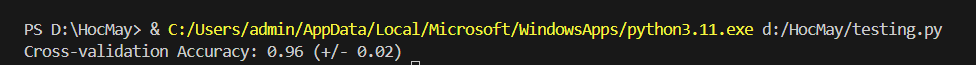
true\_y = test\_data['Obesity']

new\_y\_pred = model.predict(new\_X)

# Sử dụng cross-validation , chia dữ liệu thành 5 phần

scores = cross\_val\_score(model, new\_X, true\_y, cv=5)

print(f"Cross-validation Accuracy: {scores.mean():.2f} (+/- {scores.std() \* 2:.2f})")



#Sử dụng cross-validation , chia dữ liệu thành 10 phần

scores = cross\_val\_score(model, new\_X, true\_y, cv=10)

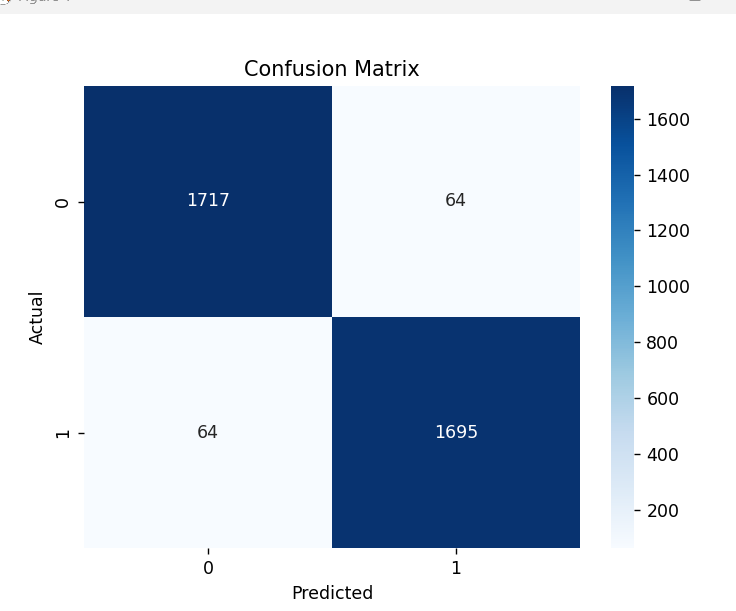
print(f"Cross-validation Accuracy: {scores.mean():.2f} (+/- {scores.std() \* 2:.2f})")



**Mô hình có hiệu suất tốt:** Độ chính xác trung bình 0.96 cho thấy mô hình có khả năng dự đoán chính xác khá tốt trên dữ liệu mới.

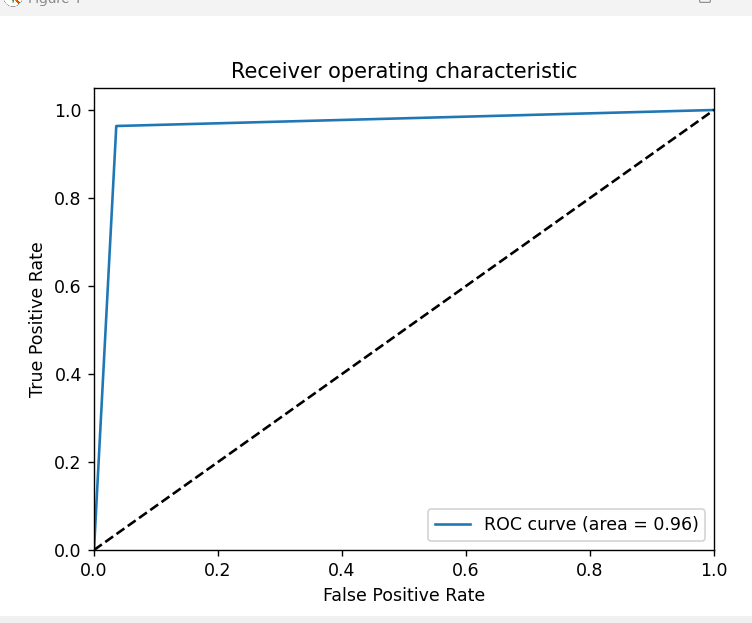
**Độ tin cậy tương đối cao:** Khoảng tin cậy 0.01- 0.02 tương đối nhỏ, cho thấy kết quả cross-validation tương đối ổn định và không bị ảnh hưởng quá nhiều bởi sự ngẫu nhiên trong việc chia dữ liệu thành các fold.

**Ma trận nhầm lẫn (Confusion Matrix)**



Hình 3‑1.Ma trận nhầm lẫn

Mô hình đã phân loại đúng 1717 mẫu thuộc lớp 0 và 1695 mẫu thuộc lớp 1. Có 64 mẫu thuộc lớp 0 bị phân loại sai thành lớp 1 và 64 mẫu thuộc lớp 1 bị phân loại sai thành lớp 0.



Hình 3‑2.Đường cong ROC

AUC = 0.96 cho thấy mô hình có khả năng phân loại rất tốt

Đường cong gần về bên trái => tỷ lệ dương tính thật cao và tỷ lệ dương tính giả thấp.

=>Mô hình đã phân loại đúng một lượng lớn mẫu dữ liệu, cho thấy hiệu suất khá tốt. phân loại tốt cả hai lớp, không có sự chênh lệch quá lớn về số lượng mẫu bị phân loại sai giữa hai lớp.

### Xác định bước tiếp theo

Triển khai ứng dụng

## 3.2. Triển khai ứng dụng

### 3.2.1.Kế hoạch triển khai

**Tích hợp mô hình vào ứng dụng web**

# Import thư viện cần thiết

from flask import Flask,render\_template,request

import numpy as np

from sklearn.ensemble import RandomForestClassifier

import joblib

# Load model Random Forest

model = joblib.load("modelRF.pkl")

# Chạy ứng dụng web

app = Flask(\_\_name\_\_)

@app.route("/")

def home():

return render\_template("index.html")

@app.route("/",methods=["POST"])

def predict():

gender = int(request.form['gender'])

age = int(request.form['age'])

height = float(request.form['height'])

weight = float(request.form['weight'])

bmi = float(request.form['bmi'])

overweight\_history = int(request.form['overweight'])

ch20 = int(request.form['ch20'])

faf = int(request.form['faf'])

data=np.array([[gender,age,height,weight,bmi,overweight\_history,ch20,faf]])

my\_prediction = model.predict(data)

if int(my\_prediction) == 1:

prediction = "Bạn bị béo phì"

elif int(my\_prediction) == 0:

prediction = "Bạn không bị béo phì"

return render\_template("index.html", prediction=prediction)

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

Kết quả

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### 3.2.2.Lập kế hoạch giám sát và bảo trì

* Thường xuyên kiểm tra và cập nhật mô hình dự đoán dựa trên dữ liệu mới.
* Bảo trì server Flask để đảm bảo ứng dụng chạy ổn định.

### 3.2.3.Lập báo cáo cuối cùng

**Ưu điểm**

* Hiệu suất của mô hình: Mô hình Random Forest được huấn luyện và đánh giá đạt độ chính xác trung bình 0.96 trên các tập dữ liệu kiểm tra, cho thấy khả năng phân loại đáng tin cậy và chính xác.
* Độ ổn định của mô hình: Kết quả cross-validation trên nhiều phần chia dữ liệu (5-fold và 10-fold) đạt được độ lệch chuẩn nhỏ, thể hiện sự ổn định và khả năng duy trì hiệu suất cao khi gặp dữ liệu mới.
* Ma trận nhầm lẫn: Mô hình cho thấy số lượng mẫu bị phân loại sai giữa các lớp là tương đối thấp và cân bằng giữa các lớp béo phì và không béo phì.

**Nhược điểm**

* Yêu cầu tài nguyên tính toán cao: Random Forest với nhiều cây quyết định đòi hỏi tài nguyên bộ nhớ và xử lý lớn, nhất là khi áp dụng trên tập dữ liệu lớn hơn.
* Giới hạn với dữ liệu hiếm: Mô hình Random Forest có thể gặp khó khăn khi phân loại những trường hợp béo phì hiếm gặp hoặc đặc biệt phức tạp.
* Thiếu khả năng tùy chỉnh chi tiết: Các thuật toán ensemble như Random Forest có thể thiếu tính linh hoạt nếu cần tối ưu sâu hơn cho từng nhóm dữ liệu cụ thể.
* Phụ thuộc vào chất lượng dữ liệu: Mô hình phụ thuộc vào tính đầy đủ và độ chính xác của dữ liệu đầu vào. Những sai sót trong dữ liệu (như dữ liệu thiếu hoặc không cân bằng) có thể ảnh hưởng đến hiệu suất của mô hình.

### 3.2.4.Đánh giá dự án

Trong quá trình làm bài tập lớn, nhóm đã thực hiện các công việc sau:

* Nghiên cứu về các phương pháp học máy có giám sát.
* Áp dụng các giai đoạn của chu trình học máy vào bài toán thực tế
* Thực hiện đánh giá, so sánh 3 mô hình học máy để lựa chọn mô hình học máy tốt nhất.
* Tích hợp được mô hình học máy vào ứng dụng để chẩn đoán bệnh béo phì.

**TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | https://www.who.int/news/item/04-03-2022-world-obesity-day-2022-accelerating-action-to-stop-obesity. |
| [2] | https://www.vinmec.com/vie/bai-viet/nao-duoc-coi-la-thua-can-beo-phi-vi. |

[3] Bài giảng Khai phá dữ liệu và máy học, TS.Vũ Xuân Hạnh

[4] Machine Learning cơ bản , Vũ Hữu Tiệp