

### ĐẠI HỌC QUỐC GIA HÀ NỘI TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

-**--- 🙞🙜 🕮 🙞🙜 ----**



BÁO CÁO TIỂU LUẬN

**Môn: Học Máy**

**Đề tài: Mô hình dự đoán bị nhồi máu cơ tim ở bệnh nhân.**

Giảng viên : Cao Văn Chung

Sinh viên thực hiện**:**

1. Đinh Thị Minh Hằng – Msv: 22001574
2. Bùi Quang Minh – Msv: 20000552
3. Trần Duy Nam – Msv: 22000110

***Hà Nội, 2025***

Mục Lục

[ĐẠI HỌC QUỐC GIA HÀ NỘI TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN 1](#)

[**A. Tiền xử lý và phân tích dữ liệu 4**](#_jcqj8tke8xog)

[I. Tiền xử lý dữ liệu 4](#_tj10yevb9nhp)

[II. Giảm chiều dữ liệu 5](#_c17w8ziwmkw9)

[1. PCA (Principal Component Analysis) 5](#_8xr9per08qzb)

[2. LDA (Linear Discriminant Analysis) 6](#_6qpp92f6fs4z)

[**B. Phân cụm dữ liệu 8**](#_badtjvp2uarp)

[I. K-Means 8](#_oo41qjsj2ad)

[1. Cơ sở lý thuyết: 8](#_tk27n087uvcf)

[2. Quy trình thuật toán K-Means: 8](#_vr0vqmp3tbk7)

[3. Code mô tả: 8](#_ru5o3gvvlik5)

[II. GMM - Gaussian Mixture Model: 8](#_slsrz994zr3r)

[1. Cơ sở lý thuyết: 9](#_id33xmlh98xf)

[2. Các thành phần chính của GMM: 9](#_ri8pdo1kodmi)

[3. Code mô tả: 9](#_uo0vjonnnkh)

[**C. Áp dụng mô hình trong phân loại 9**](#_3vy89p99g172)

[I. KNN (K láng giềng gần nhất) 9](#_iyugtfhid5j5)

[1. Cơ sở lý thuyết: 9](#_obs1j03vstyg)

[2. Đánh giá: 13](#_4n1m98ui9dn9)

[II. SVM - LinearSVC 13](#_7ex9pjexm9cn)

[1. Cơ sở lý thuyết 13](#_y2hj1nvvw3ea)

[2. Huấn luyện mô hình 14](#_dz3cmzi3u8y)

[3. Trực quan hóa tương quan giữa thực tế và dự đoán 15](#_v0e64mfwls4o)

[4. Kết luận 16](#_3ha2hc3h3kp4)

[III. Logistic Regression 16](#_3qtucgdn90bo)

[1. Cơ sở lý thuyết 16](#_fftxorscpykg)

[2. Huấn luyện mô hình 16](#_11ldz4ik60fo)

**Tóm tắt báo cáo**

Báo cáo của nhóm em được thực hiện dựa trên mẫu dữ liệu liên quan đến sức khỏe ở Úc, dùng 3 mô hình với 2 phương pháp giảm chiều dữ liệu tương ứng để chẩn đoán nguy cơ bị nhồi máu cơ tim của bệnh nhân.

Link lấy dữ liệu: [*https://www.kaggle.com/datasets/ankushpanday2/heart-attack-prediction-in-indonesia?fbclid=IwY2xjawJNB4VleHRuA2FlbQIxMAABHSydyxsBWV9QdJa3\_snALB9nt2wEKE6qa370dQJCuhcfxhSYJNHhPV5tNg\_aem\_RSG6oU9Vafp5Tp0H2yKhGw*](https://www.kaggle.com/datasets/ankushpanday2/heart-attack-prediction-in-indonesia?fbclid=IwY2xjawJNB4VleHRuA2FlbQIxMAABHSydyxsBWV9QdJa3_snALB9nt2wEKE6qa370dQJCuhcfxhSYJNHhPV5tNg_aem_RSG6oU9Vafp5Tp0H2yKhGw)

3 mô hình dự đoán phân loại với 3 thành viên tương ứng :

* K nearest neighbors (K láng giềng gần nhất)
* SVM - Linear SVC
* Logistic Regression

2 phương pháp giảm chiều bao gồm: PCA và LDA

# Tiền xử lý và phân tích dữ liệu

## Tiền xử lý dữ liệu

Đoạn code dưới đây tải và đọc dữ liệu trực tiếp từ kaggle:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Ngoài ra có thể tải data về máy:



Trong file main:



Hiển thị tên các cột để nhìn trực quan thông số dữ liệu đầu vào:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Encode các cột phân loại:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Tách đặc trưng và nhãn:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

## **Giảm chiều dữ liệu**

### 1. PCA (Principal Component Analysis)

Là một kỹ thuật giảm chiều dữ liệu bằng cách biến đổi các đặc trưng gốc thành một tập hợp các trục tọa độ mới (principal components).

PCA phù hợp sử dụng để giảm chiều cho dữ liệu tuyến tính.

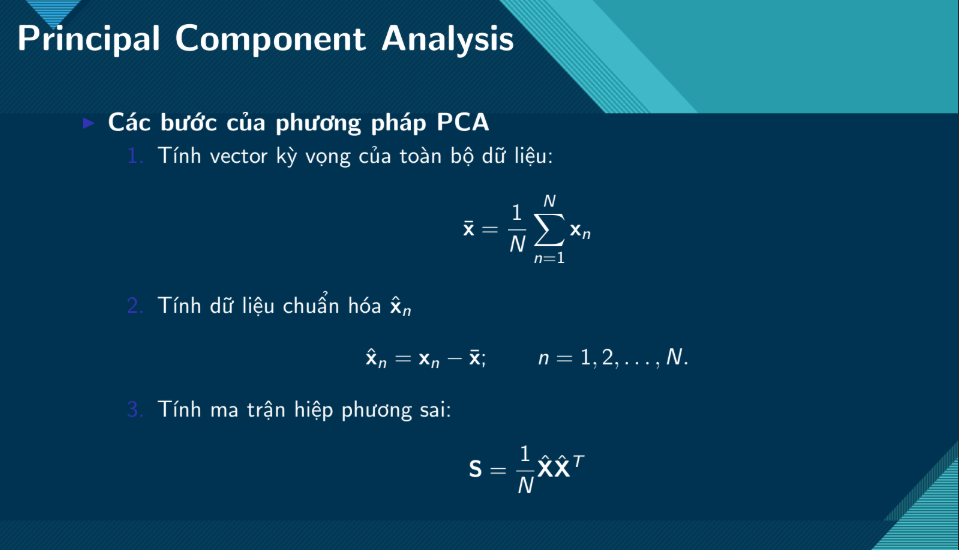
PCA là một phương pháp Unsupervised. Việc thực hiện PCA trên toàn bộ dữ liệu không phụ thuộc vào class(nếu có) của mỗi dữ liệu.

PCA đôi khi không hiệu quả thậm chí gây sai lệch khi áp dụng cho các bài toán classification.

Với các bài toán Large-scale, đôi khi việc tính toán trên toàn bộ dữ liệu là không khả thi vì còn có vấn đề về bộ nhớ. Giải pháp là thực hiện PCA lần đầu trên một tập con dữ liệu vừa với bộ nhớ, sau đó lấy một tập con khác để cập nhật nghiệm của PCA tới khi nào hội tụ.

Có nhiều hướng mở rộng của PCA, có thể tìm kiếm theo từ khoá: Sparse PCA, Kernel PCA, Robust PCA…

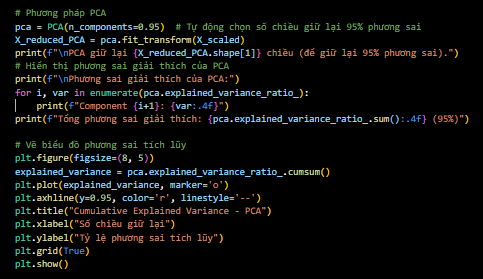
Các bước của phương pháp PCA:



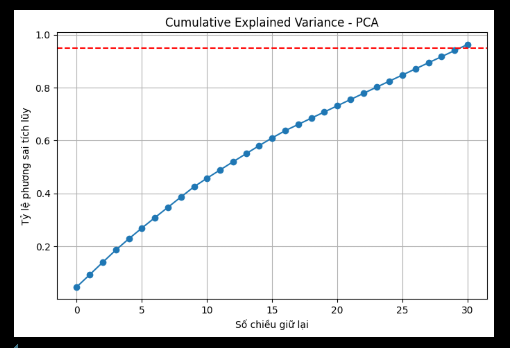
### 2. LDA (Linear Discriminant Analysis)

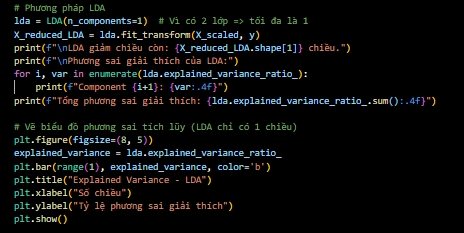
Là phương pháp giảm chiều dựa trên việc chiếu xuống không gian tuyến tính con (SubSpace), có tính đến việc áp dụng cho các loại dữ liệu có nhãn (dữ liệu có phân loại), ưu tiên độ phân biệt (Discriminant) giữa các nhóm dữ liệu tách được tuyến tính.

Đoạn code sau thực hiện lần lượt 2 phương pháp giảm chiều dữ liệu: PCA/LDA. Nếu chọn PCA, số chiều dữ liệu tùy chỉnh về mức giữ lại 95% phương sai. Phương pháp PCA được miêu tả thông qua đoạn code sau:

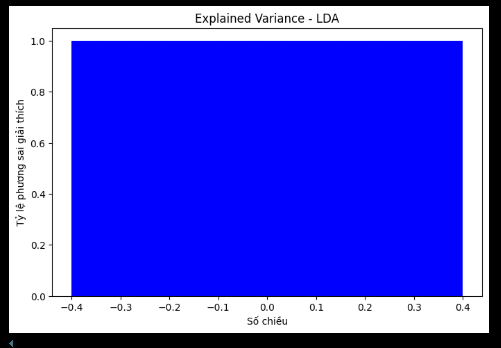


Kết quả:



Phương pháp LDA được thực hiện thông qua đoạn code sau:  
 

Kết quả:



# B. Phân cụm dữ liệu

## **I. K-Means**

### 1. Cơ sở lý thuyết:

K-Means chia dữ liệu thành K nhóm (clusters) sao cho các điểm trong cùng một nhóm thì gần nhau nhất về mặt khoảng cách (thường là Euclidean) và xa nhất so với các nhóm khác.

Elbow Method (Phương pháp khuỷu tay) là một kỹ thuật phổ biến trong phân cụm K-Means để xác định số lượng cụm tối ưu (K).

### 2. Quy trình thuật toán K-Means:

Chọn số cụm K (dựa theo Elbow Method).

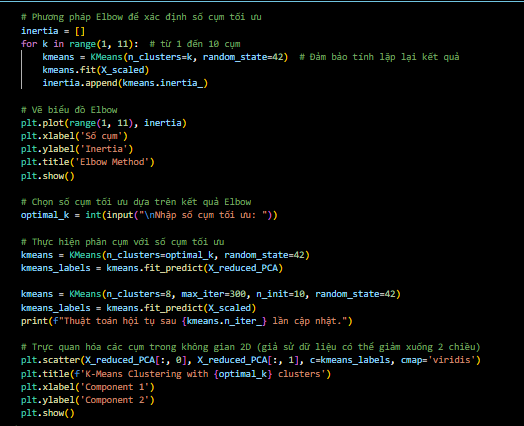
Khởi tạo K tâm cụm (centroids) ban đầu (có thể chọn ngẫu nhiên).

Gán mỗi điểm dữ liệu vào cụm có tâm gần nhất.

Tính lại tâm cụm (là trung bình của tất cả các điểm trong cụm đó).

Lặp lại bước 3–4 cho đến khi tâm cụm không đổi nữa hoặc đạt ngưỡng lặp

### 3. Code mô tả:

****

## **II. GMM - Gaussian Mixture Model:**

### 1. Cơ sở lý thuyết:

GMM - Gaussian Mixture Model (Mô hình hỗn hợp Gauss) là một mô hình thống kê dựa trên xác suất được dùng để mô phỏng dữ liệu có nhiều cụm (clusters), trong đó mỗi cụm được giả định là tuân theo một phân phối chuẩn (Gaussian).

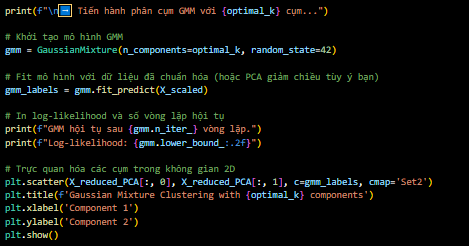
### 2. Các thành phần chính của GMM:

**Gaussian (phân phối chuẩn)**: Mỗi nhóm trong GMM được mô hình hóa bằng một phân phối Gaussian, đặc trưng bởi hai tham số:

* **Trung bình (mean)**: Vị trí trung tâm của phân phối.
* **Độ lệch chuẩn (standard deviation)**: Đo lường sự phân tán của dữ liệu xung quanh trung bình.

**Trọng số (weights)**: Mỗi phân phối Gaussian có một trọng số cho biết tỉ lệ dữ liệu thuộc vào nhóm đó. Tất cả trọng số này cộng lại phải bằng 1.

### 3. Code mô tả:

****

# C. Áp dụng mô hình trong phân loại

## **I. KNN (K láng giềng gần nhất)**

### 1. Cơ sở lý thuyết:

**KNN (K-Nearest Neighbors)** là một thuật toán phân loại hoặc hồi quy dựa trên khoảng cách đến các điểm lân cận gần nhất trong tập huấn luyện.

Các bước của phương pháp KNN để phân loại đầu vào x:

* Tính khoảng cách từ x đến tất cả các điểm trong tập huấn luyện (thường dùng khoảng cách Euclidean).
* Chọn K điểm gần nhất (lân cận nhất).
* Đếm nhãn (label) của K điểm đó.
* Phân loại: Gán nhãn phổ biến nhất trong số K điểm cho x.

Nếu là hồi quy → trung bình giá trị thay vì đa số.

KNN là một phương pháp đơn giản dễ triển khai, hoạt động tốt với những dữ liệu phân tách rõ ràng. Đối với các dữ liệu lớn, KNN trở nên kém hiệu quả vì cần thực hiện tính trung bình với tất cả các điểm.

1. **Phân loại:**

Huấn luyện mô hình KNN, sử dụng thư viện:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

In ra ma trận nhầm lẫn, các đánh giá phân loại và biểu đồ để đánh giá kết quả:

* + Đối với LDA:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, Hình chữ nhật

Nội dung do AI tạo ra có thể không chính xác.

* + Đối với PCA:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, ảnh chụp màn hình, số, bảng điểm

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Hình chữ nhật

Nội dung do AI tạo ra có thể không chính xác.

* + KNN nguyên bản:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, màu đen

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, Hình chữ nhật

Nội dung do AI tạo ra có thể không chính xác.

### Đánh giá:

* KNN + LDA tốt nhất về độ chính xác phân loại, nhưng sẽ giảm mạnh chiều (chỉ còn 1).
* KNN + PCA giữ lại nhiều thông tin hơn nhưng không phân biệt lớp mạnh bằng.
* KNN gốc (không giảm chiều) dễ bị overfit và tính toán chậm hơn.

## **II. SVM - LinearSVC**

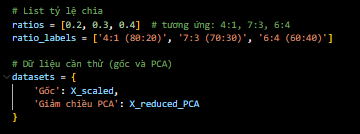
### 1. Cơ sở lý thuyết

**LinearSVC** là một biến thể của SVM, được tối ưu hóa cho các bài toán phân loại tuyến tính, đặc biệt là khi số lượng mẫu lớn và tính toán nhanh là quan trọng.

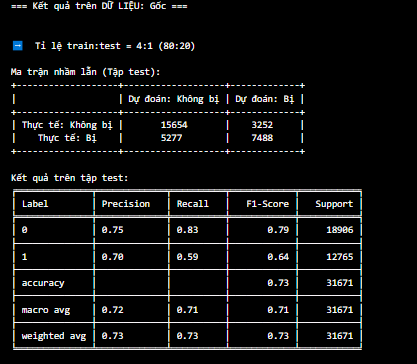
* LinearSVC sử dụng thuật toán tối ưu hóa khác so với SVM truyền thống (cụ thể là liblinear solver thay vì SMO), giúp giảm thời gian tính toán và làm cho thuật toán này phù hợp với dữ liệu lớn hơn.
* Phân loại tuyến tính: Giống như SVM, LinearSVC tìm một siêu phẳng tuyến tính để phân chia các lớp, nhưng khác biệt ở chỗ nó không sử dụng kernel để xử lý các bài toán phân loại phi tuyến tính. Vì vậy, LinearSVC chỉ áp dụng cho các bài toán phân loại mà dữ liệu có thể được phân chia rõ ràng bằng một siêu phẳng tuyến tính.

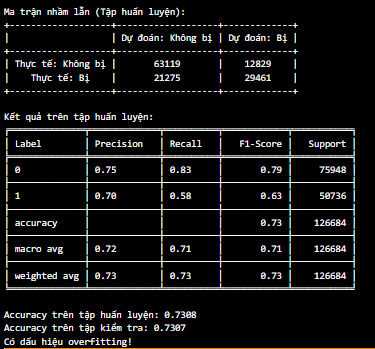
### **2. Huấn luyện mô hình**

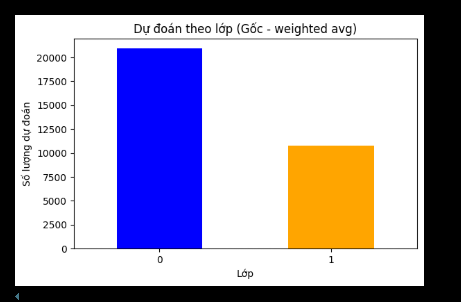
Chia tỉ lệ, chia data thành 2 phần: phần dữ liệu gốc và phần dữ liệu đã được giảm chiều (PCA)



Dưới đây là kết quả thu được trong trường hợp xử lý với dữ liệu gốc với tỉ lệ train:validation=4:1

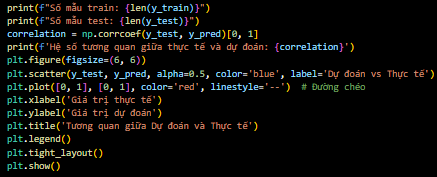


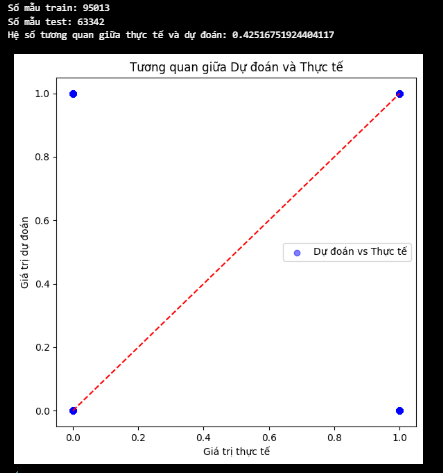




### **3. Trực quan hóa tương quan giữa thực tế và dự đoán**

Code thực hiện:

****



### **4. Kết luận**

Mô hình dự đoán có một sự tương quan nhất định với dữ liệu thực tế, nhưng còn khá nhiều sai lệch giữa dự đoán và giá trị thực tế. Mô hình có thể không chính xác hoàn toàn, và có thể cần được cải thiện.

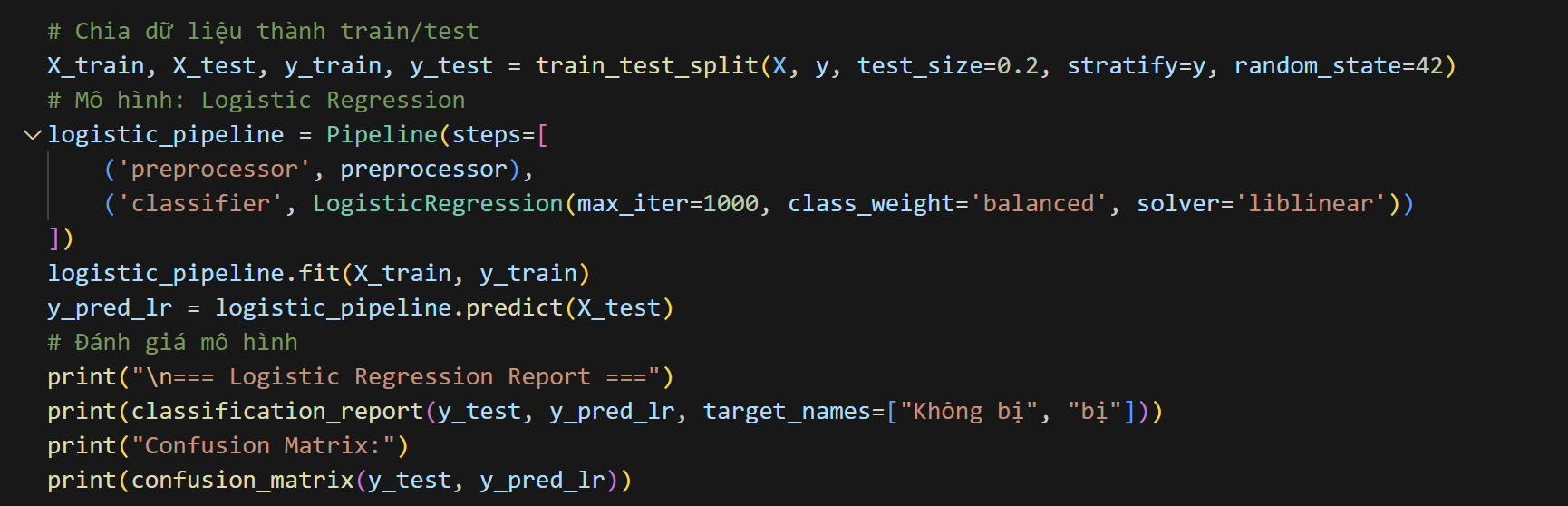
## **III. Logistic Regression**

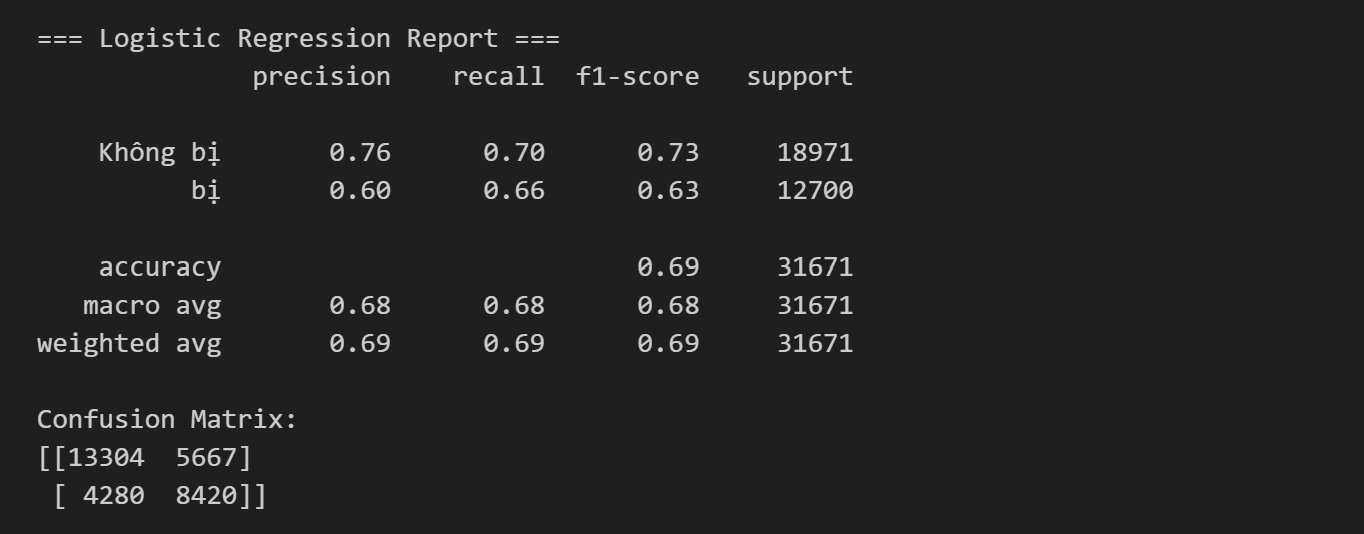
### **1. Cơ sở lý thuyết**

**Logistic Regression** là một thuật toán phân loại sử dụng hàm sigmoid để ánh xạ đầu vào thành xác suất thuộc một lớp. Mô hình tối ưu hóa hàm mất mát log-loss để tìm các tham số tốt nhất. Nó hoạt động tốt với dữ liệu có ranh giới phân tách tuyến tính và dễ triển khai.

### **2. Huấn luyện mô hình**

code mô tả:

****

kết quả thu được:

**3. Nhận xét:**

Mô hình Logistic Regression dự đoán có một sự tương quan nhất định với dữ liệu thực tế, với độ chính xác tổng thể đạt 68.58%. Tuy nhiên, vẫn còn khá nhiều sai lệch giữa dự đoán và giá trị thực tế, đặc biệt ở lớp "Bị" (dự đoán bị bệnh), nơi mô hình bỏ sót 33.70% ca (4280 mẫu) và có 29.86% báo động sai ở lớp "Không bị" (5667 mẫu). Phân bố dự đoán cũng lệch so với thực tế, với xu hướng dự đoán nhiều "Bị" hơn (44.5% so với 40.1% thực tế), nhưng không cải thiện đáng kể độ chính xác cho lớp này. Trong bối cảnh bài toán y tế nhạy cảm như dự đoán có bị bệnh không, số lượng ca bị bỏ sót (4280) là rất đáng lo ngại, vì có thể dẫn đến hậu quả nghiêm trọng.