

**ĐẠI HỌC QUỐC GIA HÀ NỘI**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**  
o0o



**BÁO CÁO HỌC PHẦN**  
**HỌC MÁY**

**Đề tài: Phân tích và dự đoán nguy cơ mắc bệnh Alzheimer**

**Nhóm thực hiện**  
Hồ Huyền Trang - 23001565  
Trần Thị Mai Anh - 23001497  
Đỗ Thị Như Quỳnh - 23001556

**Lớp: K68A5 Khoa học dữ liệu**

**HÀ NỘI - 2025**

# Mục lục

<b>1. TỔNG QUAN</b>	<b>6</b>
1.1 Giới thiệu bài toán . . . . .	6
1.2 Mục tiêu . . . . .	6
<b>2. THÔNG TIN NHÓM</b>	<b>8</b>
2.1 Thành viên nhóm . . . . .	8
2.2 Phân chia công việc . . . . .	8
<b>3. CƠ SỞ LÝ THUYẾT</b>	<b>9</b>
3.1 K-nearest neighbor . . . . .	9
3.2 Hồi quy Softmax (Softmax Regression) . . . . .	13
<b>4. TIỀN XỬ LÝ DỮ LIỆU</b>	<b>18</b>
4.1 Giới thiệu . . . . .	18
4.2 Đọc hiểu dữ liệu (Data Understanding) . . . . .	18
4.3 Phân tích thành phần và miền giá trị (Feature Domain Analysis) . . . . .	18
4.4 Làm sạch dữ liệu (Data Cleaning) . . . . .	19
4.4.1 Xử lý giá trị khuyết (Missing Values) . . . . .	19
4.4.2 Xử lý trùng lặp (Duplicates) . . . . .	19
4.4.3 Chuẩn hóa và mã hóa dữ liệu . . . . .	19
4.5 Phát hiện và xử lý ngoại lệ (Outlier Detection) . . . . .	19
4.6 Trực quan hóa dữ liệu (Data Visualization) . . . . .	20
4.7 Phân tích quan hệ giữa đặc trưng và mục tiêu (Feature–Target Relationship) . . . . .	20
4.8 Phân tích tương quan (Correlation Analysis) . . . . .	20
4.9 Chuẩn hóa và Biến đổi Dữ liệu (Data Transformation) . . . . .	20
4.9.1 Scaling . . . . .	20
4.9.2 Biến đổi đặc trưng (Feature Engineering) . . . . .	21

4.9.3 Giảm chiều dữ liệu . . . . .	21
4.10 Phân tích Thống kê Mô tả (Descriptive Statistical Analysis) . . . . .	21
4.11 Dánh giá Chất lượng Dữ liệu (Data Quality Assessment) . . . . .	21
4.11.1 Phát hiện dữ liệu không hợp lệ . . . . .	21
4.11.2 Phát hiện inconsistency . . . . .	21
4.11.3 Gắn cờ dữ liệu nghi ngờ . . . . .	21
4.12 Phân tích Mối quan hệ Phức tạp (Complex Relationship Analysis)	22
4.13 Chuẩn bị Dữ liệu cho Mô hình (Model Readiness) . . . . .	22
4.13.1 Xử lý dữ liệu mất cân bằng . . . . .	22
4.13.2 Chia dữ liệu . . . . .	22
4.13.3 Kiểm tra rò rỉ dữ liệu (Data Leakage) . . . . .	22
4.14 Trực quan hóa Nâng cao (Advanced Visualization) . . . . .	22
4.15 Kết luận và Đề xuất . . . . .	23
4.16 Thực hiện trên bộ dữ liệu Alzheimer . . . . .	23
4.16.1 Thông tin về bộ dữ liệu . . . . .	23
4.16.2 Tiền xử lý và khám phá dữ liệu . . . . .	25
<b>5. GIẢM CHIỀU VÀ TRỰC QUAN</b>	<b>35</b>
5.1 Phân tích Thành phần Chính (PCA) . . . . .	35
5.1.1 Mục tiêu của PCA . . . . .	35
5.1.2 Cơ sở toán học của PCA . . . . .	35
5.1.3 Diễn giải kết quả PCA . . . . .	36
5.2 Phân tích Phân biệt Tuyến tính (LDA) . . . . .	36
5.2.1 Mục tiêu của LDA . . . . .	36
5.2.2 Cơ sở toán học của LDA . . . . .	36
5.2.3 So sánh PCA và LDA . . . . .	37
5.2.4 Kết luận . . . . .	37
5.3 Giảm chiều bằng UMAP (Uniform Manifold Approximation and Projection) . . . . .	37
5.3.1 Giới thiệu . . . . .	37
5.3.2 Nguyên lý hoạt động . . . . .	37
5.3.3 Các bước chính . . . . .	38
5.3.4 Hàm mất mát của UMAP . . . . .	38
5.3.5 Đặc điểm nổi bật của UMAP . . . . .	38
5.3.6 So sánh PCA, LDA và UMAP . . . . .	39
5.4 Thực hành trên tập Alzheimer . . . . .	39
5.4.1 Chuẩn hóa dữ liệu . . . . .	39
5.4.2 PCA . . . . .	39

5.4.3	LDA . . . . .	44
5.4.4	UMAP . . . . .	45
5.4.5	Tổng kết . . . . .	47
<b>6.</b>	<b>KẾT QUẢ THỰC NGHIỆM</b>	<b>50</b>
6.1	K-nearest neighbor . . . . .	50
6.1.1	Trên tập dữ liệu gốc . . . . .	50
6.1.2	Trên tập dữ liệu đã giảm chiều PCA . . . . .	52
6.1.3	Trên tập dữ liệu đã giảm chiều LDA . . . . .	54
6.1.4	Tổng kết trên tập dữ liệu gốc và dữ liệu giảm chiều . . . . .	55
6.2	Softmax regression . . . . .	56
6.2.1	Trên tập dữ liệu gốc . . . . .	56
6.2.2	Trên tập dữ liệu đã giảm chiều PCA . . . . .	57
6.2.3	Trên tập dữ liệu đã giảm chiều LDA . . . . .	58
6.2.4	Tổng kết trên tập dữ liệu gốc và dữ liệu giảm chiều . . . . .	60
6.3	So sánh hai mô hình KNN và Softmax . . . . .	60
<b>7.</b>	<b>TỔNG KẾT</b>	<b>62</b>
<b>LỜI CẢM ƠN</b>		<b>63</b>
<b>Tài liệu tham khảo</b>		<b>64</b>

# Danh sách hình vẽ

3.1	Cách hoạt động của mô hình K-nearest neighbor . . . . .	10
4.1	Phân bố của biến mục tiêu. . . . .	28
4.2	Phân bố của các biến về vấn đề sức khỏe. . . . .	29
4.3	Phân bố của các biến phân loại khác. . . . .	29
4.4	Phân bố của các biến liên tục. . . . .	30
4.5	Trung bình biến liên tục với từng phân lớp mục tiêu. . . . .	31
4.6	Phân bố của biến mục tiêu với BehavioralProblems và FamilyHistory.	32
4.7	Phân bố của biến mục tiêu với MemoryComplaints và Hypertension.	32
4.8	Phân bố của các phân lớp mục tiêu trong từng nhóm học vấn. . .	32
4.9	Phân bố của các phân lớp mục tiêu trong từng nhóm dân tộc. . .	33
4.10	Xếp thứ tự tương quan giữa các biến số. . . . .	33
4.11	Ma trận tương quan giữa các biến số. . . . .	34
5.1	Scree plot. . . . .	41
5.2	Plot dữ liệu theo PC1, PC2. . . . .	41
5.3	Biplot. . . . .	42
5.4	Plot từng cặp PCs. . . . .	43
5.5	LDA . . . . .	44
5.6	Unsupervised UMAP 2D . . . . .	46
5.7	Unsupervised UMAP 3D . . . . .	46
5.8	Supervised UMAP 2D . . . . .	47

# TỔNG QUAN

## 1.1 Giới thiệu bài toán

Bệnh Alzheimer là nguyên nhân hàng đầu gây sa sút trí tuệ, ảnh hưởng nghiêm trọng đến sức khỏe cộng đồng và gánh nặng kinh tế. Việc chẩn đoán sớm có ý nghĩa sống còn để can thiệp kịp thời, làm chậm tiến trình bệnh và cải thiện chất lượng sống cho bệnh nhân.

Với sự phát triển của công nghệ và khả năng thu thập dữ liệu y tế toàn diện từ những thông số lâm sàng, lỗi sống đến kết quả xét nghiệm. Học máy trở thành công cụ đắc lực giúp phát hiện các mẫu phức tạp và mối quan hệ giữa các yếu tố nguy cơ mà con người khó nhận ra được. Từ đó xây dựng các mô hình dự đoán không xâm lấn mang lại hiệu quả và độ chính xác cao.

Trong bài báo cáo này, chúng em tập trung vào việc phân tích, chẩn đoán sớm để can thiệp kịp thời, làm chậm tiến trình bệnh, sử dụng bộ dữ liệu "alzheimers\_disease\_data" thu thập từ Kaggle. Trong bộ dữ liệu gồm các nhóm tính năng về nhân khẩu học và lối sống, lịch sử bệnh và nguy cơ, đánh giá chức năng và nhận thức, chỉ số sinh học.

Qua đó giúp chúng em xây dựng mô hình chuẩn đoán với 2 mức độ (Không mắc bệnh - mắc bệnh ở mức độ nhẹ: "not sick - mildly sick", Mắc bệnh: "sick"). Mô hình này không chỉ mang lại cái nhìn sâu sắc và khách quan về các yếu tố cốt lõi ảnh hưởng đến việc chẩn đoán bệnh của bệnh nhân, mà còn đóng góp vào việc làm chậm tiến trình bệnh và cải thiện chất lượng sống cho bệnh nhân.

## 1.2 Mục tiêu

Mục tiêu chính của báo cáo này là xây dựng và triển khai mô hình phân loại dựa trên bộ dữ liệu "alzheimers\_disease\_data" để chẩn đoán bệnh. Cụ thể, chúng em đặt ra các mục tiêu sau:

- **Phân tích dữ liệu:** Khám phá và phân tích các yếu tố ảnh hưởng đến dự đoán mắc bệnh của bệnh nhân, bao gồm nhân khẩu học và lối sống, lịch sử bệnh và nguy cơ, đánh giá chức năng và nhận thức, chỉ số sinh học.
- **Tiền xử lý dữ liệu:** Áp dụng các kỹ thuật tiền xử lý dữ liệu để xử lý các giá trị thiếu, giảm chiều dữ liệu, chuẩn hóa và mã hóa các thuộc tính cần

thiết, giúp tối ưu hóa quá trình huấn luyện mô hình học máy.

- **Xây dựng mô hình dự đoán:** Xây dựng mô hình phân loại sử dụng các thuật toán học máy như Softmax Regression, K-Nearest Neighbors để dự đoán mắc bệnh của bệnh nhân. Đánh giá và lựa chọn mô hình có hiệu quả dự đoán tốt nhất dựa trên các chỉ số như accuracy, precision, recall và F1-score.
- **Phân tích yếu tố quyết định:** Phân tích và xác định các yếu tố quan trọng nhất ảnh hưởng đến việc bệnh nhân mắc bệnh thông qua việc đánh giá các trọng số của các thuộc tính trong mô hình dự đoán.

Chúng em kỳ vọng báo cáo này sẽ cung cấp những hiểu biết sâu sắc về cách thức ứng dụng học máy trong ngành y sinh, đồng thời đóng góp vào việc nâng cao chất lượng, để can thiệp kịp thời, làm chậm tiến trình bệnh và cải thiện chất lượng sống cho bệnh nhân.

# THÔNG TIN NHÓM

## 2.1 Thành viên nhóm

Nhóm gồm ba thành viên:

- Hồ Huyền Trang - 23001565
- Trần Thị Mai Anh - 23001497
- Đỗ Thị Như Quỳnh - 23001556

## 2.2 Phân chia công việc

Với mong muốn tạo ra một môi trường làm việc thật chuyên nghiệp, công bằng và mang lại kết quả tốt nhất, nhóm đã phân chia đều các công việc cho các thành viên, mỗi thành viên sẽ phụ trách một mảng chính của đề tài. Trong quá trình làm việc nếu gặp khó khăn, cả nhóm sẽ thảo luận và đưa ra các phương án để giải quyết. Cụ thể như sau:

- Hồ Huyền Trang: Phân tích dữ liệu và áp dụng giảm chiều PCA.
- Trần Thị Mai Anh: Lý thuyết và mô hình K-Nearest Neighbors
- Đỗ Thị Như Quỳnh: Lý thuyết và mô hình Logistic Softmax

Ngoài ra các thành viên trong nhóm cùng nhau chia sẻ, trao đổi đưa ra dự đoán về nguy cơ mắc bệnh Alzheimer, viết báo cáo và làm slide.

# CƠ SỞ LÝ THUYẾT

## 3.1 K-nearest neighbor

K-nearest neighbor (KNN) là một trong những thuật toán thuộc nhóm học giám sát, nổi tiếng với sự đơn giản. Trong giai đoạn huấn luyện, KNN không thực hiện bất kỳ việc học nào từ dữ liệu huấn luyện mà nhớ lại một cách máy móc toàn bộ dữ liệu đó; toàn bộ quá trình tính toán chỉ diễn ra khi cần dự đoán đầu ra cho dữ liệu mới. Thuật toán này có thể được áp dụng cho cả hai loại bài toán phân loại và hồi quy. KNN còn được gọi là một thuật toán dựa theo mẫu hay thuật toán dựa theo trí nhớ.

Trong KNN, đối với bài toán phân loại, nhãn của một dữ liệu mới được xác định dựa trên K điểm dữ liệu gần nhất trong tập huấn luyện. Nhãn đó có thể được quyết định thông qua phương pháp bầu chọn đa số (major voting) trong số K điểm gần nhất, hoặc nó có thể được suy ra bằng cách đánh trọng số khác nhau cho mỗi trong các điểm gần nhất đó rồi suy ra kết quả.

Đối với bài toán hồi quy, kết quả đầu ra của một điểm dữ liệu mới sẽ được xác định bằng giá trị của điểm dữ liệu gần nhất trong tập huấn luyện (khi K=1), hoặc được tính dựa trên giá trị trung bình có trọng số của các điểm lân cận, hoặc thông qua một mối quan hệ phụ thuộc vào khoảng cách tới các điểm gần nhất.

Cũng đáng lưu ý rằng thuật toán KNN cũng là một phần của họ các mô hình "học lười biếng", nghĩa là nó chỉ lưu trữ một tập dữ liệu đào tạo so với việc trải qua một giai đoạn đào tạo. Điều này cũng có nghĩa là tất cả các phép tính diễn ra khi phân loại hoặc dự đoán đang được thực hiện. Vì nó phụ thuộc rất nhiều vào bộ nhớ để lưu trữ tất cả dữ liệu đào tạo của nó, nên nó cũng được gọi là phương pháp học dựa trên thể hiện hoặc dựa trên bộ nhớ.

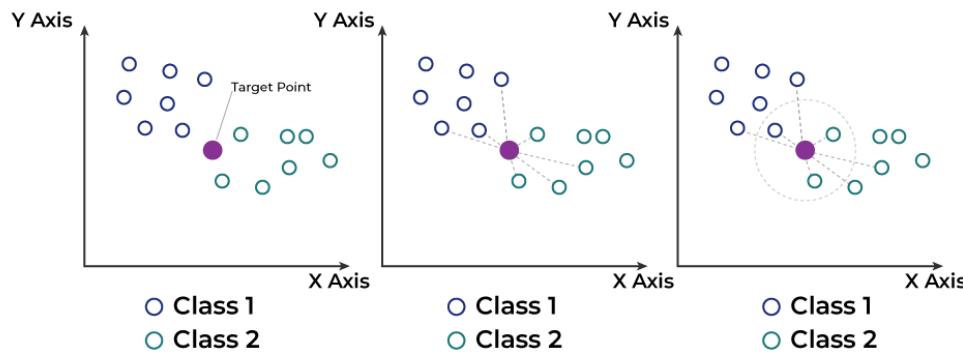
Tóm lại, KNN là thuật toán đi tìm đầu ra cho một điểm dữ liệu mới bằng cách dựa trên thông tin từ K điểm gần nhất trong tập huấn luyện, mà không xét đến việc một số điểm trong đó có thể là nhiễu.

Mặc dù không còn phổ biến như trước đây, nhưng đây vẫn là một trong những thuật toán đầu tiên mà người ta học trong khoa học dữ liệu do tính đơn giản và chính xác của nó. Tuy nhiên, khi tập dữ liệu phát triển, KNN ngày càng kém hiệu quả, làm giảm hiệu suất chung của mô hình. Nó thường được sử dụng cho các hệ thống đề xuất đơn giản, nhận dạng mẫu, khai thác dữ liệu, dự đoán thị

trường tài chính, phát hiện xâm nhập, v.v.

#### \* Cách KNN hoạt động

Thuật toán K-Nearest Neighbors (KNN) hoạt động dựa trên một nguyên tắc đơn giản: dự đoán nhãn hoặc giá trị của một điểm dữ liệu mới bằng cách tham khảo nhãn hoặc giá trị của K điểm lân cận gần nhất trong tập huấn luyện.



Hình 3.1: Cách hoạt động của mô hình K-nearest neighbor

- **Bước 1:** Lựa chọn giá trị thích hợp cho K

K là số lượng điểm lân cận gần nhất được sử dụng để đưa ra dự đoán kết quả. Ví dụ, nếu K = 3, thuật toán sẽ tìm 3 điểm gần nhất để quyết định nhãn hoặc giá trị cho điểm dữ liệu mới.

- **Bước 2:** Tính toán khoảng cách

Để xác định điểm dữ liệu nào gần nhất với điểm truy vấn nhất định, khoảng cách giữa điểm truy vấn và các điểm dữ liệu khác sẽ cần được tính toán. Các số liệu khoảng cách này giúp hình thành ranh giới quyết định, phân chia các điểm truy vấn thành các vùng khác nhau, các khoảng cách phổ biến được sử dụng bao gồm:

#### Khoảng cách Euclidean (p=2)

Khoảng cách Euclid, thường được coi là số liệu trực quan nhất, tính toán khoảng cách đường thẳng giữa hai điểm dữ liệu trong không gian đa chiều. Công thức của nó, bắt nguồn từ định lý Pythagore, rất đơn giản:

$$d_{\text{Euclidean}}(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

trong đó:

- $x$  và  $y$  là hai điểm dữ liệu.
- $x_i$  và  $y_i$  là các thành phần tại chiều thứ  $i$ .
- $n$  là số chiều của không gian dữ liệu.

### Khoảng cách Manhattan (p=1)

Khoảng cách Manhattan, còn được gọi là khoảng cách taxi, đo khoảng cách giữa hai điểm bằng cách cộng các chênh lệch tuyệt đối của tọa độ của chúng theo từng chiều. Trong hai chiều, hãy hình dung nó như việc di chuyển trong một khối thành phố dạng lưới - bạn chỉ có thể di chuyển theo chiều ngang hoặc chiều dọc.

Khoảng cách Manhattan trở nên đặc biệt hữu ích trong các không gian chiều cao thừa thớt dân cư như được mô tả bởi Aggarwal và cộng sự (“Về hành vi đáng ngạc nhiên của các phép đo khoảng cách trong không gian chiều cao,” 2001), hoặc nếu các đặc điểm có tỷ lệ khác biệt đáng kể, trong đó việc bình phương các điểm khác biệt có xu hướng khuếch đại khoảng cách do các số lớn hơn gây ra (hãy tưởng tượng một đặc điểm có giá trị từ 0 đến 10, so với đặc điểm có giá trị từ 0 đến 1).

$$d_{\text{Manhattan}}(x, y) = \sum_{i=1}^n |x_i - y_i|$$

### Khoảng cách Minkowski

Đo khoảng cách này là dạng tổng quát của các số liệu khoảng cách Euclidean và Manhattan. Tham số  $p$  trong công thức bên dưới cho phép tạo ra các số liệu khoảng cách khác.

$$d_{\text{Minkowski}}(x, y) = \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$$

- Khi  $p = 1$ , khoảng cách Minkowski trở thành khoảng cách Manhattan.
- Khi  $p = 2$ , khoảng cách Minkowski trở thành khoảng cách Euclidean.
- $p$  là một tham số tự do.

- **Bước 3:** Xác định K hàng xóm gần nhất

$K$  điểm dữ liệu có khoảng cách nhỏ nhất đến điểm mục tiêu được chọn làm các hàng xóm gần nhất. Nhìn chung, giá trị  $K$  nhỏ có thể dẫn đến kết quả quá nhạy cảm với nhiễu trong dữ liệu, trong khi giá trị  $K$  lớn có thể không nắm bắt được những hàng xóm có gần nhất.

Một số kỹ thuật có thể giúp xác định giá trị  $K$  tối ưu cho các ứng dụng tìm kiếm vectơ, bao gồm:

- **Cross-Validation:** Trong bối cảnh này, cross-validation liên quan đến việc phân vùng tập dữ liệu thành nhiều tập con. Hiệu suất của thuật toán tìm kiếm được đánh giá trên các tập con này để xác định giá trị  $K$  hàng đầu cân bằng giữa độ chính xác và khả năng thu hồi.
- **Phương pháp Elbow:** Phương pháp này vẽ biểu đồ tỷ lệ lỗi tìm kiếm hoặc số liệu hiệu suất có liên quan so với nhiều giá trị  $K$  khác nhau. 'Điểm Elbow' trên biểu đồ, nơi tỷ lệ cải thiện bắt đầu chậm lại, thường chỉ ra lựa chọn tốt cho  $K$ . Điểm này biểu thị sự cân bằng giữa việc có quá ít và quá nhiều hàng xóm.
- **GridSearchCV:** Grid search hoạt động có hệ thống thông qua nhiều tổ hợp giá trị tham số, trong trường hợp này là các giá trị  $K$  khác nhau, để tìm ra thiết lập tham số tối ưu. Kỹ thuật này đòi hỏi nhiều tính toán nhưng có thể rất hiệu quả. Giá trị hiệu suất tốt nhất có thể được chọn bằng cách đánh giá hiệu suất của thuật toán tìm kiếm trên một phạm vi giá trị  $K$ .
- **RandomizedSearchCV:** RandomizedSearchCV tìm kiếm siêu tham số bằng cách thử nghiệm ngẫu nhiên trong không gian tham số, thay vì thử tất cả các tổ hợp như GridSearchCV. Phương pháp này giúp tiết kiệm thời gian tính toán và có thể hiệu quả hơn khi số lượng tham số lớn, đồng thời tìm ra giá trị siêu tham số tối ưu cho mô hình.

Việc sử dụng các kỹ thuật này có thể xác định giá trị  $K$  cao nhất giúp cân bằng tối ưu độ chính xác tìm kiếm và hiệu quả tính toán, do đó cải thiện hiệu suất và độ mạnh mẽ của tìm kiếm vectơ.

- **Bước 4:** Dự đoán kết quả dựa trên Phân loại hoặc Hồi quy

- Đối với bài toán **Phân loại**, nhãn của điểm mới được xác định thông qua **bỏ phiếu đa số** từ  $K$  điểm lân cận. Nhãn xuất hiện nhiều nhất sẽ là nhãn dự đoán.
- Đối với bài toán **Hồi quy**, giá trị dự đoán được tính bằng **trung bình có trọng số** của các giá trị từ  $K$  điểm lân cận gần nhất.

Thuật toán KNN, giống như bất kỳ công cụ nào khác, đều có những ưu điểm và nhược điểm riêng:

#### Những lợi ích:

- **Dễ hiểu và triển khai:** Bản chất trực quan của KNN giúp nhiều người dùng, từ người mới bắt đầu đến người có kinh nghiệm, đều có thể sử dụng được. Do tính đơn giản và chính xác của thuật toán, đây là một trong những bộ phân loại đầu tiên mà một nhà khoa học dữ liệu mới vào nghề sẽ học.
- **Dễ thích ứng:** Khi các mẫu đào tạo mới được thêm vào, thuật toán sẽ điều chỉnh để tính đến bất kỳ dữ liệu mới nào vì tất cả dữ liệu đào tạo đều được lưu trữ trong bộ nhớ.
- **Ít siêu tham số:** KNN chỉ yêu cầu giá trị  $k$  và số liệu khoảng cách  $p$ , thấp khi so sánh với các thuật toán học máy khác, giúp huấn luyện đơn giản hơn.
- **Tính linh hoạt trong việc xử lý nhiều loại dữ liệu khác nhau:** KNN hỗ trợ nhiều loại dữ liệu khác nhau, phù hợp với nhiều ứng dụng khác nhau.

#### Mặt hạn chế:

- **Tốn kém về mặt tính toán:** Việc tính toán khoảng cách giữa tất cả các điểm dữ liệu, đặc biệt là trong các tập dữ liệu lớn, có thể tốn nhiều công sức tính toán, ảnh hưởng đến tốc độ.
- **Độ nhạy với giá trị ngoại lệ:** Hiệu suất của KNN có thể bị ảnh hưởng đáng kể bởi các giá trị ngoại lệ trong dữ liệu.
- **Dễ bị quá khớp:** KNN gặp phải "lời nguyền của đa chiều", khi phải xử lý dữ liệu với quá nhiều chiều, làm giảm khả năng phân loại chính xác. Điều này cũng khiến thuật toán dễ bị quá khớp, và để khắc phục, các kỹ thuật giảm chiều và lựa chọn đặc điểm thường được áp dụng.

## 3.2 Hồi quy Softmax (Softmax Regression)

Bài toán Phân loại Đa lớp (Multiclass classification) là một trong những ứng dụng của học sâu/học máy, trong đó mô hình nhận đầu vào và đưa ra đầu ra phân loại tương ứng với một trong các nhãn trong tập các nhãn đầu ra đã có. Để thực hiện điều này, khi mô hình phải xuất ra nhiều giá trị cho mỗi lớp, hàm logistic đơn giản (hay còn gọi là hàm sigmoid) không thể được sử dụng. Thay vào đó, một hàm kích hoạt khác gọi là hàm Softmax được sử dụng cùng với hàm mất mát cross-entropy.

### Công thức của hàm Softmax

Chúng ta cần một mô hình xác suất để với mỗi đầu vào  $\mathbf{x}$ , giá trị  $a_i$  biểu diễn xác suất đầu vào thuộc về lớp  $i$ . Điều kiện cần thiết là các giá trị  $a_i$ , phải dương và tổng của chúng bằng 1. Để đảm bảo các điều kiện này, chúng ta cần quan sát giá trị  $z_i$ ; và xác định mối liên hệ giữa  $z_i$  và  $a_i$  để tính  $a_i$ .

Bên cạnh đó, ngoài việc  $a_i$  phải dương và tổng  $a_i$  bằng 1, cần thêm một điều kiện khác:  $a_i$  tăng theo  $z_i$ . Nghĩa là, nếu  $z_i$  càng lớn thì xác suất  $a_i$  càng cao hay xác suất lớp  $i$  được dự đoán cao hơn, điều này có thể đạt được bằng một hàm số đơn điệu tăng.

Với  $z_i$  có thể là bất kỳ giá trị thực nào (âm hoặc dương), một cách tự nhiên để chuyển nó thành giá trị dương, đồng thời đảm bảo tính đơn điệu, là sử dụng hàm  $e^{z_i}$ . Khi đó, để đảm bảo tổng các  $a_i$  bằng 1, ta định nghĩa:

$$a_i = \frac{e^{z_i}}{\sum_{j=1}^C e^{z_j}}, \quad \forall i = 1, 2, \dots, C$$

Hàm này, gọi là **Hàm Softmax (Softmax Function)**, tính toán tất cả các giá trị  $a_i$  từ các giá trị  $z_i$ , thỏa mãn:  $a_i > 0$ , tổng các  $a_i = 1$ , và thứ tự của  $z_i$  được bảo toàn. Lưu ý rằng, không có  $a_i$  nào tuyệt đối bằng 0 hoặc 1, trừ khi  $z_i$  rất nhỏ hoặc rất lớn so với các  $z_j$ ,  $j \neq i$ .

Khi đó, xác suất để điểm dữ liệu  $\mathbf{x}$  thuộc về lớp  $i$  được viết là:

$$P(y = i | \mathbf{x}; \mathbf{W}) = a_i$$

Trong đó,  $P(y = i | \mathbf{x}; \mathbf{W})$  là xác suất mà mô hình (với tham số  $\mathbf{W}$ ) dự đoán điểm  $\mathbf{x}$  thuộc lớp  $i$ .

### One-hot coding

Trong các mô hình mạng nơ-ron, đầu ra thường không phải là một giá trị đơn lẻ đại diện cho mỗi lớp, mà là một vector gọi là *one-hot vector*. Vector này có duy nhất một phần tử bằng 1, các phần tử còn lại bằng 0. Phần tử bằng 1 nằm ở vị trí tương ứng với lớp của mẫu đầu vào, biểu diễn rằng mẫu này chắc chắn thuộc về lớp đó (xác suất 1). Đây chính là phương pháp mã hóa **one-hot coding**. Ví dụ, với 3 lớp, lớp thứ 2 sẽ có one-hot vector là  $[0, 1, 0]$ .

Khi áp dụng mô hình Softmax Regression, với mỗi đầu vào  $\mathbf{x}$ , ta tính **đầu ra dự đoán** bằng công thức:

$$\mathbf{a} = \text{softmax}(\mathbf{W}^\top \mathbf{x})$$

Trong đó,  $\mathbf{a}$  là xác suất dự đoán cho từng lớp. Ngược lại, đầu ra thực sự (ground truth label)  $\mathbf{y}$  được biểu diễn dưới dạng one-hot vector, với giá trị đúng là 1 tại vị trí của lớp đúng và 0 ở các vị trí khác.

Hàm mất mát (loss function) được dùng để đo lường mức độ khác biệt giữa đầu ra dự đoán  $\mathbf{a}_i$  và nhãn thực sự  $\mathbf{y}_i$ . Một lựa chọn đơn giản là sử dụng khoảng cách bình phương (MSE):

$$J(\mathbf{W}) = \sum_{i=1}^N \|\mathbf{a}_i - \mathbf{y}_i\|_2^2$$

*Tuy nhiên, khi làm việc với phân phối xác suất (softmax), hàm MSE không phải lựa chọn phù hợp, do nó không phản ánh rõ mức độ khác biệt giữa hai phân phối.* Để cải thiện, ta sử dụng một đại lượng hiệu quả hơn gọi là **cross-entropy**.

### Cross-Entropy

Cross-entropy là một hàm mất mát phổ biến trong bài toán phân loại nhiều lớp, đo lường khoảng cách giữa phân phối xác suất thực tế và phân phối dự đoán. Khoảng cách giữa hai phân phối  $\mathbf{p}$  và  $\mathbf{q}$  được định nghĩa là:

$$H(\mathbf{p}, \mathbf{q}) = \mathbb{E}_{\mathbf{p}}[-\log \mathbf{q}]$$

Trong trường hợp phân phối rời rạc (như vector  $\mathbf{y}$  và  $\mathbf{a}$ ), công thức trên có thể viết lại dưới dạng:

$$H(\mathbf{p}, \mathbf{q}) = - \sum_{i=1}^C p_i \log q_i$$

*Trong bài toán phân loại,  $p_i$  thường là nhãn thực sự (dạng one-hot) và  $q_i$  là xác suất dự đoán.*

Cross-entropy giúp đo lường hiệu quả sự khác biệt giữa hai phân phối, đặc biệt hữu ích trong các bài toán phân loại. Vì vậy, cross-entropy thường được sử dụng làm hàm mất mát chính trong huấn luyện mô hình softmax regression hoặc mạng nơ-ron phân loại.

**Chú ý:** Hàm Cross-Entropy không có tính đối xứng, nghĩa là  $H(\mathbf{p}, \mathbf{q}) \neq H(\mathbf{q}, \mathbf{p})$ . Điều này có thể được giải thích qua công thức của hàm. Trong đó, các thành phần của phân phối  $\mathbf{q}$  bằng 0 sẽ dẫn tới giá trị  $\log(0)$ , vốn không xác định. Vì vậy, khi sử dụng Cross-Entropy trong bài toán học có giám sát

(supervised learning), giá trị đầu ra thực sự  $\mathbf{y}$  được biểu diễn dưới dạng one-hot vector: một phần tử có giá trị 1 (ứng với lớp đúng), còn lại bằng 0. Trong khi đó, đầu ra dự đoán  $\mathbf{q}$  là phân phối xác suất dự đoán của mô hình, giá trị này không bao giờ tuyệt đối bằng 0 hay 1.

Trong **Logistic Regression**, đầu ra dự đoán là một phân phối đơn giản:

- **Đầu ra thực sự** của điểm dữ liệu đầu vào  $\mathbf{x}_i$  có phân phối xác suất là  $[y_i; 1 - y_i]$ , với  $y_i$  là xác suất để đầu vào rơi vào lớp thứ nhất (bằng 1 nếu đúng, bằng 0 nếu sai).
- **Đầu ra dự đoán** là phân phối xác suất dự đoán:  $\hat{y}_i = \text{sigmoid}(\mathbf{w}^\top \mathbf{x}_i)$ , biểu diễn xác suất dự đoán rơi vào lớp thứ nhất. Do đó, xác suất dự đoán rơi vào lớp thứ hai là  $1 - \hat{y}_i$ .

Với cách định nghĩa này, hàm mất mát của Logistic Regression được viết dưới dạng:

$$J(\mathbf{w}) = - \sum_{i=1}^N \left( y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right)$$

Hàm mất mát này là một trường hợp đặc biệt của cross-entropy khi số lớp  $C = 2$ , trong đó  $N$  là số lượng điểm dữ liệu trong tập huấn luyện.

Khi số lớp  $C > 2$ , mô hình logistic được mở rộng thành Softmax Regression, trong đó mỗi điểm đầu ra là một vector xác suất có tổng bằng 1. Hàm mất mát giữa đầu ra dự đoán  $\mathbf{a}_i$  và đầu ra thực sự  $\mathbf{y}_i$  của một điểm dữ liệu được định nghĩa như sau:

$$J(\mathbf{W}; \mathbf{x}_i, \mathbf{y}_i) = - \sum_{j=1}^C y_{ij} \log(a_{ij})$$

Trong đó:

- $y_{ij}$  là thành phần thứ  $j$  trong vector  $\mathbf{y}_i$  (xác suất thực sự),
- $a_{ij}$  là thành phần thứ  $j$  trong vector  $\mathbf{a}_i$  (xác suất dự đoán),
- $\mathbf{W}$  là ma trận trọng số của mô hình. Nếu nhãn thực sự là one-hot, chỉ phần tử ứng với lớp đúng là  $y_{ij} = 1$ , do đó chỉ có một phần tử đóng góp vào hàm mất mát.

### Hàm mất mát tổng quát cho Softmax Regression

Với tập dữ liệu gồm  $N$  cặp đầu vào  $(\mathbf{x}_i, \mathbf{y}_i)$  với  $i = 1, 2, \dots, N$ , hàm mất mát tổng quát cho Softmax Regression được tính như sau:

$$J(\mathbf{W}; \mathbf{X}, \mathbf{Y}) = - \sum_{i=1}^N \sum_{j=1}^C y_{ij} \log(a_{ij})$$

Khi thay  $\mathbf{a}_i$  bằng biểu thức Softmax, ta có:

$$J(\mathbf{W}; \mathbf{X}, \mathbf{Y}) = - \sum_{i=1}^N \sum_{j=1}^C y_{ij} \log \left( \frac{e^{\mathbf{w}_j^\top \mathbf{x}_i}}{\sum_{k=1}^C e^{\mathbf{w}_k^\top \mathbf{x}_i}} \right)$$

# TIỀN XỬ LÝ DỮ LIỆU VÀ PHÂN TÍCH DỮ LIỆU

## 4.1 Giới thiệu

Tiền xử lý dữ liệu (*Data Preprocessing*) là giai đoạn đầu tiên và quan trọng nhất trong quy trình phân tích dữ liệu và học máy. Mục tiêu chính là:

- Hiểu rõ cấu trúc, ý nghĩa và chất lượng dữ liệu.
- Làm sạch, chuẩn hóa và biến đổi dữ liệu để phù hợp cho mô hình.
- Giảm nhiễu, xử lý dữ liệu thiếu, trùng lặp hoặc bất thường.

## 4.2 Đọc hiểu dữ liệu (Data Understanding)

- Kiểm tra kích thước dữ liệu: số dòng (samples), số cột (features).
- Xác định kiểu dữ liệu: số (*numeric*), chuỗi (*string*), danh mục (*categorical*).
- Thống kê mô tả cơ bản:

mean, std, min, max, percentile

- Kiểm tra phân phối và sự mất cân bằng của biến mục tiêu (target).

## 4.3 Phân tích thành phần và miền giá trị (Feature Domain Analysis)

- Xác định loại biến: định lượng, định tính, nhị phân, thứ tự, v.v.
- Kiểm tra miền giá trị hợp lệ (domain): giá trị nhỏ nhất, lớn nhất, phạm vi hợp lý.
- Phát hiện giá trị hiếm, sai phạm, hoặc nằm ngoài phạm vi nghiệp vụ.

## 4.4 Làm sạch dữ liệu (Data Cleaning)

### 4.4.1 Xử lý giá trị khuyết (Missing Values)

- Tính tỷ lệ thiếu dữ liệu:

$$\text{Missing Rate} = \frac{\text{số giá trị thiếu}}{\text{tổng số quan sát}}$$

- Chiến lược xử lý:

- Loại bỏ cột/dòng có quá nhiều giá trị khuyết.
- Diền trung bình, trung vị, hoặc mode.
- Dự đoán giá trị thiếu bằng mô hình thống kê.

### 4.4.2 Xử lý trùng lặp (Duplicates)

- Phát hiện dòng trùng lặp hoàn toàn hoặc gần giống.
- Giữ lại bản ghi duy nhất, loại bỏ bản sao.

### 4.4.3 Chuẩn hóa và mã hóa dữ liệu

- Chuẩn hóa dữ liệu số: z-score, min-max scaling, log-transform.
- Mã hóa dữ liệu phân loại:
  - One-hot encoding.
  - Label encoding.

## 4.5 Phát hiện và xử lý ngoại lệ (Outlier Detection)

- Quy tắc 3-sigma:

$$|x - \mu| > 3\sigma$$

- Quy tắc IQR:

$$\text{IQR} = Q_3 - Q_1, \quad x < Q_1 - 1.5 \times \text{IQR} \text{ hoặc } x > Q_3 + 1.5 \times \text{IQR}$$

- Phương pháp mô hình: Isolation Forest, LOF, DBSCAN.

## 4.6 Trực quan hóa dữ liệu (Data Visualization)

- Histogram: hiển thị phân phối giá trị.
- Boxplot: nhận diện ngoại lệ.
- Pairplot/Scatter matrix: quan sát mối quan hệ giữa các đặc trưng.
- Heatmap: hiển thị ma trận tương quan giữa các biến.

## 4.7 Phân tích quan hệ giữa đặc trưng và mục tiêu (Feature–Target Relationship)

- Hồi quy: dùng Pearson, Spearman hoặc kiểm định t.
- Phân loại: dùng ANOVA hoặc Chi-square để đo ảnh hưởng của đặc trưng đến target.
- Trực quan hóa bằng violin plot, boxplot hoặc bar chart.

## 4.8 Phân tích tương quan (Correlation Analysis)

- Ma trận tương quan:

$$r_{ij} = \frac{\text{Cov}(X_i, X_j)}{\sigma_i \sigma_j}$$

- $r_{ij} \in [-1, 1]$ : giá trị gần 1 là tương quan mạnh.
- Loại bỏ các đặc trưng tương quan cao để tránh đa cộng tuyến.

## 4.9 Chuẩn hóa và Biến đổi Dữ liệu (Data Transformation)

### 4.9.1 Scaling

- Standardization:

$$x' = \frac{x - \mu}{\sigma}$$

- Min–Max Scaling:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

- Robust Scaling: dùng trung vị và IQR.

#### 4.9.2 Biến đổi đặc trưng (Feature Engineering)

- Tạo đặc trưng tổng hợp, tỉ lệ, hoặc chênh lệch giữa các cột.
- Trích xuất thuộc tính thời gian (ngày, tháng, năm, mùa).
- Sinh biến nhị phân dựa theo điều kiện nghiệp vụ.

#### 4.9.3 Giảm chiều dữ liệu

- PCA, t-SNE, UMAP cho trực quan hóa.
- Feature selection dựa trên thông tin tương hỗ (Mutual Information).

### 4.10 Phân tích Thống kê Mô tả (Descriptive Statistical Analysis)

- Kiểm tra độ lệch (skewness) và độ nhọn (kurtosis).
- Kiểm định phân phối chuẩn (Shapiro–Wilk, KS test).
- Thống kê trung bình, trung vị, tứ phân vị theo từng nhóm dữ liệu.

### 4.11 Đánh giá Chất lượng Dữ liệu (Data Quality Assessment)

#### 4.11.1 Phát hiện dữ liệu không hợp lệ

- Kiểm tra logic (ví dụ: tuổi > 0).
- Đối chiếu domain nghiệp vụ.

#### 4.11.2 Phát hiện inconsistency

- Chuẩn hóa định dạng chuỗi, xóa ký tự đặc biệt.
- Gộp các giá trị trùng nghĩa (“Hà Nội”, “Ha Noi”).

#### 4.11.3 Gắn cờ dữ liệu nghi ngờ

- Tạo cột `is_suspect` = 1 cho quan sát có vấn đề.

## 4.12 Phân tích Mối quan hệ Phức tạp (Complex Relationship Analysis)

- Sử dụng Spearman hoặc Kendall cho tương quan phi tuyến.
- Trực quan hóa scatter + đường hồi quy.
- Phân tích đa biến (multivariate): heatmap, cluster analysis, factor analysis.

## 4.13 Chuẩn bị Dữ liệu cho Mô hình (Model Readiness)

### 4.13.1 Xử lý dữ liệu mất cân bằng

- Oversampling (SMOTE), undersampling.
- Gán trọng số lớp (class weighting).

### 4.13.2 Chia dữ liệu

- Chia train/test theo tỷ lệ (70/30 hoặc 80/20).
- Dùng stratified sampling để bảo toàn phân phối lớp.
- Áp dụng cross-validation để đánh giá ổn định.

### 4.13.3 Kiểm tra rò rỉ dữ liệu (Data Leakage)

- Đảm bảo dữ liệu test không bị ảnh hưởng bởi tiền xử lý hoặc chọn đặc trưng.

## 4.14 Trực quan hóa Nâng cao (Advanced Visualization)

- Heatmap có hiển thị giá trị tương quan.
- Pairplot phân lớp theo target.
- Violinplot, swarmplot để so sánh phân phối.
- Biểu đồ phân phối so sánh giữa các nhóm dữ liệu.

## 4.15 Kết luận và Đề xuất

- Tổng hợp các vấn đề dữ liệu phát hiện được trong quá trình phân tích.
- Đề xuất hướng xử lý bổ sung: thu thập thêm dữ liệu, chọn đặc trưng, loại bỏ nhiễu.
- Cung cấp insight hỗ trợ xây dựng và huấn luyện mô hình học máy hiệu quả hơn.

## 4.16 Thực hiện trên bộ dữ liệu Alzheimer

### 4.16.1 Thông tin về bộ dữ liệu

Bộ dữ liệu được sử dụng trong dự án bao gồm thông tin về bệnh nhân, yếu tố lối sống, tiền sử y tế, các chỉ số lâm sàng, đánh giá nhận thức – chức năng và triệu chứng liên quan đến bệnh Alzheimer. Cụ thể như sau:

#### Thông tin bệnh nhân

- **PatientID:** Mã định danh duy nhất được gán cho từng bệnh nhân (từ 4751 đến 6900).

#### Thông tin nhân khẩu học

- **Tuổi (Age):** Độ tuổi của bệnh nhân, dao động từ 60 đến 90.
- **Giới tính (Gender):** 0 biểu thị Nam, 1 biểu thị Nữ.
- **Chủng tộc (Ethnicity):**
  - 0: Người da trắng (Caucasian)
  - 1: Người Mỹ gốc Phi (African American)
  - 2: Người châu Á (Asian)
  - 3: Khác (Other)
- **Trình độ học vấn (EducationLevel):**
  - 0: Không có
  - 1: Trung học phổ thông
  - 2: Cử nhân
  - 3: Sau đại học

## Yếu tố lối sống

- **BMI:** Chỉ số khối cơ thể, từ 15 đến 40.
- **Hút thuốc (Smoking):** 0 là Không, 1 là Có.
- **Uống rượu (AlcoholConsumption):** Lượng tiêu thụ rượu hàng tuần (đơn vị), từ 0 đến 20.
- **Hoạt động thể chất (PhysicalActivity):** Số giờ hoạt động thể chất mỗi tuần, từ 0 đến 10.
- **Chất lượng chế độ ăn (DietQuality):** Điểm đánh giá chất lượng chế độ ăn, từ 0 đến 10.
- **Chất lượng giấc ngủ (SleepQuality):** Điểm đánh giá chất lượng giấc ngủ, từ 4 đến 10.

## Tiền sử y tế

- **FamilyHistoryAlzheimers:** Tiền sử gia đình mắc bệnh Alzheimer (0: Không, 1: Có).
- **CardiovascularDisease:** Bệnh tim mạch (0: Không, 1: Có).
- **Diabetes:** Bệnh tiểu đường (0: Không, 1: Có).
- **Depression:** Trầm cảm (0: Không, 1: Có).
- **HeadInjury:** Tiền sử chấn thương đầu (0: Không, 1: Có).
- **Hypertension:** Tăng huyết áp (0: Không, 1: Có).

## Chỉ số lâm sàng

- **SystolicBP:** Huyết áp tâm thu (90–180 mmHg).
- **DiastolicBP:** Huyết áp tâm trương (60–120 mmHg).
- **CholesterolTotal:** Tổng lượng cholesterol (150–300 mg/dL).
- **CholesterolLDL:** Mức cholesterol LDL (50–200 mg/dL).
- **CholesterolHDL:** Mức cholesterol HDL (20–100 mg/dL).
- **CholesterolTriglycerides:** Mức triglyceride (50–400 mg/dL).

## Đánh giá nhận thức và chức năng

- **MMSE:** Điểm kiểm tra Mini-Mental State Examination (0–30). Điểm thấp hơn cho thấy suy giảm nhận thức.
- **FunctionalAssessment:** Điểm đánh giá chức năng (0–10). Điểm thấp hơn thể hiện suy giảm nghiêm trọng hơn.
- **MemoryComplaints:** Có phàn nàn về trí nhớ hay không (0: Không, 1: Có).
- **BehavioralProblems:** Có vấn đề hành vi hay không (0: Không, 1: Có).
- **ADL:** Điểm đánh giá hoạt động sinh hoạt hàng ngày (0–10). Điểm thấp hơn biểu thị suy giảm chức năng.

## Triệu chứng

- **Confusion:** Có tình trạng lú lẫn (0: Không, 1: Có).
- **Disorientation:** Có tình trạng mất định hướng (0: Không, 1: Có).
- **PersonalityChanges:** Có thay đổi tính cách (0: Không, 1: Có).
- **DifficultyCompletingTasks:** Khó khăn khi hoàn thành công việc (0: Không, 1: Có).
- **Forgetfulness:** Hay quên (0: Không, 1: Có).

## Thông tin chẩn đoán

- **Diagnosis:** Tình trạng chẩn đoán bệnh Alzheimer (0: Không, 1: Có).

## Thông tin bảo mật

- **DoctorInCharge:** Thông tin bí mật về bác sĩ phụ trách, với giá trị mặc định là “XXXConfid” cho tất cả bệnh nhân.

### 4.16.2 Tiền xử lý và khám phá dữ liệu

Trong phần này, quá trình tiền xử lý và phân tích khám phá dữ liệu (EDA) được thực hiện nhằm hiểu rõ hơn về đặc điểm của bộ dữ liệu, phát hiện các giá trị bất thường, mối tương quan giữa các biến, và mối liên hệ với biến mục tiêu (chẩn đoán bệnh Alzheimer).

## 1. Tìm hiểu và làm sạch dữ liệu

Bộ dữ liệu được kiểm tra để phát hiện các giá trị bị thiếu, giá trị ngoại lai, và định dạng không hợp lệ.

- Các giá trị đều là số.
- **Giá trị thiếu/ lặp:** Không.

Number of duplicate rows: 0

Number of missing values per column:

Age	0
Gender	0
Ethnicity	0
EducationLevel	0
BMI	0
Smoking	0
AlcoholConsumption	0
PhysicalActivity	0
DietQuality	0
SleepQuality	0
FamilyHistoryAlzheimers	0
CardiovascularDisease	0
Diabetes	0
Depression	0
HeadInjury	0
Hypertension	0
SystolicBP	0
DiastolicBP	0
CholesterolTotal	0
CholesterolLDL	0
CholesterolHDL	0
CholesterolTriglycerides	0
MMSE	0
FunctionalAssessment	0
MemoryComplaints	0
BehavioralProblems	0
ADL	0
Confusion	0

```

Disorientation          0
PersonalityChanges     0
DifficultyCompletingTasks 0
Forgetfulness          0
Diagnosis              0
dtype: int64

```

- Giá trị ngoại lai: Không.

Number of outliers per numerical column (IQR method):

```

Age: 0
BMI: 0
AlcoholConsumption: 0
PhysicalActivity: 0
DietQuality: 0
SleepQuality: 0
SystolicBP: 0
DiastolicBP: 0
CholesterolTotal: 0
CholesterolLDL: 0
CholesterolHDL: 0
CholesterolTriglycerides: 0
MMSE: 0
FunctionalAssessment: 0
ADL: 0

```

- Các biến phân loại:

BehavioralProblems, CardiovascularDisease, Confusion,  
 Depression, Diabetes, DifficultyCompletingTasks, Disorientation,  
 EducationLevel, Ethnicity, FamilyHistoryAlzheimers, Forgetfulness,  
 Gender, HeadInjury, Hypertension, MemoryComplaints,  
 PersonalityChanges, Smoking.

Trong đó, ngoại trừ EducationLevel và Ethnicity, các biến đều là nhị phân.

- Các biến liên tục:

'Age', 'BMI', 'AlcoholConsumption', 'PhysicalActivity',

'DietQuality', 'SleepQuality', 'SystolicBP', 'DiastolicBP',  
'CholesterolTotal', 'CholesterolLDL', 'CholesterolHDL',  
'CholesterolTriglycerides', 'MMSE', 'FunctionalAssessment', 'ADL'.

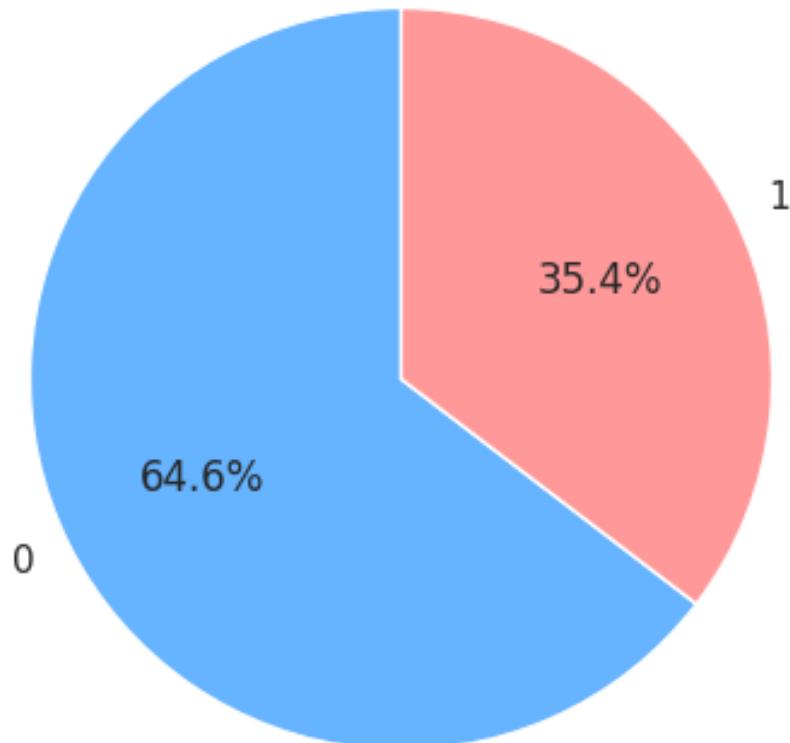
- Chúng ta sẽ bỏ 2 cột là PatientID và DoctorInCharge.

## 2. Trực quan hóa phân bố dữ liệu

Phân bố của các đặc trưng chính được biểu diễn bằng biểu đồ cột, boxplot, pieplot để quan sát xu hướng tổng thể.

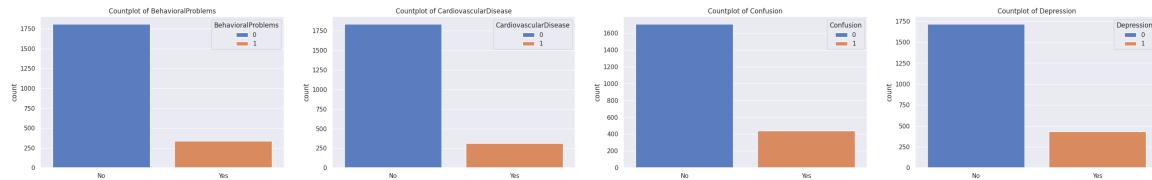
**Với biến mục tiêu:** Số quan sát của lớp 0 gần gấp đôi lớp 1.

Distribution of Diagnosis



Hình 4.1: Phân bố của biến mục tiêu.

## Với các biến phân loại:

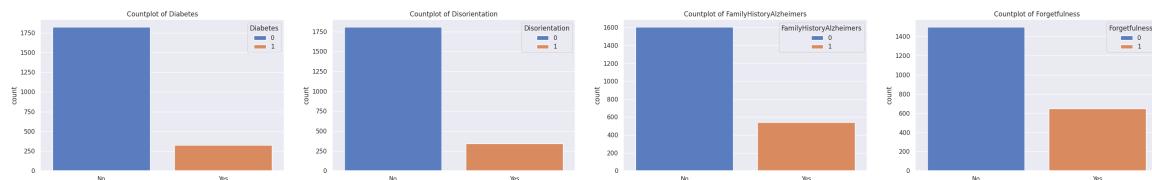


(a) Behavioral Problems

(b) CardiovascularDisease

(c) Confusion

(d) Depression

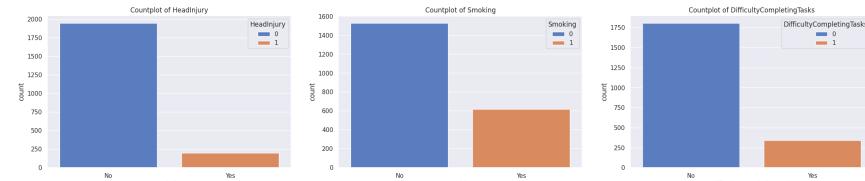


(e) Behavioral Problems

(f) CardiovascularDisease

(g) Confusion

(h) Depression

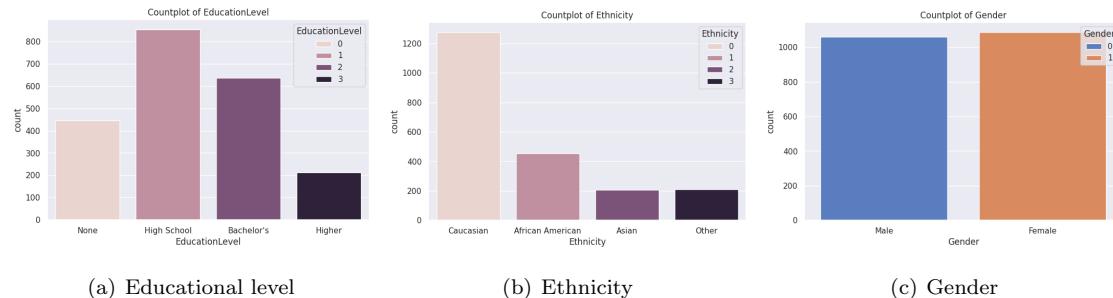


(i) Head injury

(j) Smoking

(k) DifficultyCompletingTasks

Hình 4.2: Phân bố của các biến về vấn đề sức khỏe.



(a) Educational level

(b) Ethnicity

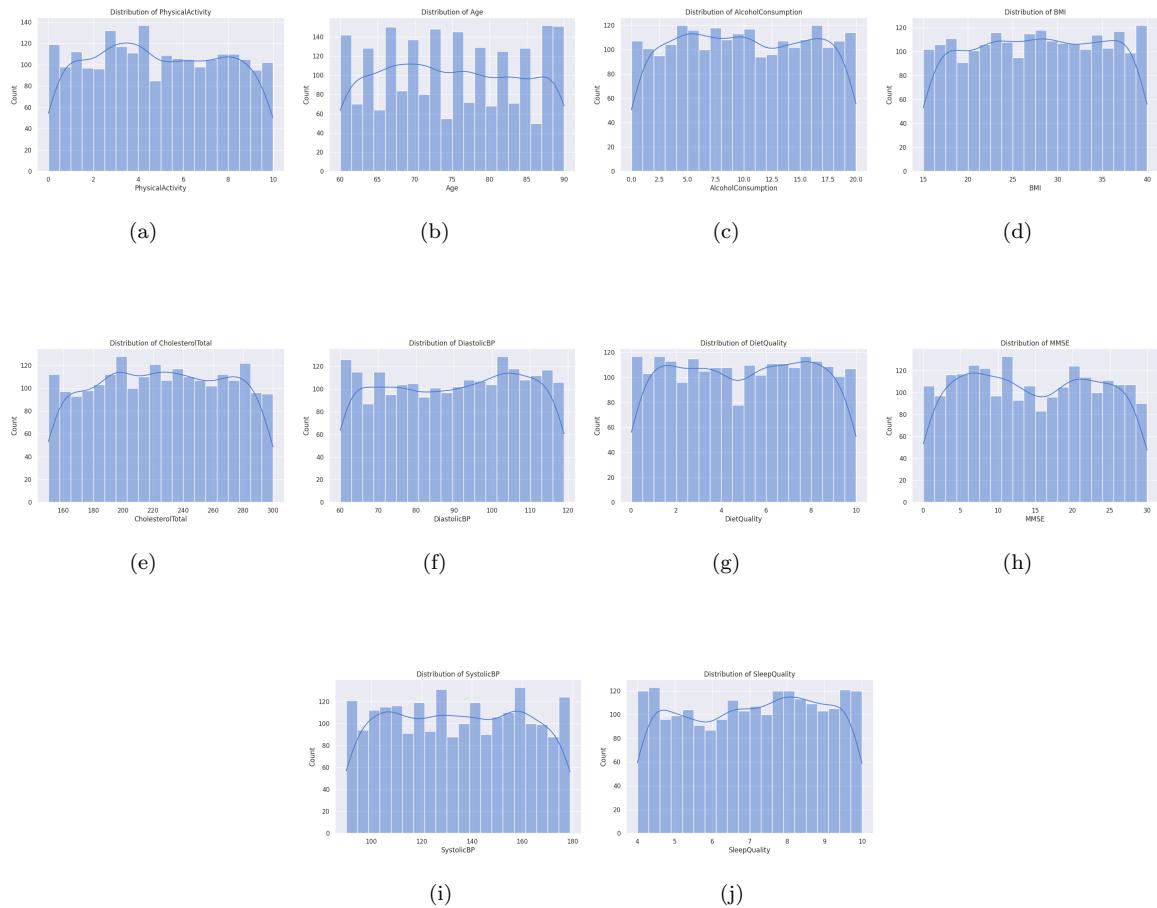
(c) Gender

Hình 4.3: Phân bố của các biến phân loại khác.

- Vấn đề sức khỏe/ tiền sử bệnh gia đình: 'No' chiếm phần lớn;

- Dân tộc: đa phần Caucasian;
- Học vấn: trung học chiếm nhiều nhất, tiếp theo là cử nhân;
- Giới tính: tỉ lệ nam-nữ rất cân bằng.

### Với các biến liên tục:



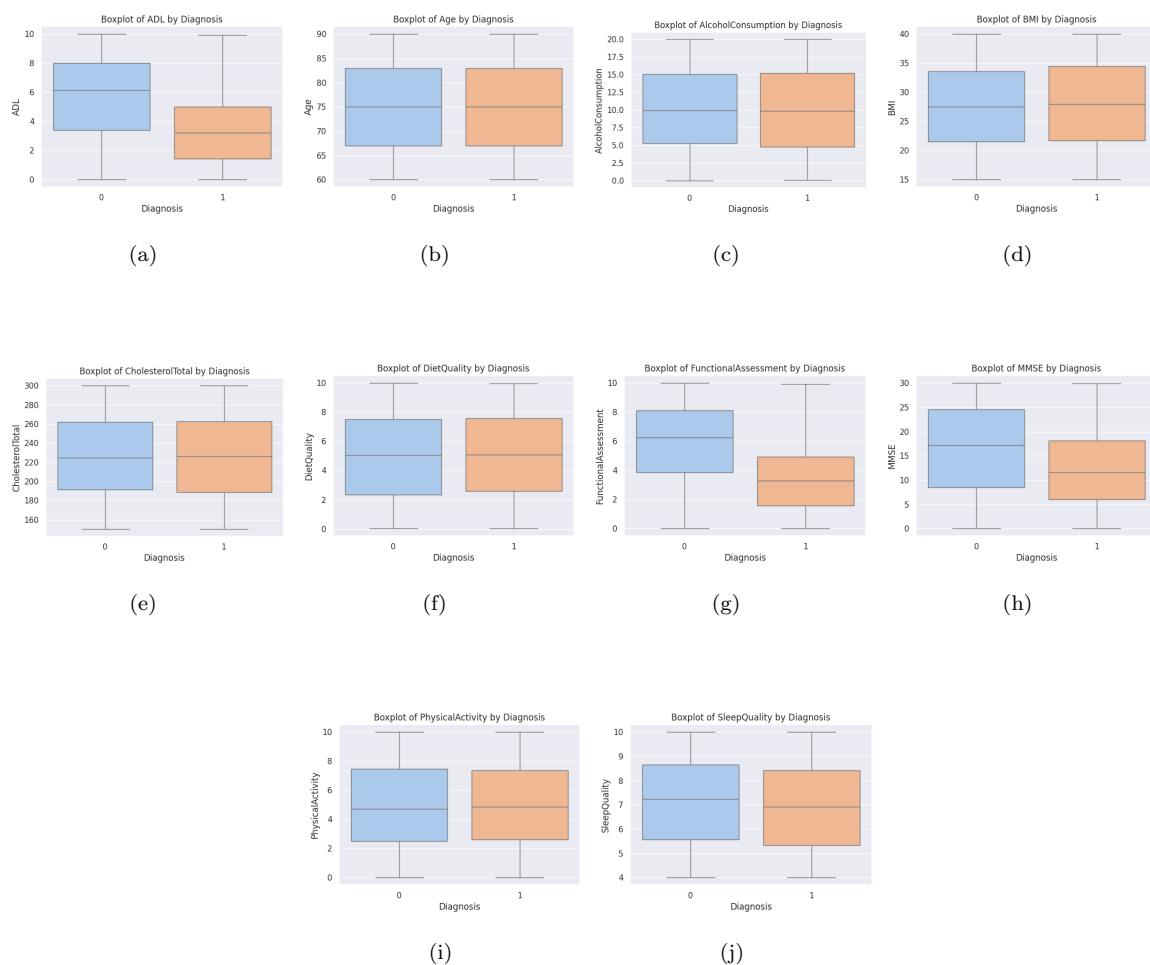
Hình 4.4: Phân bố của các biến liên tục.

- Đa phần các biến đều có phân bố khá đều (density line đường thẳng);
- MMSE có hai đỉnh rõ ràng, biểu thị rằng trong tập dữ liệu tồn tại hai nhóm bệnh nhân có mức điểm MMSE khác biệt — một nhóm với điểm thấp hơn (suy giảm nhận thức) và một nhóm với điểm cao hơn (nhận thức bình thường).

### 3. Biểu diễn đặc trưng theo biến mục tiêu

Các đặc trưng được biểu diễn so với biến mục tiêu (*Diagnosis*) để kiểm tra sự khác biệt giữa nhóm mắc Alzheimer và nhóm không mắc.

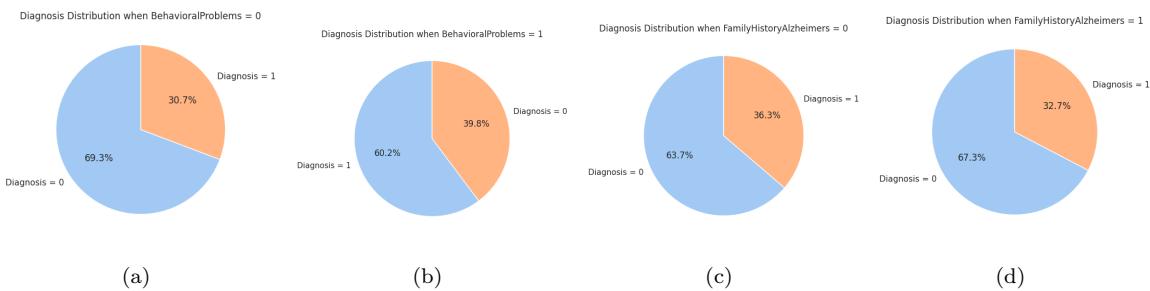
**Với các biến liên tục:**



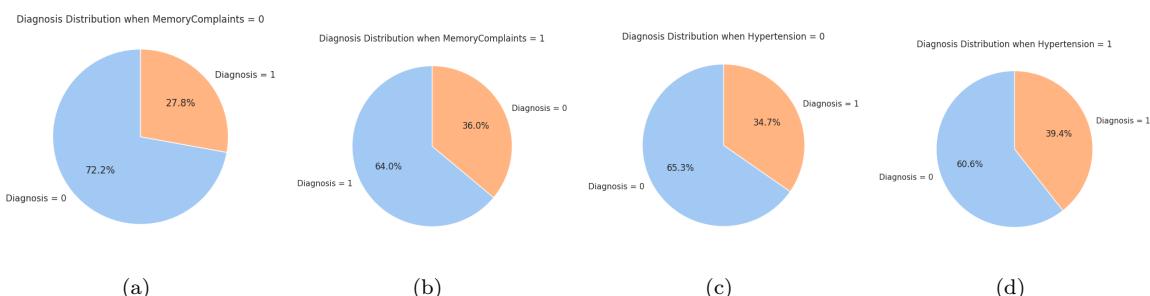
Hình 4.5: Trung bình biến liên tục với từng phân lớp mục tiêu.

- Bệnh nhân Alzheimer có **MMSE**, **avg ADL** và **FunctionalAssessment** thấp hơn rõ rệt.
- Không có khác biệt đáng kể với các biến khác.

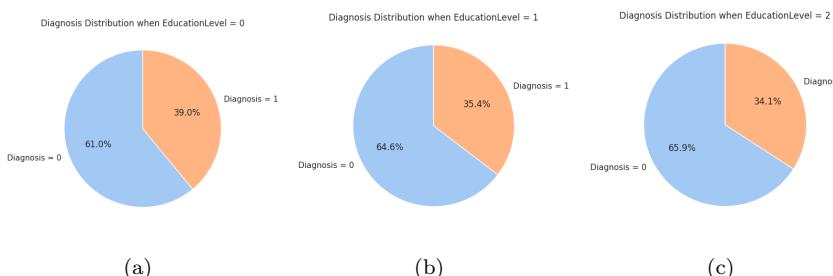
**Với các biến phân loại:**



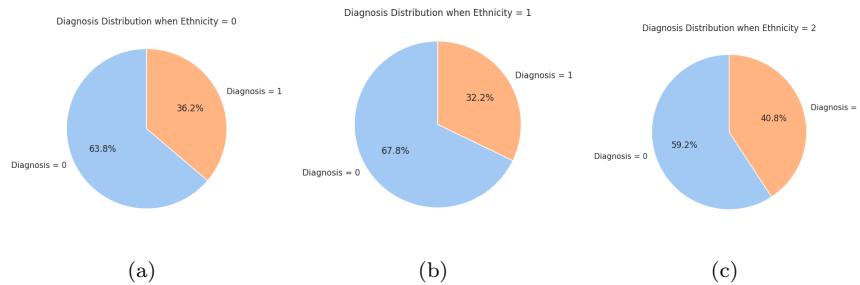
Hình 4.6: Phân bố của biến mục tiêu với BehavioralProblems và FamilyHistory.



Hình 4.7: Phân bố của biến mục tiêu với MemoryComplaints và Hypertension.



Hình 4.8: Phân bố của các phân lớp mục tiêu trong từng nhóm học vấn.

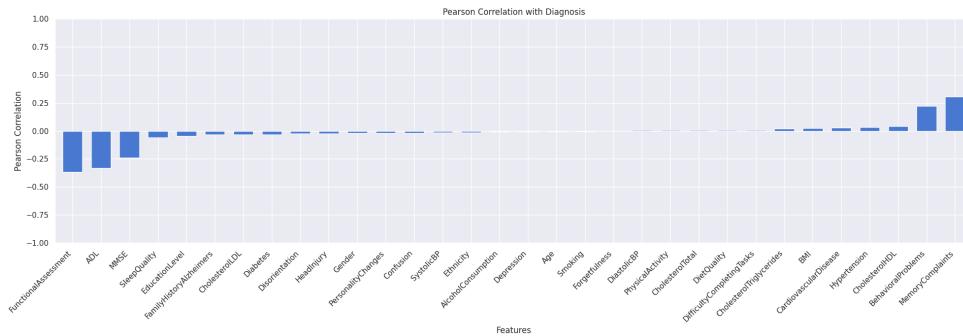


Hình 4.9: Phân bố của các phân lớp mục tiêu trong từng nhóm dân tộc.

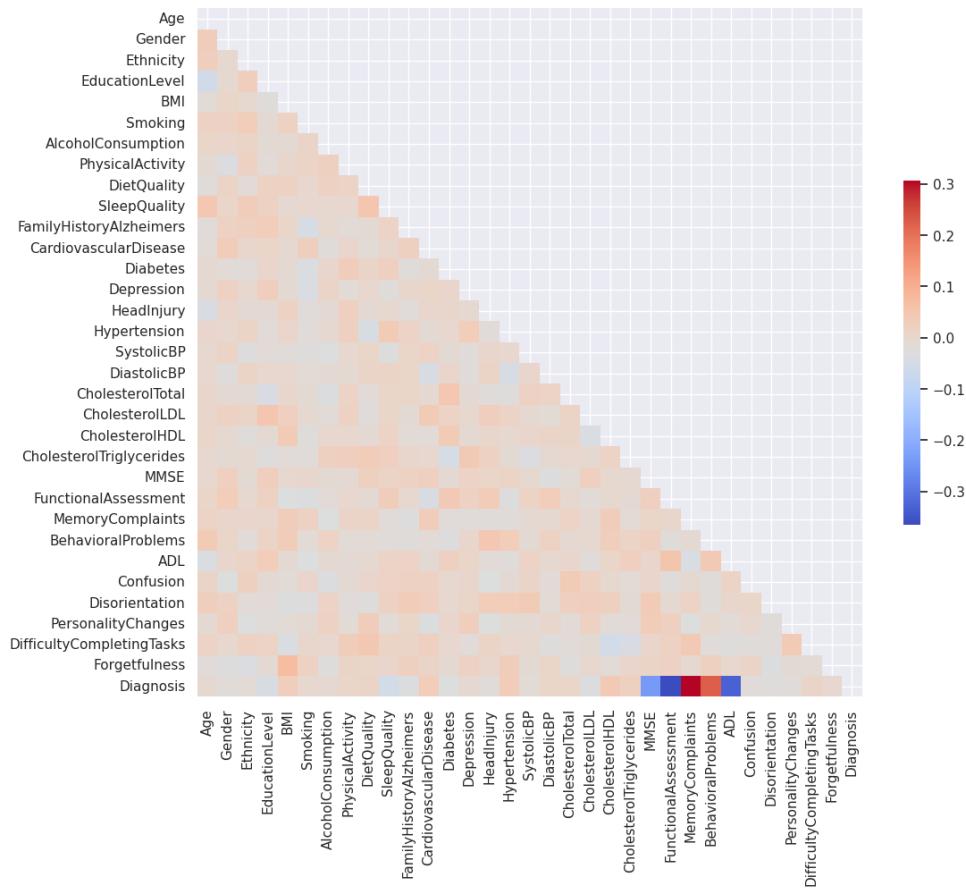
- Tỉ lệ giữa class 0 và class 1 ko chênh lệch nhiều giữa các trường của đa số các biến (Gender, Confusion, Depression, Diabetes,...)
- Với BehavioralProblems, Ethnicity, EducationLeval, MemoryComplaints: có sự chênh lệch rõ ràng hơn.

#### 4. Phân tích tương quan

Hệ số tương quan Pearson được tính để đánh giá mối liên hệ giữa các biến số.



Hình 4.10: Xếp thứ tự tương quan giữa các biến số.



Hình 4.11: Ma trận tương quan giữa các biến số.

- MMSE, Functional Assessment, Memory Complaints, Behavioral Problems, ADL có tương quan đáng kể nhất.
- Các biến khác đều có tương quan rất gần .
- Chúng ta cũng thấy xu hướng này trong các plots phía trên.

**Kết luận:** Quá trình tiền xử lý và phân tích khám phá giúp hiểu rõ hơn cấu trúc dữ liệu, mối liên hệ giữa các đặc trưng và biến mục tiêu, đồng thời hỗ trợ việc lựa chọn đặc trưng phù hợp cho các mô hình học máy ở bước tiếp theo.

# GIẢM CHIỀU VÀ TRỰC QUAN

## 5.1 Phân tích Thành phần Chính (PCA)

### 5.1.1 Mục tiêu của PCA

Giả sử ta có tập dữ liệu gồm  $n$  điểm dữ liệu, mỗi điểm là một vector trong không gian  $\mathbb{R}^p$ , tức là có  $p$  đặc trưng. Mục tiêu của PCA (Principal Component Analysis) là:

- Tìm hệ trục tọa độ mới (*thành phần chính*) sao cho:
  - Các trục này **vuông góc** với nhau.
  - Khi chiếu dữ liệu lên các trục này, **phương sai của dữ liệu là lớn nhất**.
- Giữ lại  $k < p$  trục đầu tiên để giảm chiều dữ liệu mà vẫn bảo toàn phần lớn thông tin.

### 5.1.2 Cơ sở toán học của PCA

#### Chuẩn hóa dữ liệu

Trước hết, dữ liệu được chuẩn hóa sao cho trung bình của mỗi đặc trưng bằng 0:

$$\mathbf{X}_{centered} = \mathbf{X} - \bar{\mathbf{X}}$$

#### Ma trận hiệp phương sai

$$\mathbf{C} = \frac{1}{n-1} \mathbf{X}_{centered}^\top \mathbf{X}_{centered}$$

#### Trị riêng và vector riêng

Giải:

$$\mathbf{C}\mathbf{v}_k = \lambda_k \mathbf{v}_k$$

$\lambda_k$ : trị riêng;  $\mathbf{v}_k$ : vector riêng (thành phần chính).

## Chiếu dữ liệu

$$\mathbf{Z} = \mathbf{X}_{centered} \mathbf{W}, \quad \mathbf{W} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k]$$

### 5.1.3 Diễn giải kết quả PCA

- PC1: hướng có phương sai lớn nhất.
- PC2: vuông góc với PC1 và giữ phương sai còn lại.
- Phương sai giữ lại:

$$\text{Explained Variance Ratio} = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^p \lambda_i}$$

## 5.2 Phân tích Phân biệt Tuyến tính (LDA)

### 5.2.1 Mục tiêu của LDA

LDA (*Linear Discriminant Analysis*) là phương pháp giảm chiều có giám sát, tối đa hóa sự phân tách giữa các lớp.

- PCA: không giám sát (unsupervised), tối đa hóa phương sai.
- LDA: có giám sát (supervised), tối đa hóa khả năng phân biệt lớp.

### 5.2.2 Cơ sở toán học của LDA

Giả sử có  $c$  lớp với trung bình lớp  $\boldsymbol{\mu}_i$  và trung bình toàn cục  $\boldsymbol{\mu}$ .

## Ma trận tán xạ

- Trong lớp:

$$\mathbf{S}_W = \sum_{i=1}^c \sum_{\mathbf{x} \in C_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^\top$$

- Giữa các lớp:

$$\mathbf{S}_B = \sum_{i=1}^c N_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^\top$$

## Bài toán tối ưu

$$J(\mathbf{W}) = \frac{|\mathbf{W}^\top \mathbf{S}_B \mathbf{W}|}{|\mathbf{W}^\top \mathbf{S}_W \mathbf{W}|}$$

Giải bài toán trị riêng tổng quát:

$$\mathbf{S}_B \mathbf{w}_i = \lambda_i \mathbf{S}_W \mathbf{w}_i$$

### 5.2.3 So sánh PCA và LDA

Đặc điểm	PCA	LDA
Loại phương pháp	Không giám sát	Có giám sát
Mục tiêu	Tối đa hóa phương sai	Tối đa hóa phân tách giữa các lớp
Sử dụng nhãn lớp	Không	Có
Dạng bài toán	Trị riêng chuẩn	Trị riêng tổng quát
Số chiều tối đa sau chiếu	$p$	$c - 1$ (với $c$ là số lớp)

### 5.2.4 Kết luận

LDA hữu ích khi dữ liệu có nhãn lớp rõ ràng, giúp không gian đặc trưng mới tách biệt tốt hơn giữa các lớp.

## 5.3 Giảm chiều bằng UMAP (Uniform Manifold Approximation and Projection)

### 5.3.1 Giới thiệu

UMAP là phương pháp giảm chiều dữ liệu phi tuyến hiện đại, dựa trên lý thuyết hình học và tôpô. Nó đặc biệt hiệu quả trong trực quan hóa dữ liệu có chiều cao (high-dimensional data) và thường được dùng thay thế t-SNE.

### 5.3.2 Nguyên lý hoạt động

UMAP giả định dữ liệu nằm trên một **đa tạp Riemann** (Riemannian manifold). Mục tiêu của UMAP:

- Xây dựng đồ thị k-láng giềng gần nhất (kNN graph) biểu diễn cấu trúc cục bộ.

- Biểu diễn đồ thị đó trong không gian thấp hơn sao cho **cấu trúc cục bộ và toàn cục được bảo toàn tốt nhất**.

### 5.3.3 Các bước chính

1. **Tạo đồ thị lân cận:** Với mỗi điểm dữ liệu, tìm  $k$  láng giềng gần nhất và tính trọng số:

$$w_{ij} = \exp\left(-\frac{d(x_i, x_j) - \rho_i}{\sigma_i}\right)$$

trong đó  $\rho_i$  là khoảng cách tới láng giềng gần nhất và  $\sigma_i$  được chọn sao cho mật độ được chuẩn hóa.

2. **Tối ưu đồ thị thấp chiều:** Xây dựng không gian nhúng mới (thường 2D hoặc 3D) bằng cách cực tiểu hóa độ mất mát cross-entropy giữa đồ thị gốc và đồ thị trong không gian thấp hơn.

### 5.3.4 Hàm mất mát của UMAP

UMAP tối thiểu hóa hàm mất mát dựa trên **cross-entropy** giữa phân bố đồ thị cao chiều và thấp chiều:

$$L = \sum_{i \neq j} \left[ w_{ij} \log \frac{w_{ij}}{w'_{ij}} + (1 - w_{ij}) \log \frac{1 - w_{ij}}{1 - w'_{ij}} \right]$$

Trong đó  $w_{ij}$  là trọng số trong không gian gốc và  $w'_{ij}$  là trọng số trong không gian nhúng.

### 5.3.5 Đặc điểm nổi bật của UMAP

- Bảo toàn tốt cả cấu trúc **cục bộ** và **toàn cục**.
- Tốc độ nhanh hơn t-SNE và mở rộng được cho dữ liệu lớn.
- Cho phép tái sử dụng embedding khi thêm dữ liệu mới.

### 5.3.6 So sánh PCA, LDA và UMAP

Đặc điểm	PCA	LDA	UMAP
Loại phương pháp	Tuyến tính	Tuyến tính (có nhãn)	Phi tuyến
Tính giám sát	Không giám sát	Có giám sát	Thường không giám sát (có thể mở rộng supervised)
Bảo toàn cấu trúc	Toàn cục (global variance)	Phân tách lớp	Cục bộ và toàn cục
Khả năng mở rộng	Cao	Trung bình	Cao (nhanh hơn t-SNE)
Ứng dụng chính	Giảm chiều, nén dữ liệu	Phân loại	Trực quan hóa dữ liệu phức tạp

## 5.4 Thực hành trên tập Alzheimer

### 5.4.1 Chuẩn hóa dữ liệu

- Biến nhị phân: giữ nguyên
- Ordinal: giữ nguyên (0, 1, 2,...) nếu thứ tự class có ý nghĩa, nếu không dùng one-hot encode  
(Chúng ta có thể coi EducationalLevel là biến ordinal hoặc nominal đều được.);
- Nominal: one-hot encoding (Ethnicity);
- Biến liên tục: standard scaler ( $std=1$ ,  $mean=0$ ).

### 5.4.2 PCA

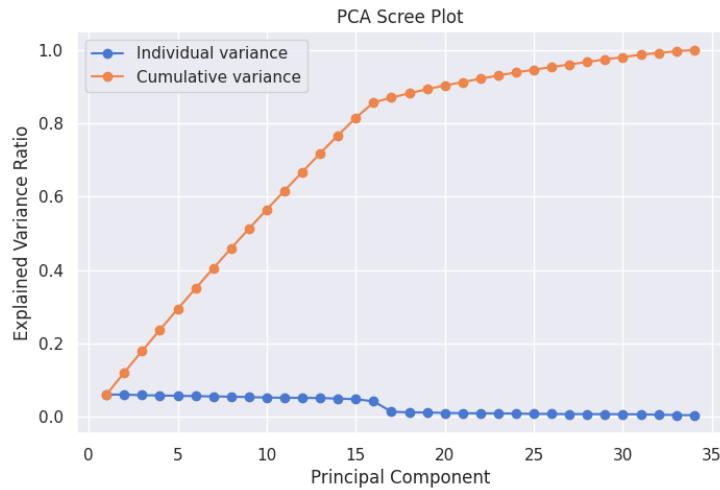
PCA tìm các tổ hợp tuyến tính của các đặc trưng nhằm tối đa hóa phương sai của dữ liệu và giảm chiều trong khi vẫn giữ được phần lớn thông tin. Phương pháp này hữu ích cho trực quan hóa, khám phá cấu trúc dữ liệu và tiền xử lý, nhưng cần chuẩn hóa dữ liệu vì nhạy cảm với thang đo.

Bảng 5.1: Tỷ lệ phương sai giải thích của các thành phần chính (PCA)

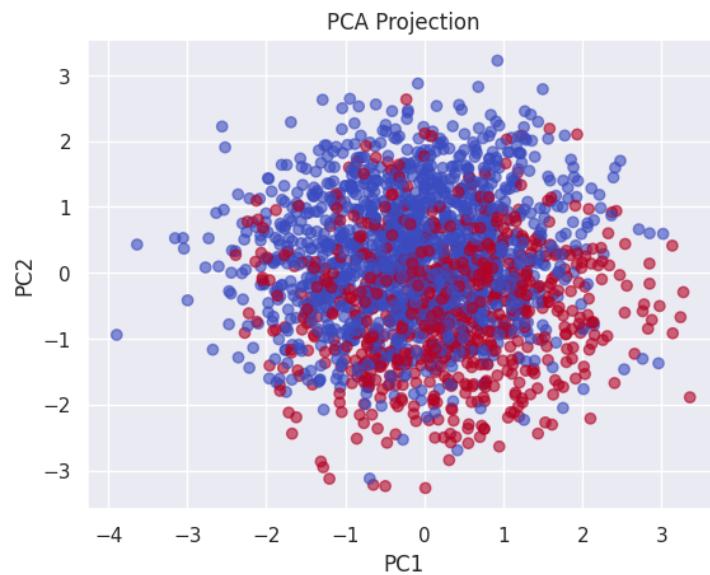
Thành phần chính	Tỷ lệ phương sai giải thích	Phương sai tích luỹ
PC1	0.0603	0.0603
PC2	0.0601	0.1204
PC3	0.0585	0.1789
PC4	0.0577	0.2365
PC5	0.0569	0.2934
PC6	0.0560	0.3495
PC7	0.0551	0.4045
PC8	0.0545	0.4591
PC9	0.0533	0.5123
PC10	0.0522	0.5646
PC11	0.0515	0.6161
PC12	0.0512	0.6673
PC13	0.0508	0.7180
PC14	0.0491	0.7672
PC15	0.0477	0.8149
PC16	0.0417	0.8566
PC17	0.0136	0.8701
PC18	0.0115	0.8816
PC19	0.0114	0.8930
PC20	0.0100	0.9030
PC21	0.0094	0.9124
PC22	0.0090	0.9215
PC23	0.0087	0.9302
PC24	0.0085	0.9387
PC25	0.0075	0.9462
PC26	0.0073	0.9534

Kết quả PCA cho thấy 15 thành phần chính đầu tiên giải thích phần lớn phương sai của dữ liệu, khoảng **81.49%**. Sau thành phần thứ 16, giá trị phương sai giảm đáng kể.

Dựa trên biểu đồ phương sai tích luỹ (Scree plot), có thể lựa chọn từ **13 đến 15 thành phần chính** để giữ được phần lớn thông tin của dữ liệu mà vẫn giảm được độ phức tạp của mô hình.

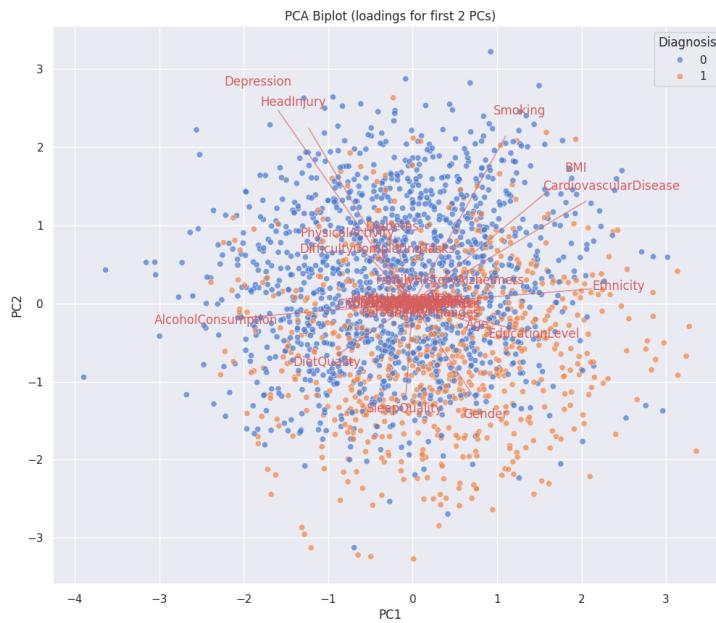


Hình 5.1: Scree plot.



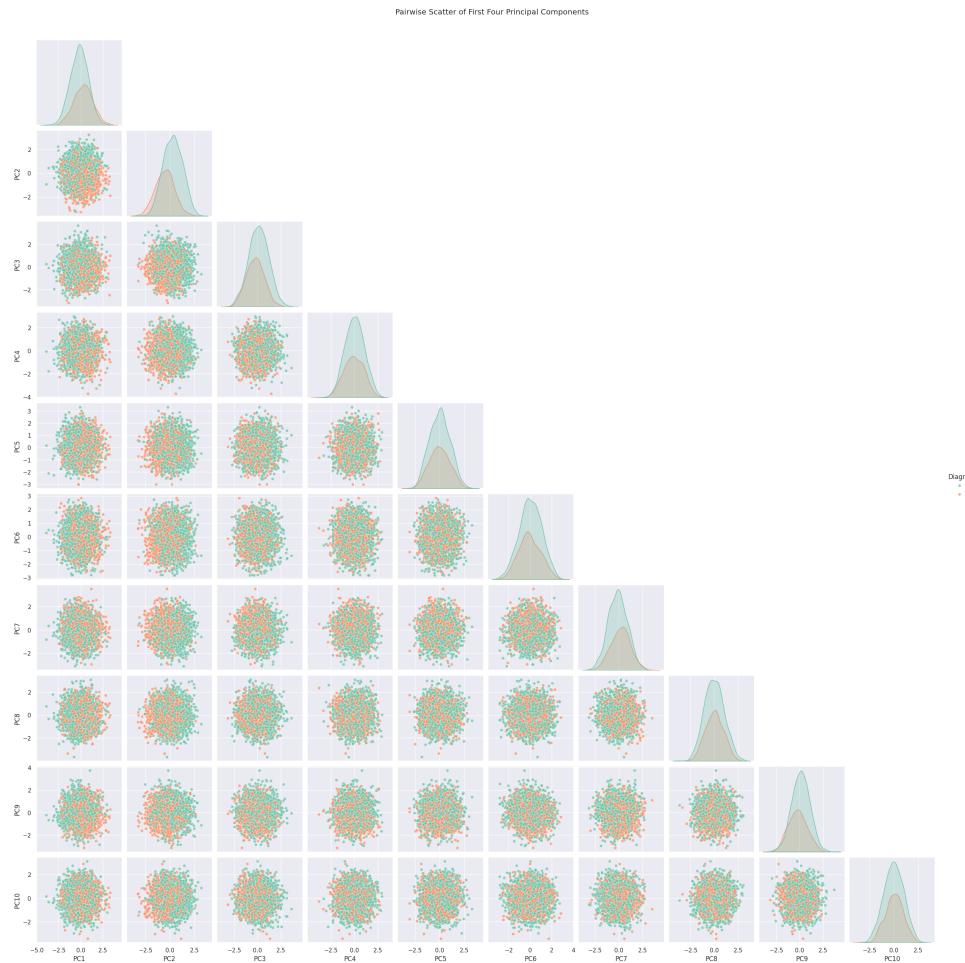
Hình 5.2: Plot dữ liệu theo PC1, PC2.

Biểu đồ phân tán theo hai thành phần chính (**PC1** và **PC2**) cho thấy **sự chồng lấn đáng kể** giữa các nhóm dữ liệu, cho thấy PC1 và PC2 chưa thể tách biệt rõ ràng giữa các bệnh nhân Alzheimer và không Alzheimer.



Hình 5.3: Biplot.

Tất cả các vectơ đặc trưng đều hướng theo các hướng khác nhau, cho thấy rằng các đặc trưng **gần như không tương quan** và **đóng góp theo nhiều hướng độc lập** vào hai thành phần chính. Điều này phản ánh rằng dữ liệu có **cấu trúc phức tạp**, không phát hiện **trục phương sai nổi trội**.



Hình 5.4: Plot từng cặp PCs.

Tất cả các biểu đồ cặp (*pairplots*) cho thấy **sự chồng lấn đáng kể** giữa các nhóm dữ liệu, cho thấy rằng các cặp thành phần chính này **chưa nắm bắt được nhiều thông tin phân biệt giữa các lớp**.

**Tóm tắt kết quả PCA:** Phân tích thành phần chính (PCA) cho thấy cần ít nhất **15 thành phần chính** để giải thích trên **80% phương sai** của dữ liệu gồm 33 đặc trưng. Điều này cho thấy mức độ tương quan giữa các đặc trưng không cao và phương sai được phân bố tương đối đồng đều giữa nhiều hướng trong không gian đặc trưng. Các biểu đồ phân tán (PC1–PC2) và biểu đồ cặp cho thấy **sự chồng lấn đáng kể** giữa các nhóm, chứng tỏ các thành phần chính đầu tiên **chưa thể hiện rõ khả năng phân biệt lớp**. Biplot cũng cho thấy các vectơ đặc trưng hướng theo nhiều hướng khác nhau, phản ánh cấu trúc dữ liệu phức tạp và không tồn tại trực biến thiên chi phối rõ rệt.

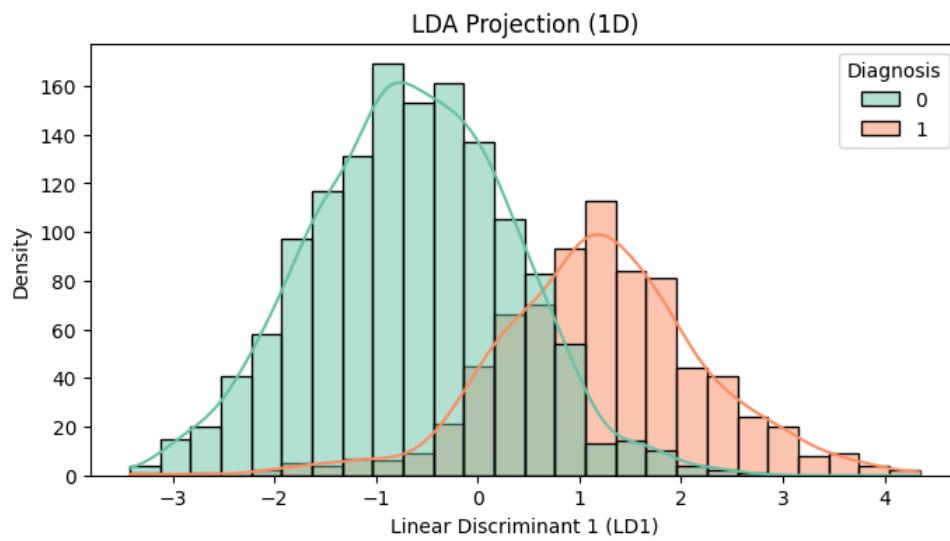
**Nhận xét về tác động của PCA:**

- **Giảm chiều dữ liệu:** PCA giúp loại bỏ nhiễu và mối tương quan tuyến tính giữa các đặc trưng, tuy nhiên mức độ giảm chiều còn hạn chế (từ 33 xuống khoảng 15 chiều).
- **Trực quan hóa dữ liệu:** Hai hoặc ba thành phần đầu tiên cho phép biểu diễn dữ liệu trong không gian thấp chiều, giúp quan sát được xu hướng phân bố và cấu trúc tổng thể của các nhóm.
- **Phục vụ bài toán phân loại:** Mặc dù PCA không tách biệt rõ hai lớp, nhưng kết quả giảm chiều giúp **chuẩn hóa và nén thông tin** đầu vào, hỗ trợ hiệu quả cho các phương pháp phân loại tuyến tính và phi tuyến được áp dụng ở các bước tiếp theo.

Tổng thể, PCA đóng vai trò quan trọng trong việc **chuẩn bị dữ liệu, giảm nhiễu và tăng tính ổn định cho mô hình**, dù khả năng tách biệt lớp còn hạn chế trong không gian tuyến tính.

#### 5.4.3 LDA

LDA tìm các tổ hợp tuyến tính của đặc trưng nhằm tối đa hóa khả năng phân tách giữa các lớp và tối thiểu hóa phương sai trong lớp. Phương pháp này nhạy cảm với thang đo, nên cần chuẩn hóa dữ liệu trước khi áp dụng.



Hình 5.5: LDA

Xét về khả năng phân tách tuyến tính giữa các lớp, thành phần phân biệt đầu tiên của LDA (**LD1**) cho thấy hiệu quả tốt hơn so với các thành phần

chính đầu tiên của PCA (**PC1, PC2**). Phân bố của hai lớp dọc theo LD1 thể hiện **các đỉnh mật độ tách biệt rõ hơn** so với khi quan sát theo PC1 hoặc PC2, cho thấy LDA đã tìm được hướng chiếu tối ưu hơn cho việc phân loại. Tuy nhiên, vẫn tồn tại **sự chồng lấn đáng kể** giữa hai lớp, phản ánh rằng ranh giới tuyến tính chưa thể tách biệt hoàn toàn các nhóm dữ liệu — điều này có thể do sự tương đồng về đặc trưng giữa các đối tượng trong tập dữ liệu.

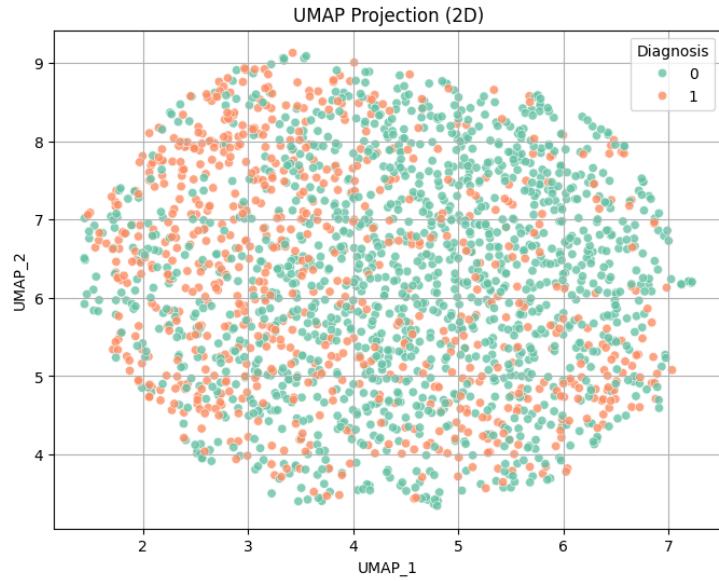
#### 5.4.4 UMAP

**UMAP** là một phương pháp giảm chiều dữ liệu phi tuyến, được thiết kế để **bảo tồn cấu trúc toàn cục và cục bộ** của dữ liệu gốc trong không gian có chiều thấp hơn. Khác với PCA hay LDA, UMAP không chỉ dựa trên các mối quan hệ tuyến tính mà còn khai thác các **mối quan hệ phi tuyến** giữa các điểm dữ liệu, giúp:

- Giữ gần các điểm dữ liệu có quan hệ gần nhau trong không gian cao chiều;
- Tách biệt các nhóm dữ liệu có cấu trúc khác nhau, ngay cả khi các ranh giới giữa lớp là phi tuyến.

UMAP đặc biệt hữu ích trong các bài toán **trực quan hóa dữ liệu phức tạp** và **khám phá cấu trúc nhóm** mà các phương pháp tuyến tính như PCA hay LDA khó phát hiện được. Phương pháp này **không yêu cầu chuẩn hóa dữ liệu** nghiêm ngặt, nhưng kết quả có thể bị ảnh hưởng bởi các siêu tham số như `n_neighbors` và `min_dist`, do đó cần điều chỉnh cẩn thận để đạt hiệu quả trực quan tốt nhất.

Với 2 thành phần chính:

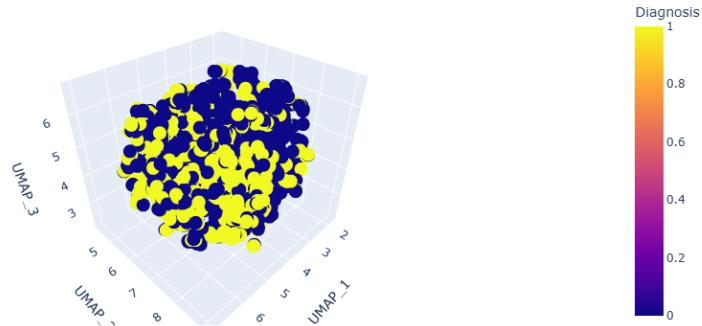


Hình 5.6: Unsupervised UMAP 2D

Chúng ta chưa thấy được sự phân cụm rõ ràng, thực tế 2 lớp 0, 1 bị trùng lặp rất nhiều. Có thể thấy lớp 0 dày hơn ở bên phải (tương ứng giá trị cao hơn của UMAP 1).

### Với 3 thành phần chính:

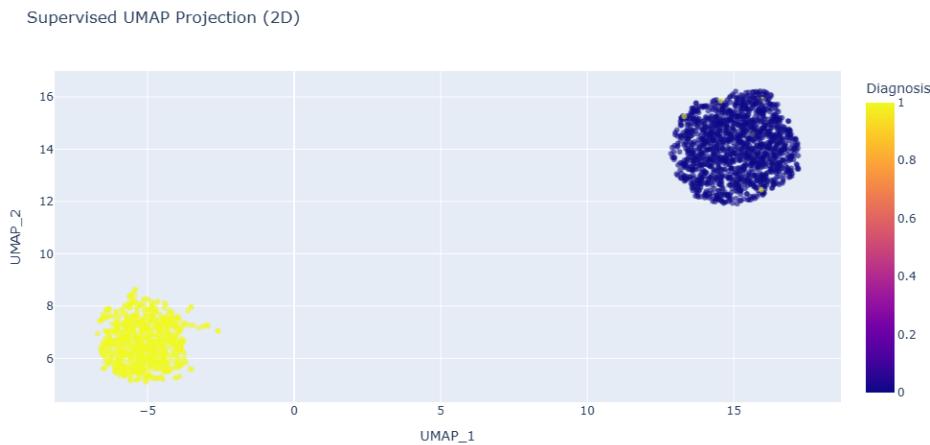
UMAP Projection (3D) - n\_neighbors=5, min\_dist=0.05



Hình 5.7: Unsupervised UMAP 3D

Ta thấy thêm một số góc tập hợp nhiều class 0/1 (với giá trị lớn hơn của UMAP-1 ta thấy tập trung nhiều lớp 0 hơn, lớp 1 thường có UMAP-1 nhỏ), tuy nhiên vẫn ko có sự tách biệt rõ ràng giữa 2 lớp.

## UMAP có giám sát



Hình 5.8: Supervised UMAP 2D

**Khả năng phân tách lớp mạnh:** Việc UMAP, được hướng dẫn bởi nhãn chẩn đoán, có thể tìm ra một không gian 2 chiều mà các lớp dữ liệu tách biệt rõ ràng cho thấy tồn tại các :mẫu và sự khác biệt cơ bản đáng kể trong các đặc trưng, có mối liên hệ chặt chẽ với nhãn Diagnosis. Dữ liệu có cấu trúc tốt, cho phép tách biệt các lớp dựa trên thông tin phân loại.

**Biểu diễn hiệu quả trong không gian chiềut thấp:** Điều này cho thấy thông tin quan trọng nhất để phân biệt hai nhóm chẩn đoán có thể được biểu diễn hiệu quả chỉ với hai chiều. Điều này thuận lợi cho việc :trực quan hóa dữ liệu và gợi ý rằng các mô hình dự đoán có thể không cần phải xử lý toàn bộ độ phức tạp của không gian gốc cao chiều để đạt độ chính xác cao.

**Mối quan hệ phi tuyến là chìa khóa:** Vì PCA (không giám sát) và UMAP (không giám sát) không thể hiện mức độ phân tách rõ ràng như vậy, trong khi UMAP có giám sát lại làm được, điều này nhấn mạnh rằng các mẫu phân biệt giữa các lớp khả năng cao là :phi tuyến. UMAP có giám sát tận dụng nhãn lớp để tìm các ranh giới và cấu trúc phi tuyến này, giúp phân tách các lớp hiệu quả hơn.

### 5.4.5 Tổng kết

#### PCA (Phân tích thành phần chính)

**Mục tiêu:** Tìm các thành phần trực giao (orthogonal components) nhằm nắm bắt tối đa phương sai trong dữ liệu. PCA là phương pháp **không giám sát**.

**Phương pháp:** Biến đổi tuyến tính các đặc trưng.

**Kết quả:**

- PC1 chỉ giải thích một tỷ lệ phương sai tương đối thấp (khoảng 6%), cho thấy dữ liệu có nhiều chiều và phương sai phân tán trên nhiều đặc trưng.
- Trực quan hóa PC1 vs PC2 (và các cặp thành phần khác) cho thấy sự **chồng lấn đáng kể** giữa hai nhóm chẩn đoán.
- Cần giữ một số lượng lớn thành phần (khoảng 16 PC) để bảo toàn phần lớn phương sai tổng thể (khoảng 85%).

**Nhận xét:** PCA hữu ích để hiểu cấu trúc phương sai và giảm chiều dữ liệu trong khi vẫn giữ được phương sai, nhưng các thành phần đầu tiên không thể hiện rõ sự tách biệt tuyến tính giữa các lớp cho trực quan hóa.

### LDA (Phân tích discriminant tuyến tính)

**Mục tiêu:** Tìm các thành phần tuyến tính tối đa hóa sự phân tách giữa các lớp trong khi giảm thiểu phương sai trong từng lớp. LDA là phương pháp **giám sát** sử dụng nhãn mục tiêu.

**Phương pháp:** Biến đổi tuyến tính hướng tới phân loại, số lượng thành phần giới hạn bằng  $n_{\text{classes}} - 1$ .

**Kết quả:**

- LDA tạo ra một thành phần duy nhất (LD1) cho bài toán phân loại nhị phân.
- Biểu đồ histogram của LD1 cho thấy **chồng lấn vừa phải** giữa hai nhóm chẩn đoán, mặc dù các đỉnh phân bố có sự khác biệt nhất định.

**Nhận xét:** LDA tìm được trực tuyến tính tốt nhất để phân tách các lớp. Mặc dù có một số tách biệt, nhưng không hoàn toàn tuyến tính, phù hợp với kết quả UMAP gợi ý sự cấu trúc phi tuyến có thể quan trọng.

### UMAP (Uniform Manifold Approximation and Projection)

**Mục tiêu:** Tìm một biểu diễn không gian chiềudưới thấp bảo tồn cấu trúc topo của dữ liệu. UMAP là phương pháp **phi tuyến, không giám sát** thường dùng cho trực quan hóa.

**Phương pháp:** Giảm chiều phi tuyến dựa trên học manifold.

**Kết quả:** Biểu đồ 2D UMAP cho thấy sự tách biệt trực quan rõ ràng hơn giữa hai nhóm chẩn đoán so với PCA 2D.

**Nhận xét:** UMAP hiệu quả hơn PCA trong việc phát hiện cấu trúc hoặc các cụm phi tuyến liên quan đến chẩn đoán, là công cụ tốt hơn để trực quan hóa sự tách biệt các lớp.

### So sánh tổng quan

- Về trực quan hóa sự phân tách các lớp, UMAP tỏ ra hiệu quả nhất, gợi ý rằng các mối quan hệ phi tuyến quan trọng để phân biệt các nhóm chẩn đoán.
- Về giảm chiều dữ liệu trong khi giữ phương sai tổng thể, PCA vẫn là phương pháp tiêu chuẩn.
- Về tìm kiếm tuyến tính tối ưu cho phân tách lớp, LDA có thể áp dụng, nhưng hiệu quả bị giới hạn bởi giả định tuyến tính và số lượng thành phần.

Việc UMAP cho thấy tách biệt trực quan tốt hơn cả PCA và LDA cho thấy việc thử các mô hình phân loại phi tuyến trên dữ liệu gốc hoặc dữ liệu đã giảm chiều bằng PCA là một hướng đi triển vọng.

# KẾT QUẢ THỰC NGHIỆM

## 6.1 K-nearest neighbor

Việc chọn giá trị tối ưu cho các tham số như k (số lượng láng giềng) trong mô hình K-Nearest Neighbors rất quan trọng vì chúng ảnh hưởng trực tiếp đến hiệu suất và độ chính xác của mô hình. Quá trình tìm kiếm các giá trị này có thể được thực hiện thông qua phương pháp GridSearchCV, cho phép thử nghiệm với nhiều giá trị khác nhau của k, đồng thời đánh giá hiệu quả của mô hình trên mỗi giá trị này.

Nếu giá trị k quá nhỏ, mô hình có thể bị overfitting (quá khớp với dữ liệu huấn luyện), tức là mô hình sẽ học quá kỹ vào các đặc điểm cụ thể của dữ liệu huấn luyện, dẫn đến khả năng tổng quát kém trên dữ liệu mới. Ngược lại, nếu k quá lớn, mô hình có thể bị underfitting, tức là không học đủ các đặc trưng quan trọng của dữ liệu.

Để xác định giá trị k tối ưu, chúng ta có thể sử dụng GridSearchCV để thử nghiệm trên nhiều giá trị của k. Phương pháp này sẽ phân chia dữ liệu thành các phần nhỏ và huấn luyện mô hình với các giá trị của k trên các phần dữ liệu khác nhau, sau đó tính toán độ chính xác của mô hình trên mỗi tập kiểm tra. Cuối cùng, giá trị k mang lại độ chính xác cao nhất sẽ được chọn là tham số tối ưu cho mô hình KNN.

### 6.1.1 Trên tệp dữ liệu gốc

Sau khi chạy GridSearchCV trên ba tỷ lệ phân chia dữ liệu 4:1, 7:3, 6:4, với giá trị K được chọn là 15 cho phân chia 4:1 và K là 11 cho phân chia 7:3, 6:4, mô hình KNN cho thấy một số kết quả như sau:

Class	Precision 4:1	Recall 4:1	F1-score 4:1
0	0.92	0.88	0.90
1	0.80	0.86	0.83
Accuracy			0.88
Macro Avg	0.86	0.87	0.87
Weighted Avg	0.88	0.88	0.88

Bảng 6.1: Bảng kết quả phân loại cho phân chia 4:1

Class	Precision 7:3	Recall 7:3	F1-score 7:3
0	0.93	0.89	0.91
1	0.82	0.88	0.85
<b>Accuracy</b>			0.89
<b>Macro Avg</b>	0.87	0.89	0.88
<b>Weighted Avg</b>	0.89	0.89	0.89

Bảng 6.2: Bảng kết quả phân loại cho phân chia 7:3

Class	Precision 6:4	Recall 6:4	F1-score 6:4
0	0.91	0.87	0.89
1	0.77	0.84	0.80
<b>Accuracy</b>			0.86
<b>Macro Avg</b>	0.84	0.85	0.85
<b>Weighted Avg</b>	0.86	0.86	0.86

Bảng 6.3: Bảng kết quả phân loại cho phân chia 6:4

- **Tổng quan độ chính xác:** Độ chính xác (Accuracy) của mô hình giữ được hiệu suất ổn định trong khoảng 86-89% ở ba tỷ lệ phân chia 4:1, 7:3, 6:4. Điều này cho thấy mô hình có khả năng phân loại ổn định bất kể tỷ lệ huấn luyện và kiểm tra. Sự ổn định này chứng tỏ rằng mô hình không bị ảnh hưởng nhiều bởi sự thay đổi trong tỷ lệ phân chia giữa tập huấn luyện và tập kiểm tra, giúp tăng tính tổng quát của mô hình.
- **Hiệu suất theo lớp:** Các lớp 0 và 1 đạt kết quả Precision, Recall và F1-score khá cao, đặc biệt là Precision của lớp 0 là 0.93, F1-score của lớp 0 là 0.91. Nhưng lớp 1 có hiệu suất kém hơn, đặc biệt ở Precision cho thấy rằng một số dự đoán dương cho lớp 1 là sai (false positives) hoặc đặc trưng lớp 1 có phần trùng lặp với lớp khác.
- **Khó khăn khi phân loại:** Mô hình KNN dựa trên khoảng cách giữa các điểm dữ liệu. Vì vậy, khi các mẫu của hai lớp phân bố gần nhau trong không gian đặc trưng, ranh giới giữa các lớp trở nên không rõ ràng, dễ nhầm lẫn. Sự mất cân bằng giữa hai lớp cũng khiến lớp thiểu số khó được nhận dạng chính xác. Ở đây số lượng mẫu lớp 0 nhiều hơn, dẫn đến việc các điểm lân cận của lớp thiểu số (lớp 1) thường bị bao quanh bởi các điểm thuộc lớp đa số. Điều này khiến mô hình có xu hướng dự đoán sai sang lớp chiếm ưu thế.
- **Sự cải thiện khi sử dụng GridSearchCV:** GridSearchCV giúp mô hình

KNN lựa chọn giá trị K tối ưu cho mỗi tỉ lệ phân chia dữ liệu. Dựa vào đó hiệu suất phân loại được cải thiện rõ so với kết quả ban đầu. Đặc biệt, độ chính xác tổng thể tăng lên thể hiện rằng việc tinh chỉnh siêu tham số bằng GridSearchCV giúp mô hình tổng quát tốt hơn, giảm hiện tượng overfitting và cải thiện khả năng nhận diện chính xác giữa các lớp.

Mô hình KNN với việc lựa chọn K tối ưu thông qua GridSearchCV cho thấy sự ổn định và hiệu quả cao trong việc phân loại dữ liệu bất kể tỷ lệ phân chia huấn luyện và kiểm tra. Mặc dù mô hình hoạt động rất tốt với lớp 0 và 1, nhưng vẫn gặp một số khó khăn đó là lớp 1 có hiệu suất kém hơn so với lớp 0. Tuy nhiên, độ chính xác tổng thể của mô hình vẫn ổn định ở mức cao (khoảng 89%).

### 6.1.2 Trên tệp dữ liệu đã giảm chiều PCA

Sau khi chạy GridSearchCV trên ba tỷ lệ phân chia dữ liệu 4:1, 7:3, 6:4, với giá trị K được chọn là 15 cho phân chia 4:1, 7:3 và 6:4, mô hình KNN cho thấy một số kết quả như sau:

Class	Precision 4:1	Recall 4:1	F1-score 4:1
0	0.82	0.68	0.74
1	0.55	0.74	0.63
Accuracy			0.70
Macro Avg	0.69	0.71	0.69
Weighted Avg	0.73	0.70	0.70

Bảng 6.4: Bảng kết quả phân loại cho phân chia 4:1

Class	Precision 7:3	Recall 7:3	F1-score 7:3
0	0.85	0.68	0.76
1	0.57	0.79	0.66
Accuracy			0.72
Macro Avg	0.71	0.73	0.71
Weighted Avg	0.75	0.72	0.72

Bảng 6.5: Bảng kết quả phân loại cho phân chia 7:3

Class	Precision 6:4	Recall 6:4	F1-score 6:4
0	0.83	0.64	0.73
1	0.54	0.77	0.63
<b>Accuracy</b>			0.69
<b>Macro Avg</b>	0.69	0.70	0.68
<b>Weighted Avg</b>	0.73	0.69	0.69

Bảng 6.6: Bảng kết quả phân loại cho phân chia 6:4

- **Tổng quan độ chính xác:** Độ chính xác của mô hình mang lại hiệu suất ổn định ở mức trung bình khá khoảng 69-72% ở ba tỷ lệ phân chia 4:1, 7:3, 6:4. Tỷ lệ 7:3 cho hiệu suất tổng thể tốt nhất về độ chính xác. Ngoài ra tỷ lệ 6:4 cho F1 cao nhất trong cross-validation, cho thấy mô hình tổng quát hóa tốt hơn trong huấn luyện nhưng bị giảm nhẹ khi test do nhiều dữ liệu kiểm thử.
- **Hiệu suất theo lớp:** Cả ba tỷ lệ đều cho ra precision ở lớp 0 rất tốt (82-85%) nhưng recall dao động thấp chỉ (64-68%). Ngược lại, lớp 1 có recall cao (74-79%) cho thấy mô hình phát hiện tốt nhiều mẫu của lớp 1 nhưng cũng dễ bị nhầm lẫn. Điều đó cũng không ảnh hưởng quá nhiều, độ chính xác tổng thể duy trì ổn định hơn, chứng tỏ rằng mô hình thích nghi tốt hơn với dữ liệu đã chuyển sang không gian đặc trưng mới.
- **Sự cải thiện khi sử dụng GridSearchCV:** Khi kết hợp giảm chiều bằng PCA với tinh chỉnh siêu tham số bằng GridSearchCV thì PCA loại bỏ các chiều dữ liệu không quan trọng hoặc tương quan cao, tập trung vào những thành phần đặc trưng có sức phân biệt tốt nhất. Ngoài ra, cải thiện độ ổn định và khả năng tổng quát giúp hoạt động ổn định hơn trên các tập dữ liệu khác nhau, thể hiện qua Accuracy dao động trong khoảng 69-72% nhưng không giảm mạnh ở bất kỳ tỷ lệ chia dữ liệu nào.

Việc giảm chiều dữ liệu bằng PCA đã không làm giảm quá nhiều hiệu suất của mô hình, mặc dù có một sự giảm nhẹ về độ chính xác và các chỉ số liên quan đến lớp ‘1’. Các chỉ số tổng thể vẫn duy trì ổn định, cho thấy mô hình có thể hoạt động khá tốt với dữ liệu đã giảm chiều. Mô hình KNN cho thấy khả năng phân loại ổn định và tốt đối với lớp ‘0’, tuy nhiên cần cải thiện khả năng phân loại lớp ‘1’ khi giảm chiều dữ liệu.

Mô hình vẫn có thể cải thiện thêm hiệu suất, đặc biệt đối với các lớp có sự chồng chéo giữa các đặc trưng, thông qua việc điều chỉnh tham số GridSearchCV tốt hơn hoặc thử nghiệm các phương pháp giảm chiều dữ liệu khác.

### 6.1.3 Trên tệp dữ liệu đã giảm chiều LDA

Sau khi chạy GridSearchCV trên ba tỷ lệ phân chia dữ liệu 4:1, 7:3, 6:4, với giá trị K được chọn là 15 cho phân chia 4:1 và K là 13 cho phân chia 7:3, 6:4, mô hình KNN cho thấy một số kết quả như sau:

Class	Precision 4:1	Recall 4:1	F1-score 4:1
0	0.85	0.86	0.86
1	0.74	0.72	0.73
<b>Accuracy</b>			0.81
<b>Macro Avg</b>	0.80	0.79	0.80
<b>Weighted Avg</b>	0.81	0.81	0.81

Bảng 6.7: Bảng kết quả phân loại cho phân chia 4:1

Class	Precision 7:3	Recall 7:3	F1-score 7:3
0	0.86	0.88	0.87
1	0.76	0.74	0.75
<b>Accuracy</b>			0.83
<b>Macro Avg</b>	0.81	0.81	0.81
<b>Weighted Avg</b>	0.83	0.83	0.83

Bảng 6.8: Bảng kết quả phân loại cho phân chia 7:3

Class	Precision 6:4	Recall 6:4	F1-score 6:4
0	0.86	0.88	0.87
1	0.77	0.74	0.76
<b>Accuracy</b>			0.83
<b>Macro Avg</b>	0.82	0.81	0.81
<b>Weighted Avg</b>	0.83	0.83	0.83

Bảng 6.9: Bảng kết quả phân loại cho phân chia 6:4

- **Tổng quan độ chính xác:** Độ chính xác của mô hình có hiệu suất ổn định (81-83%) ở ba tỷ lệ phân chia, cao hơn rõ rệt so với giảm chiều bằng PCA. Điều này chứng tỏ rằng LDA hoạt động hiệu quả và ổn định hơn nhờ khả năng tìm ra tổ hợp tuyến tính tối ưu giữa các đặc trưng để tách biệt hai lớp.

- **Hiệu suất theo lớp:** Ở lớp 0 có Precision và Recall cao dao động 85-88%, F1-score dao động 86-87% cho thấy rằng mô hình phân loại đúng hầu hết các mẫu thuộc lớp này. Lớp 1 mặc dù thấp hơn lớp 0 nhưng Precision và Recall đều ở mức 72-77%, F1-score khoảng 73-76% cho thấy giảm chiều LDA nhận diện lớp thiểu số tốt hơn so với dữ liệu gốc.
- **Sự cải thiện khi sử dụng GridSearchCV:** Khi kết hợp giảm chiều bằng LDA không chỉ giúp tăng độ chính xác tổng thể mà còn cải thiện rõ rệt Precision, Recall và F1-score cho cả hai lớp. Sự chênh lệch giữa hai lớp đã được thu hẹp giúp mô hình cân bằng hơn trong việc nhận dạng cả hai lớp. Giảm chiều đồng thời giữ lại thông tin quan trọng giúp phân biệt lớp giúp mô hình tránh được hiện tượng nhiễu và chồng lấn dữ liệu.

Như vậy, việc giảm chiều dữ liệu bằng LDA đã giúp mô hình duy trì hiệu suất phân loại ổn định, nhưng vẫn cần cải thiện khả năng phân loại lớp 1, đặc biệt khi có sự chồng chéo giữa các lớp tuy nhiên độ chính xác nhỏ hơn dữ liệu gốc.

#### 6.1.4 Tổng kết trên tập dữ liệu gốc và dữ liệu giảm chiều

Sau khi thử nghiệm mô hình KNN với phương pháp GridSearchCV trên ba tập dữ liệu với ba phương pháp phân chia khác nhau 4:1, 7:3 và 6:4. Chúng ta nhận thấy mô hình vẫn duy trì hiệu suất phân loại ổn định bất kể việc giảm chiều dữ liệu bằng PCA hay LDA. Tuy nhiên, có sự khác biệt nhẹ trong kết quả giữa dữ liệu gốc và dữ liệu sau khi giảm chiều.

- **Dữ liệu gốc:** Độ chính xác trên dữ liệu gốc đạt khoảng 89%, với mô hình phân loại rất tốt cho cả lớp 0 và lớp 1. Các lớp đạt kết quả gần như xuất sắc, đặc biệt là lớp 0 với Precision đạt 0.93.
- **Dữ liệu giảm chiều PCA:** Sau khi giảm chiều dữ liệu bằng PCA, độ chính xác của mô hình giảm nhẹ xuống 69-72% so với dữ liệu gốc. Các chỉ số Precision, Recall và F1-score đối với hai lớp giảm nhẹ, đặc biệt lớp 1 gấp khó khăn hơn, đặc biệt là giảm nhẹ Recall và F1-score, cho thấy mô hình gấp khó khăn trong việc phân loại lớp 1. Tuy nhiên, các chỉ số tổng thể duy trì ở mức ổn định.
- **Dữ liệu giảm chiều LDA:** Sau khi giảm chiều dữ liệu bằng LDA, mô hình vẫn duy trì độ chính xác ở mức 81-83%, có sự giảm nhẹ so với dữ liệu gốc, nhưng vẫn ổn định. Các chỉ số cho lớp 0 vẫn giữ ở mức rất tốt, nhưng lớp 1 gấp phải sự giảm nhẹ ở Recall và F1-score, đặc biệt là ở các tỷ lệ phân chia

như 4:1. Mặc dù vậy, mô hình vẫn duy trì được kết quả khá tốt với các chỉ số tổng thể ở mức ổn định.

## 6.2 Softmax regression

Mặc dù bài toán đặt ra là phân loại nhị phân, nhóm vẫn lựa chọn sử dụng Logistic Regression dạng softmax với thiết lập `mult_class='multinomial'` nhằm khai thác khả năng ước lượng xác suất chính xác cho từng lớp. Việc sử dụng hàm softmax thay vì hàm sigmoid truyền thống không gây ảnh hưởng tới độ chính xác của mô hình nhị phân, đồng thời giúp mô hình duy trì sự nhất quán nếu sau này mở rộng sang phân loại đa lớp.

Bên cạnh đó, `solver='lbfgs'` được lựa chọn do đây là thuật toán tối ưu phù hợp với bài toán logistic regression có sử dụng hàm mất mát mềm như softmax. Để đảm bảo mô hình hội tụ ổn định ngay cả trong trường hợp dữ liệu nhiều chiều hoặc phân bố phức tạp, tham số `max_iter` được đặt giá trị lớn là 20000. Nếu để giá trị quá thấp, mô hình có thể không hội tụ được, dẫn đến kết quả không chính xác hoặc cảnh báo dừng sớm khi huấn luyện.

### 6.2.1 Trên tệp dữ liệu gốc

Class	Precision 4:1	Recall 4:1	F1-score 4:1
0	0.87	0.87	0.87
1	0.76	0.76	0.76
<b>Accuracy</b>			0.83
<b>Macro Avg</b>	0.81	0.82	0.81
<b>Weighted Avg</b>	0.83	0.83	0.83

Bảng 6.10: Bảng kết quả phân loại cho phân chia 4:1

Class	Precision 7:3	Recall 7:3	F1-score 7:3
0	0.87	0.88	0.88
1	0.78	0.77	0.77
<b>Accuracy</b>			0.84
<b>Macro Avg</b>	0.83	0.83	0.83
<b>Weighted Avg</b>	0.84	0.84	0.84

Bảng 6.11: Bảng kết quả phân loại cho phân chia 7:3

Class	Precision 6:4	Recall 6:4	F1-score 6:4
0	0.88	0.87	0.87
1	0.77	0.77	0.77
<b>Accuracy</b>			0.84
<b>Macro Avg</b>	0.82	0.82	0.82
<b>Weighted Avg</b>	0.84	0.84	0.84

Bảng 6.12: Bảng kết quả phân loại cho phân chia 6:4

Mô hình phân loại cho các tỉ lệ phân chia dữ liệu 4:1, 7:3 và 6:4 đều đạt độ chính xác (accuracy) tốt, dao động quanh mức 83-84%. Tuy nhiên, các chỉ số precision, recall và F1-score phân bố chưa thật sự đồng đều giữa hai lớp. Cụ thể, lớp 0 đạt kết quả cao hơn với precision, recall và F1-score trong khoảng 87-88%; trong khi đó, lớp 1 có các chỉ số thấp hơn, dao động từ 76-78%.

Dù vậy, sự chênh lệch giữa hai lớp không quá lớn, thể hiện qua các chỉ số macro average và weighted average đều nằm trong khoảng 0.81-0.84, phản ánh hiệu suất mô hình tương đối ổn định và cân bằng giữa các lớp.

### 6.2.2 Trên tệp dữ liệu đã giảm chiều PCA

Class	Precision 4:1	Recall 4:1	F1-score 4:1
0	0.86	0.87	0.86
1	0.75	0.74	0.74
<b>Accuracy</b>			0.82
<b>Macro Avg</b>	0.80	0.80	0.80
<b>Weighted Avg</b>	0.82	0.82	0.82

Bảng 6.13: Bảng kết quả phân loại cho phân chia 4:1

Class	Precision 7:3	Recall 7:3	F1-score 7:3
0	0.87	0.88	0.88
1	0.78	0.76	0.77
<b>Accuracy</b>			0.84
<b>Macro Avg</b>	0.82	0.82	0.82
<b>Weighted Avg</b>	0.84	0.84	0.84

Bảng 6.14: Bảng kết quả phân loại cho phân chia 7:3

Class	Precision 6:4	Recall 6:4	F1-score 6:4
0	0.87	0.87	0.87
1	0.76	0.76	0.76
<b>Accuracy</b>			0.83
<b>Macro Avg</b>	0.82	0.82	0.82
<b>Weighted Avg</b>	0.83	0.83	0.83

Bảng 6.15: Bảng kết quả phân loại cho phân chia 6:4

Sau khi giảm chiều dữ liệu bằng PCA, mô hình đạt độ chính xác (accuracy) dao động từ 0.82 đến 0.84 cho ba tỉ lệ phân chia dữ liệu (4:1, 7:3 và 6:4), thấp hơn một chút so với khi huấn luyện trên tập dữ liệu gốc. Các chỉ số precision, recall và F1-score của cả hai lớp đều giảm nhẹ, đặc biệt là ở lớp 1, với F1-score chỉ đạt khoảng 0.74-0.77, trong khi lớp 0 vẫn duy trì kết quả khá tốt, khoảng 0.86-0.88.

Việc giảm chiều dữ liệu bằng kỹ thuật PCA(Principal Component Analysis) giúp loại bỏ nhiễu và giảm độ phức tạp của mô hình, tuy nhiên cũng có thể dẫn đến mất mát thông tin quan trọng. PCA chỉ giữ lại những thành phần chính có phương sai lớn nhất trong dữ liệu, nhưng không đảm bảo rằng các thành phần đó là những đặc trưng tốt nhất cho mục tiêu phân loại. Do đó, một phần thông tin đặc trưng phân lớp có thể bị loại bỏ, làm mô hình khó phân biệt hai lớp chính xác như khi sử dụng dữ liệu gốc đầy đủ.

Ngoài ra, việc giảm chiều đồng nghĩa với việc biểu diễn dữ liệu trên một không gian con có ít thông tin hơn, dẫn đến rủi ro giảm độ phân tách giữa các lớp. Điều này giải thích tại sao các chỉ số như accuracy, precision, recall, F1-score hay macro/weighted average đều có xu hướng giảm nhẹ sau PCA.

Mặc dù vậy, mô hình vẫn thể hiện được hiệu suất ổn định trên các tỉ lệ phân chia khác nhau, thể hiện qua các giá trị macro average và weighted average dao động quanh 0.80-0.84. Điều này cho thấy PCA đã giữ lại được phần lớn thông tin quan trọng, giúp mô hình tiếp tục học hiệu quả trên không gian đặc trưng đã giảm chiều, dù hiệu suất tổng thể có giảm nhẹ so với dữ liệu gốc.

### 6.2.3 Trên tệp dữ liệu đã giảm chiều LDA

<b>Class</b>	<b>Precision 4:1</b>	<b>Recall 4:1</b>	<b>F1-score 4:1</b>
0	0.87	0.86	0.86
1	0.75	0.76	0.75
<b>Accuracy</b>			0.83
<b>Macro Avg</b>	0.81	0.81	0.81
<b>Weighted Avg</b>	0.83	0.83	0.83

Bảng 6.16: Bảng kết quả phân loại cho phân chia 4:1

<b>Class</b>	<b>Precision 7:3</b>	<b>Recall 7:3</b>	<b>F1-score 7:3</b>
0	0.87	0.89	0.88
1	0.79	0.76	0.77
<b>Accuracy</b>			0.84
<b>Macro Avg</b>	0.83	0.82	0.83
<b>Weighted Avg</b>	0.84	0.84	0.84

Bảng 6.17: Bảng kết quả phân loại cho phân chia 7:3

<b>Class</b>	<b>Precision 6:4</b>	<b>Recall 6:4</b>	<b>F1-score 6:4</b>
0	0.87	0.88	0.88
1	0.77	0.77	0.77
<b>Accuracy</b>			0.84
<b>Macro Avg</b>	0.82	0.82	0.82
<b>Weighted Avg</b>	0.84	0.84	0.84

Bảng 6.18: Bảng kết quả phân loại cho phân chia 6:4

Trên tập dữ liệu sau khi giảm chiều bằng LDA (Linear Discriminant Analysis), mô hình đạt hiệu suất gần như tương đồng với khi huấn luyện trên tập dữ liệu gốc, với độ chính xác (accuracy) dao động trong khoảng 0.83–0.84 ở cả ba tỉ lệ phân chia dữ liệu (4:1, 7:3 và 6:4). Các chỉ số precision, recall và F1-score của hai lớp đều được duy trì ở mức cao, chỉ có một vài thay đổi rất nhỏ, chẳng hạn precision của lớp 0 ở tỉ lệ 6:4 giảm khoảng 1%, hay macro average của recall giảm khoảng 1%, mức chênh lệch này là không đáng kể.

Khác với PCA, kỹ thuật LDA không chỉ quan tâm đến phương sai dữ liệu mà còn tối đa hóa khả năng phân tách giữa các lớp. Cụ thể, LDA tìm kiếm một không gian con mà tại đó khoảng cách giữa các lớp là lớn nhất, trong khi khoảng cách nội lớp là nhỏ nhất. Do đó, đặc trưng tạo ra từ LDA giữ lại được

thông tin phân loại cốt lõi, giúp mô hình vẫn phân biệt tốt giữa hai lớp ngay cả khi số chiều bị giảm.

Sự dao động nhẹ ở một vài chỉ số có thể xuất phát từ biến động tự nhiên của dữ liệu khi chia tập train-test, chứ không phải do mất mát thông tin trong quá trình giảm chiều. Nhìn chung, mô hình vẫn giữ được độ chính xác và tính ổn định cao, cho thấy LDA là phương pháp giảm chiều hiệu quả cho bài toán phân loại nhị phân, đặc biệt trong các trường hợp phân bố của hai lớp có độ tách biệt rõ ràng.

Kết quả này cũng khẳng định rằng việc áp dụng LDA không chỉ giúp giảm số chiều và rút ngắn thời gian huấn luyện, mà còn duy trì được hiệu suất tương đương với mô hình huấn luyện trên tập dữ liệu đầy.

#### 6.2.4 Tổng kết trên tập dữ liệu gốc và dữ liệu giảm chiều

Qua quá trình thực nghiệm trên tập dữ liệu gốc và hai phương pháp giảm chiều PCA và LDA, có thể nhận thấy rằng mô hình duy trì được hiệu suất ổn định và độ chính xác cao ở cả ba trường hợp. Trong đó, mô hình trên tập dữ liệu gốc vẫn đạt kết quả cao và ổn định nhất. Từ đó, phản ánh được khả năng khai thác thông tin đặc trưng đầy đủ của tập dữ liệu ban đầu.

Với PCA, mặc dù độ chính xác và các chỉ số như precision, recall, F1-score giảm nhẹ do mất mát một phần thông tin trong quá trình chiếu dữ liệu, song mô hình vẫn đảm bảo khả năng phân loại tốt và tương đối ổn định.

Ngược lại, LDA cho thấy hiệu quả vượt trội hơn khi duy trì hiệu suất gần như tương đương dữ liệu gốc, đồng thời tăng tính ổn định giữa các tỉ lệ phân chia và giảm đáng kể số chiều đặc trưng, giúp mô hình huấn luyện nhanh hơn mà không ảnh hưởng nhiều đến chất lượng.

Nhìn chung, việc áp dụng các kỹ thuật giảm chiều không làm suy giảm đáng kể hiệu suất mô hình; trong đó, LDA được đánh giá là phương pháp phù hợp hơn PCA cho bài toán phân loại nhị phân này, nhờ khả năng giữ vững độ chính xác, ổn định và tối ưu hóa hiệu quả tính toán.

### 6.3 So sánh hai mô hình KNN và Softmax

Trong khuôn khổ bài thực hành, hai mô hình phân loại K-Nearest Neighbors (KNN) và Softmax Regression đã được triển khai trên cùng một tập dữ liệu nhị phân với ba dạng biểu diễn khác nhau gồm dữ liệu gốc, dữ liệu giảm chiều bằng PCA, và dữ liệu giảm chiều bằng LDA. Dựa trên kết quả đánh giá định lượng và trực quan hóa, có thể rút ra một số nhận định tổng quan như sau:

Kết quả thực nghiệm cho thấy mô hình KNN đạt hiệu năng cao hơn trên dữ liệu gốc, trong khi Softmax Regression thể hiện độ ổn định tốt hơn khi giảm chiều dữ liệu bằng PCA, và cả hai mô hình cho kết quả tương đương nhau trên LDA. Cụ thể, trên tập dữ liệu gốc, KNN đạt độ chính xác trung bình khoảng 86–89%, cao hơn Softmax Regression khoảng 3–6%. Tuy nhiên, khi giảm chiều dữ liệu bằng PCA, cả hai mô hình đều bị suy giảm hiệu suất do mất mát thông tin, trong đó Softmax Regression thể hiện sự ổn định hơn, giữ được độ chính xác quanh mức 78–80%, còn KNN giảm rõ rệt xuống còn 60–72%. Ngược lại, với LDA, hiệu năng của cả hai mô hình đều được phục hồi đáng kể và gần tương đương nhau, dao động quanh 83–84%, cho thấy LDA là phương pháp giảm chiều hiệu quả trong việc bảo toàn thông tin phân biệt lớp.

Xét về độ ổn định khi thay đổi biểu diễn dữ liệu, Softmax Regression có xu hướng giữ được hiệu năng khá ổn định. Là một mô hình tuyến tính với cơ chế tối ưu hóa rõ ràng, Softmax ít bị ảnh hưởng tiêu cực khi chiều dữ liệu giảm, đặc biệt là với LDA - nơi không gian được tối ưu theo hướng phân biệt các lớp. KNN thì ngược lại, phụ thuộc nhiều vào cấu trúc hình học của dữ liệu trong không gian đặc trưng, do đó hiệu năng của nó bị suy giảm đáng kể khi sử dụng PCA (vì PCA không đảm bảo duy trì khoảng cách giữa các lớp). Tuy nhiên, khi chuyển sang LDA - vốn bảo toàn biên phân lớp - hiệu năng của KNN lại phục hồi rõ rệt, thậm chí có thể vượt Softmax trong một số trường hợp.

Về khả năng phân biệt lớp, KNN thể hiện độ nhạy cao hơn đối với lớp dương tính (class 1), với giá trị Recall cao hơn, cho thấy mô hình có khả năng phát hiện các mẫu dương tốt hơn, dù đôi khi phải đánh đổi bằng Precision thấp hơn. Ngược lại, Softmax Regression có xu hướng dự đoán ổn định hơn giữa các lớp, nhưng đôi khi bỏ sót một số mẫu dương, thể hiện qua Recall thấp hơn và tỉ lệ False Negative cao hơn.

Cuối cùng, xét về hiệu quả tính toán, Softmax Regression có ưu thế rõ rệt nhờ thời gian huấn luyện nhanh, khả năng mở rộng tốt và tiêu tốn ít tài nguyên trong giai đoạn dự đoán, phù hợp với các hệ thống lớn hoặc các bài toán thời gian thực. Trong khi đó, KNN không cần giai đoạn huấn luyện phức tạp nhưng lại tốn kém chi phí tính toán khi dự đoán vì phải tính khoảng cách đến toàn bộ tập huấn luyện. Điều này làm cho KNN kém phù hợp với các tập dữ liệu có kích thước lớn hoặc yêu cầu phản hồi nhanh.

Tóm lại, có thể thấy rằng KNN phù hợp hơn với những bài toán có quy mô dữ liệu vừa phải, yêu cầu độ chính xác cao và khả năng phát hiện mẫu dương nhạy, trong khi Softmax Regression lại là lựa chọn tối ưu cho các ứng dụng đòi hỏi tốc độ, khả năng triển khai nhanh và hoạt động tốt trên không gian dữ liệu đã được giảm chiều hiệu quả như LDA.

# TỔNG KẾT

Trong khuôn khổ bài báo cáo này, chúng em đã triển khai mô hình chuẩn đoán sớm khả năng mắc bệnh ở bệnh nhân, sử dụng bộ dữ liệu "alzheimers\_disease\_data" từ Kaggle. Quá trình nghiên cứu được thực hiện đầy đủ và có hệ thống từ tiền xử lý dữ liệu, huấn luyện mô hình đến đánh giá kết quả, với mục tiêu tối ưu hóa, làm chậm tiến trình bệnh và cải thiện chất lượng sống cho bệnh nhân.

Cụ thể, chúng em đã áp dụng hai phương pháp phân loại chính là K-Nearest Neighbors (KNN) và Softmax Regression để dự đoán khả năng mắc bệnh của bệnh nhân. Kết quả thử nghiệm cho thấy KNN có độ chính xác và hiệu quả cao hơn Softmax Regression, đặc biệt khi đánh giá qua các chỉ số như accuracy, precision, recall và F1-score.

Mặc dù Softmax Regression có lợi thế về lý thuyết và tính toán, nhưng KNN tỏ ra ổn định và hiệu quả hơn trong việc dự đoán, đặc biệt trong bối cảnh dữ liệu phức tạp và có sự biến động lớn giữa các thuộc tính. Mô hình KNN đã vượt trội hơn Softmax về mặt chỉ số hiệu quả nhờ vào khả năng học linh hoạt và xử lý tốt các đặc trưng phức tạp trong dữ liệu. Các yếu tố như nguy cơ mắc bệnh, đánh giá chức năng, nhận thức, chỉ số sinh học đã được mô hình nhận diện rõ ràng và chính xác. Điều này minh chứng cho việc KNN phù hợp với các bài toán có đặc trưng không gian phân lớp phức tạp.

Tuy báo cáo đã đạt được những kết quả khả quan trong việc xây dựng mô hình dự đoán khả năng mắc bệnh, nhưng vẫn còn một số hạn chế nhất định cần được xem xét trong các nghiên cứu tiếp theo. Đầu tiên, bộ dữ liệu sử dụng tuy phong phú nhưng vẫn có thể chưa đầy đủ, dẫn đến khả năng mô hình bị lệch nếu áp dụng trên dữ liệu mới. Thứ hai, việc lựa chọn chỉ hai mô hình đơn giản (KNN và Softmax Regression) phần nào giới hạn tiềm năng khai thác sâu hơn các mối quan hệ phi tuyến và phức tạp giữa các đặc trưng. Ngoài ra, mặc dù đã áp dụng hai kỹ thuật giảm chiều phổ biến là PCA và LDA, nhưng việc so sánh ảnh hưởng của từng kỹ thuật lên hiệu suất mô hình vẫn còn mang tính định tính, cần được phân tích sâu hơn.

Trong tương lai, hướng phát triển tiếp theo có thể bao gồm việc mở rộng tập dữ liệu, thử nghiệm thêm các mô hình học sâu (deep learning), áp dụng các kỹ thuật chọn đặc trưng nâng cao, và triển khai mô hình trong môi trường thực tế để đánh giá tính ứng dụng và khả năng thích nghi của mô hình trong các tình huống đa dạng và phức tạp hơn.

# LỜI CẢM ƠN

Cuối cùng, chúng em xin gửi lời cảm ơn sâu sắc đến thầy Cao Văn Chung đã dành thời gian đọc và đánh giá về đề tài "Phân tích và dự đoán nguy cơ mắc bệnh Alzheimer".

Sự hỗ trợ và góp ý từ thầy không chỉ giúp nhóm chúng em hoàn thiện bài báo cáo mà còn thúc đẩy chúng em phát triển và tiến bộ hơn trong quá trình học tập. Chúng em rất biết ơn và trân trọng sự tận tâm trong giảng dạy mà thầy đã dành cho lớp.

Tuy cả nhóm đã cố gắng để hoàn thiện tiểu luận một cách trọn vẹn nhất, song không thể tránh khỏi vẫn còn những sai sót nhất định. Chúng em rất mong muôn sê nhận được những góp ý tích cực từ thầy để hoàn thiện hơn sản phẩm của mình.

Chúng em xin chân thành cảm ơn!

**Nhóm sinh viên thực hiện**

Hồ Huyền Trang  
Trần Thị Mai Anh  
Đỗ Thị Như Quỳnh

# Tài liệu tham khảo

- [1] Robert H. Shumway, David S. Stoffer - Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis  
<https://journalofbigdata.springeropen.com/articles/10.1186/s40537-024-00973-y>
- [2] Vinh Dinh Nguyen, KNN (Basic, Advanced Concepts and Its Applications), AI VIET NAM, [Date Unknown].  
<https://www.aivietnam.edu.vn>
- [3] Michael Franke, Judith Degen - The softmax function: Properties, motivation, and interpretation  
[https://alpslab.stanford.edu/papers/FrankeDegen\\_submitted.pdf](https://alpslab.stanford.edu/papers/FrankeDegen_submitted.pdf)
- [4] Bolin Gao, Lacra Pavel - On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning  
<https://arxiv.org/pdf/1704.00805>
- [5] Bài 6: K-nearest neighbors, Fundy, Jan 8, 2017.  
<https://machinelearningcoban.com/2017/01/08/knn/>
- [6] Abdullah al-Mamun, K-Láng giềng gần nhất (KNN), SlideShare, Dec 16, 2023.  
<https://www.slideshare.net/slideshow/knearest-neighborsknn/264679060>
- [7] “Machine learning căn bản.”  
<https://machinelearningcoban.com>, 2024. Accessed: 2024-12-15.
- [8] k-nearest neighbors algorithm  
[https://en.wikipedia.org/wiki/K-nearest\\_neighbors\\_algorithm](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm)