Trường Đại học Khoa học Tự nhiên, ĐHQGHN
Khoa Toán – Cơ – Tin học

Phân tích và dự đoán nguy cơ mắc bệnh Alzheimer

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

Nội dung chính

Nội dung

Giới thiệu bài toán

- Bệnh Alzheimer là nguyên nhân hàng đầu gây sa sút trí tuệ, ảnh hưởng đến sức khỏe và gánh nặng kinh tế.
- Việc chẩn đoán sớm có ý nghĩa sống còn để can thiệp kịp thời.
- Bài toán tập trung vào việc phân tích, chuẩn đoán sớm để can thiệp kịp thời, 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,...

Mục tiêu của bài toán

- **Phân tích dữ liệu:** Khám phá và phân tích các yếu tố ảnh hưởng đến dự đoán mắc bệnh của bệnh nhân.
- **Tiền xử lý dữ liệu:** Áp dụng các kỹ thuật tiền xử lý dữ liệu 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 để dự đoán mắc bệnh của bênh nhân.
- **Phân tích yếu tố quyết định:** Phân tích và xác định các yếu tố quan trọng nhất thông qua việc đánh giá các trọng số của các thuộc tính trong mô hình dự đoán.

Nội dung

Mô hình K-NN

*** Cách KNN hoạt động**

Thuật toán KNN hoạt động dựa trên nguyên tắc dự đoán nhãn hoặc giá trị một điểm dữ liệu mới.

- **Bước 1:** Lựa chọn giá trị thích hợp cho K
- **Bước 2:** Tính toán khoảng cách
  - Khoảng cách Euclidean (p=2):

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

  - Khoảng cách Manhattan (p=1):

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

Mô hình K-NN

- Khoảng cách Minkowski:

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

- **Bước 3:** Xác định K hàng xóm gần nhất
- **Bước 4:** Dự đoán kết quả dựa trên Phân loại hoặc Hồi quy

Hồi quy Softmax

***Công thức của hàm Softmax**

- Chúng ta cần một mô hình xác suất để với mỗi đầu vào x, xác suất đầu ra của lớp i là z_i.
- Khi đó, để đảm bảo tổng các ai bằng 1, ta định nghĩa hàm Softmax (Softmax Function) như sau:

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

- Khi đó, xác suất để điểm dữ liệu x thuộc về lớp i được viết là:

$$P(y = i|x; W) = a_i$$

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

Hồi quy Softmax

***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.

Khi áp dụng mô hình Softmax Regression, với mỗi đầu vào x, ta tính đầu ra dự đoán bằng công thức:

$$\mathsf{a} = \mathrm{softmax}(\mathsf{W}^\top \mathsf{x})$$

Trong đó, a là xác suất dự đoán cho từng lớp. Ngược lại, đầu ra thực sự y được biểu diễn dưới dạng one-hot vector.

***Cross-Entropy**

Hồi quy Softmax

Cross-entropy là một hàm mất mát phổ biến trong bài toán phân loại nhiều lớp.

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

Trong đó:

- y_{ij} là thành phần thứ j trong vector y_i (xác suất thực sự)
- a_{ij} là thành phần thứ j trong vector a_i (xác suất dự đoán)
- W là ma trận trọng số của mô hình.

Hồi quy Softmax

***Hàm mất mát tổng quát cho SoftMax Regression**

Với tập dữ liệu gồm N cặp đầu vào (x_i, y_i) với i = 1,2,...,N khi thay a_{ij} bằng biểu thức Softmax, ta có:

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

Nội dung

Bộ dữ liệu

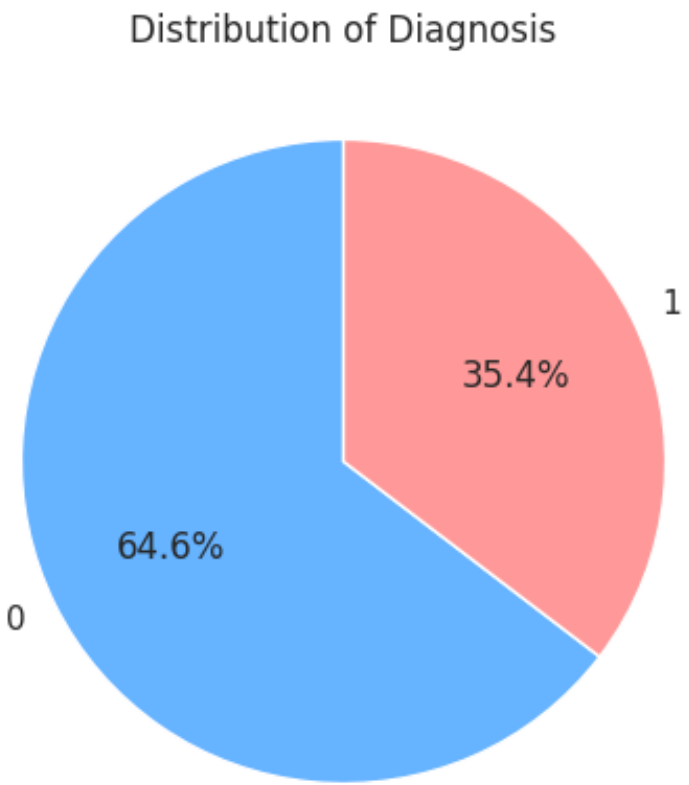
Gồm 34 trường về: thông tin về bệnh nhân, yếu tố lối sống, tiền sử y tế, các chỉ số lâm sàng, đánh giá nhận thức – chức năng và triệu chứng liên quan đến bệnh Alzheimer.
Các trường đều là giá trị số (trừ DoctorInCharge):

- 17 biến liên tục có các khoảng giá trị khác nhau.
- Hai trường phân loại là Ethnicity và EducationLevel.
- Ngoài ra còn có 15 trường nhị phân.

Hiểu & làm sạch dữ liệu

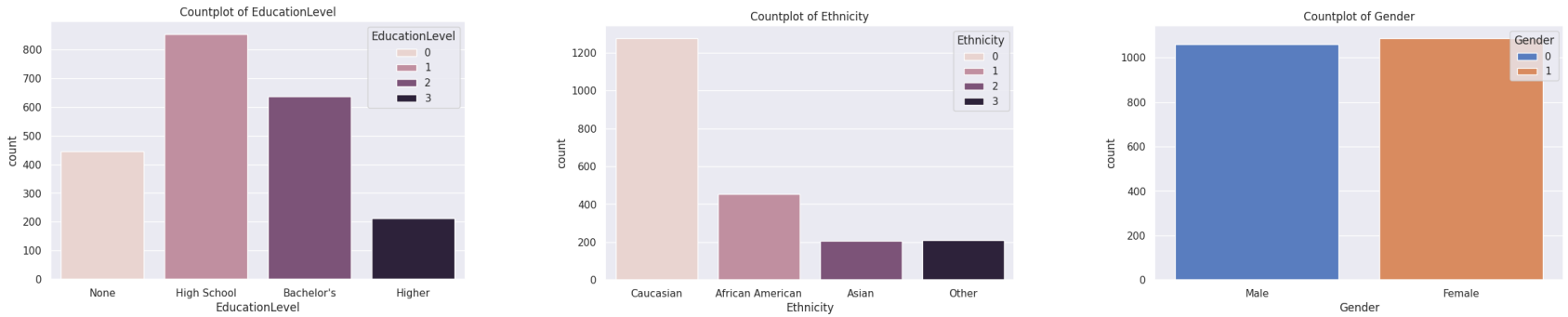
- Các giá trị đều là số.
- **Giá trị thiếu/ lặp:** Không.
- **Giá trị ngoại lai:** Không.
- **Các biến phân loại:** 17 biến.
  Trong đó, ngoại trừ EducationLevel và Ethnicity, các biến đều là nhị phân.
- **Các biến liên tục:** 15 biến với các khoảng phân bố khác nhau.
- Chúng ta sẽ bỏ 2 cột là PatientID và DoctorInCharge.

Trực quan hóa phân bố dữ liệu



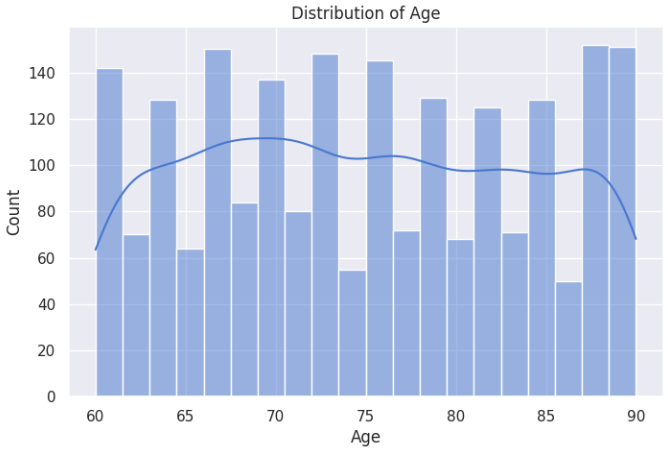
Hình: Phân bố của biến mục tiêu.

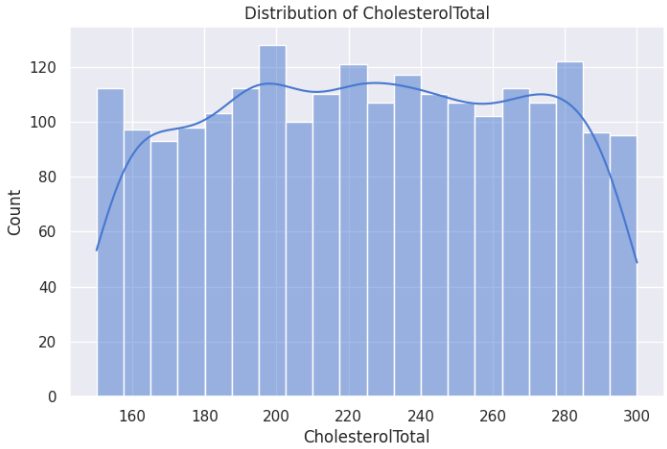
Phân bố một số biến phân loại

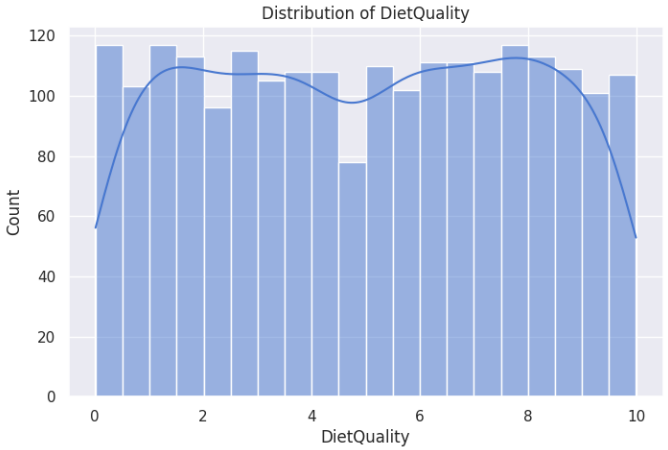


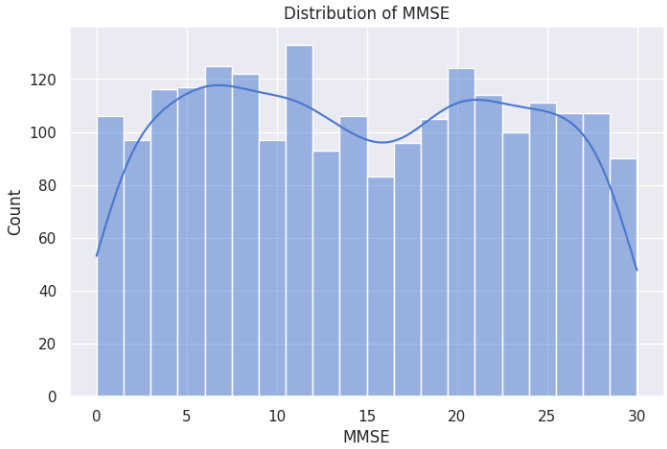
Hình: Phân bố của EducationLevel, Ethnicity, Gender.

Biến liên tục





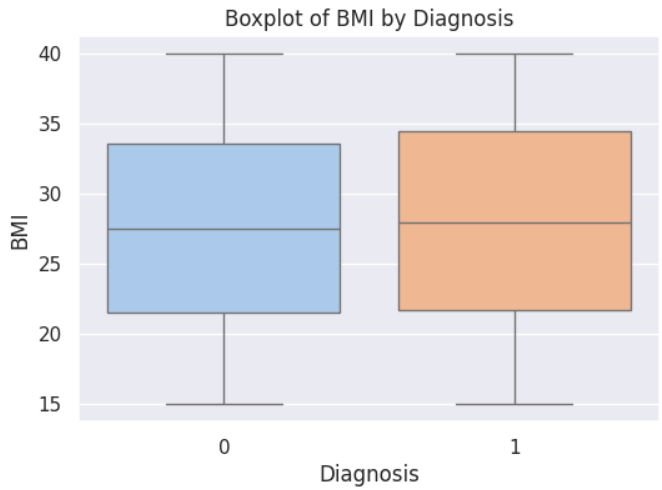


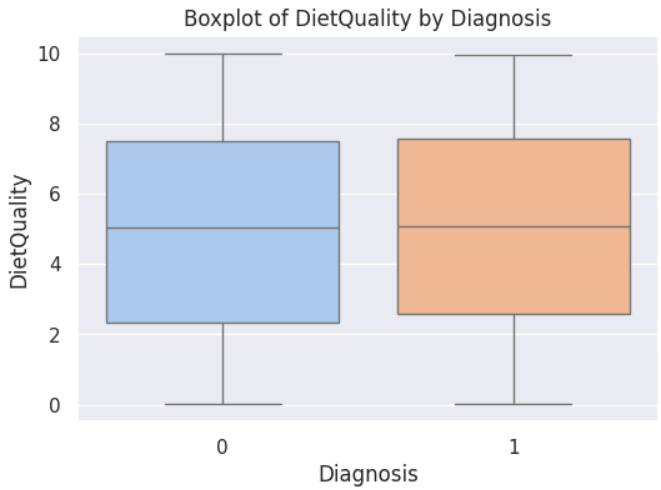


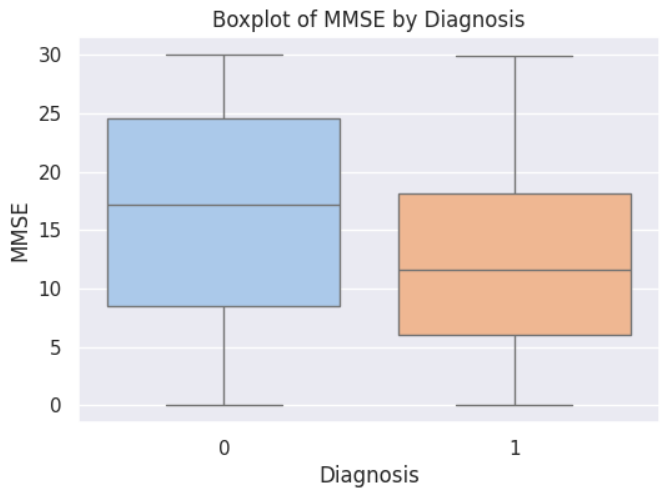
Hình: Phân bố của một số biến liên tục.

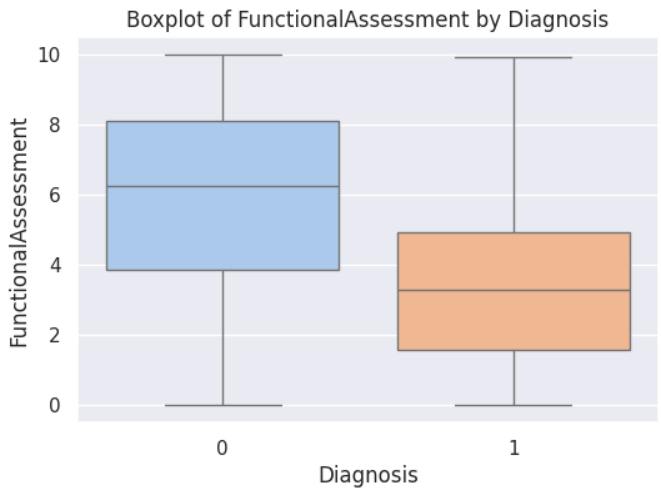
Biểu diễn đặc trưng theo biến mục tiêu

Phân bố của các biến liên tục





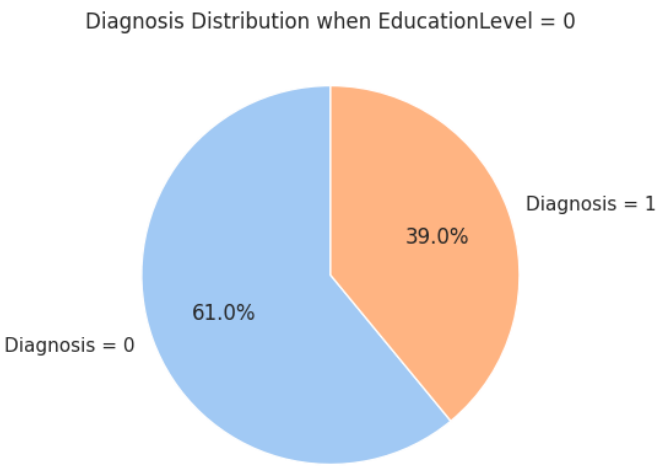


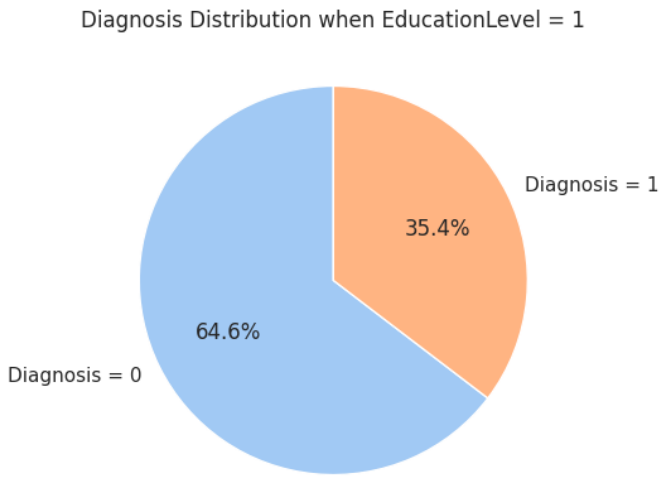


Biểu diễn đặc trưng theo biến mục tiêu

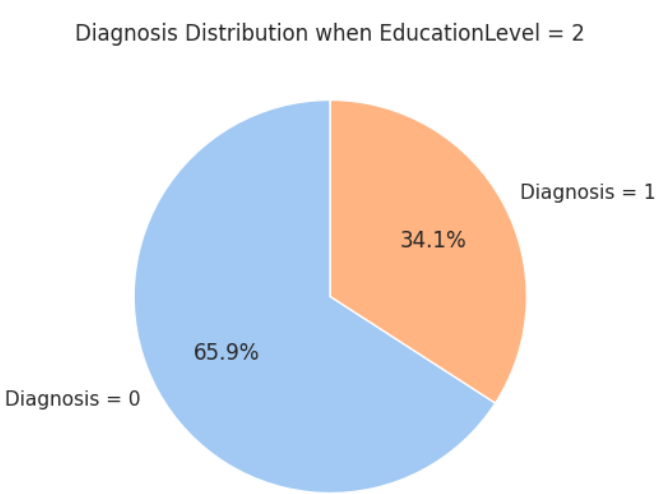
- Bệnh nhân Alzheimer có **MMSE**, **avg ADL** và **FunctionalAssessment** thấp hơn rõ rệt.
- Không có khác biệt đáng kể với các biến khác.

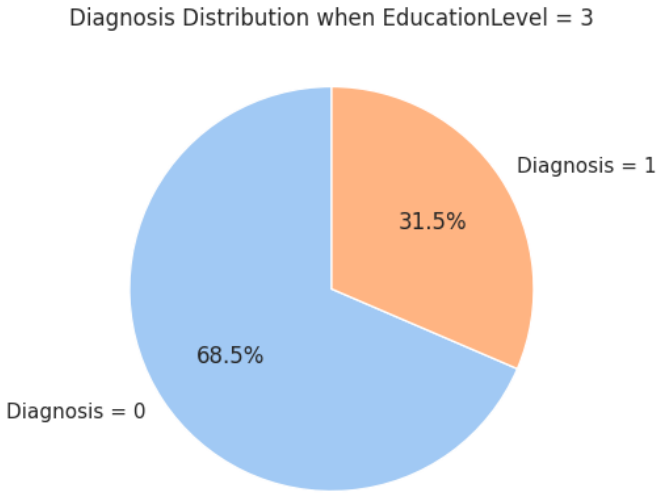
Phân bố với các biến phân loại





Biểu diễn đặc trưng theo biến mục tiêu





- 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.

Nội dung

Phân tích thành phần chính - PCA

- 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.

Chuẩn hóa:

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

Ma trận hiệp phương sai:

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

Phân tích Phân biệt Tuyến tính - LDA

Phương pháp giảm chiều có giám sát, tối đa hóa sự phân tách giữa các lớp.
Giả sử có c lớp với trung bình lớp $\boldsymbol{\mu}_i$ và trung bình toàn cục $\boldsymbol{\mu}$.

- Trong lớp:

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

- Giữa các lớp:

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

Phân tích Phân biệt Tuyến tính - LDA

Bài toán tối ưu:

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

Giải bài toán trị riêng tổng quát:

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

UMAP (Uniform Manifold Approximation and Projection)

Phương pháp giảm chiều dữ liệu phi tuyến, dựa trên lý thuyết hình học và tôpô.
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.

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 ρ_i là khoảng cách tới láng giềng gần nhất và σ_i được chọn sao cho mật độ được chuẩn hóa.

UMAP (Uniform Manifold Approximation and Projection)

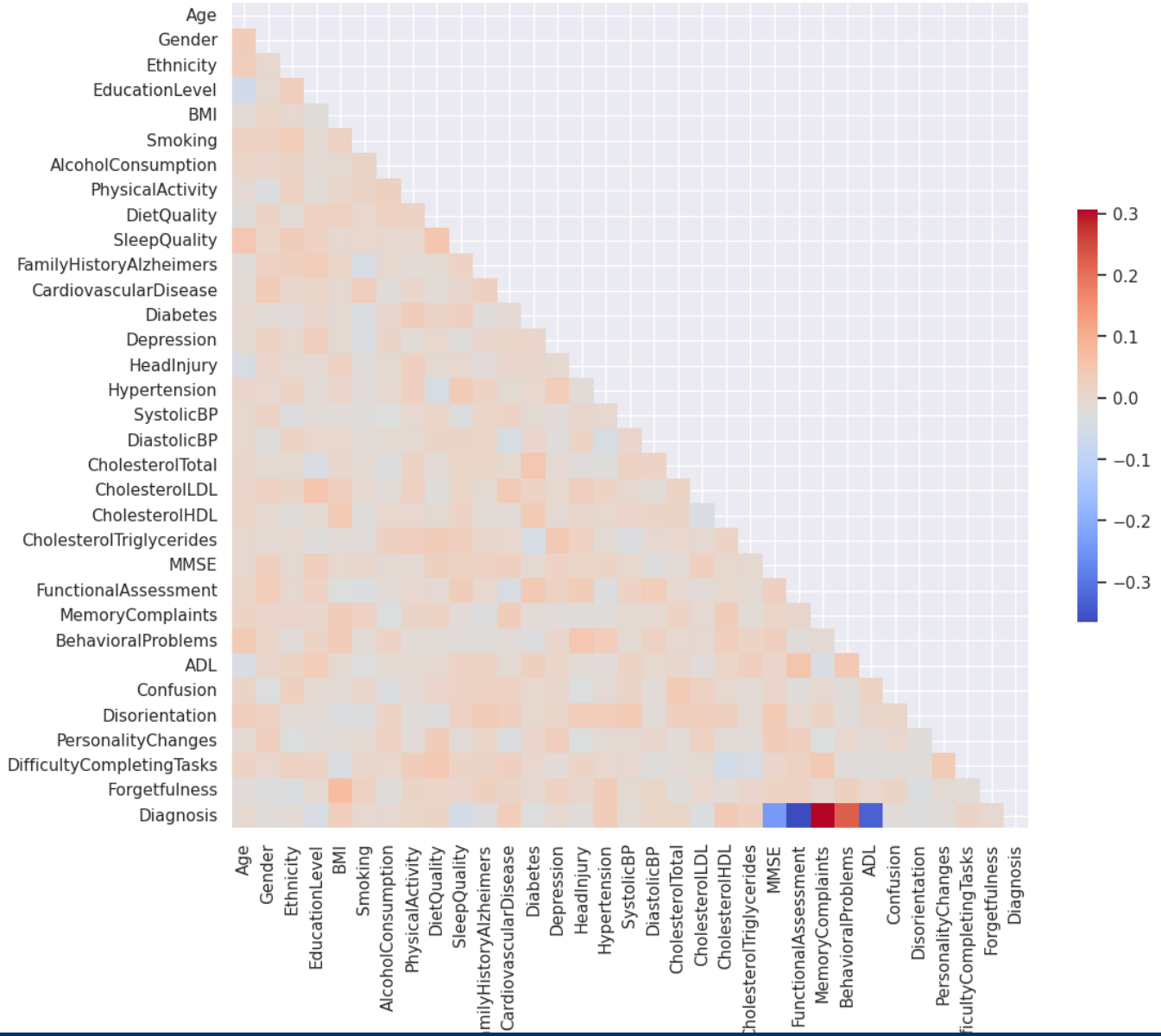
2. **Tối ưu đồ thị thấp chiều:** Xây dựng không gian nhúng mới bằng cách cực tiểu hóa độ mất mát cross-entropy giữa đồ thị gốc và đồ thị trong không gian thấp hơn.

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:

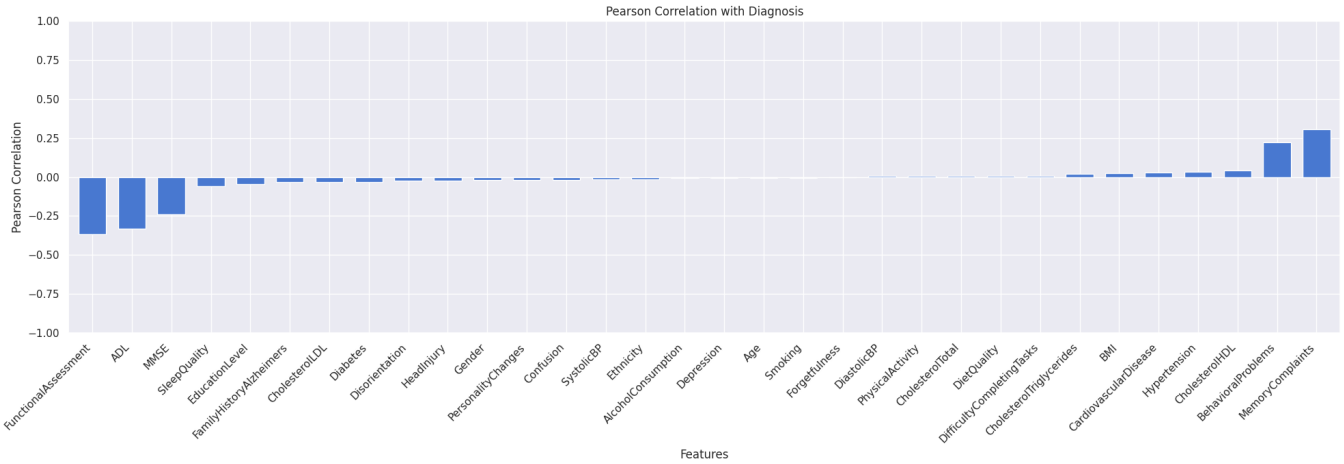
$$L = \sum_{i \neq j} \left[w_{ij} \log \frac{w_{ij}}{w'_{ij}} + (1 - w_{ij}) \log \frac{1 - w_{ij}}{1 - w'_{ij}} \right]$$

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

Tương quan

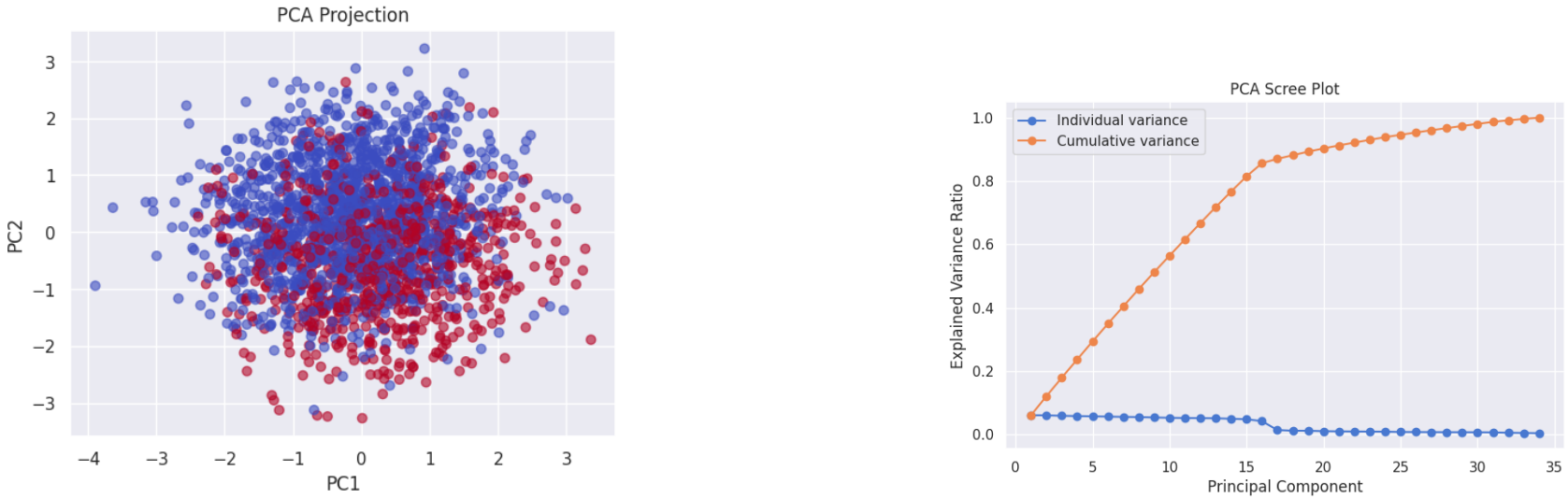

Age
Gender
Ethnicity
EducationLevel
BMI
Smoking
AlcoholConsumption
PhysicalActivity
DietQuality
SleepQuality
FamilyHistoryAlzheimers
CardiovascularDisease
Diabetes
Depression
HeadInjury
Hypertension
SystolicBP
DiastolicBP
CholesterolTotal
CholesterolLDL
CholesterolHDL
CholesterolTriglycerides
MMSE
FunctionalAssessment
MemoryComplaints
BehavioralProblems
ADL
Confusion
Disorientation
PersonalityChanges
DifficultyCompletingTasks
Forgetfulness
Diagnosis
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3

Tương quan



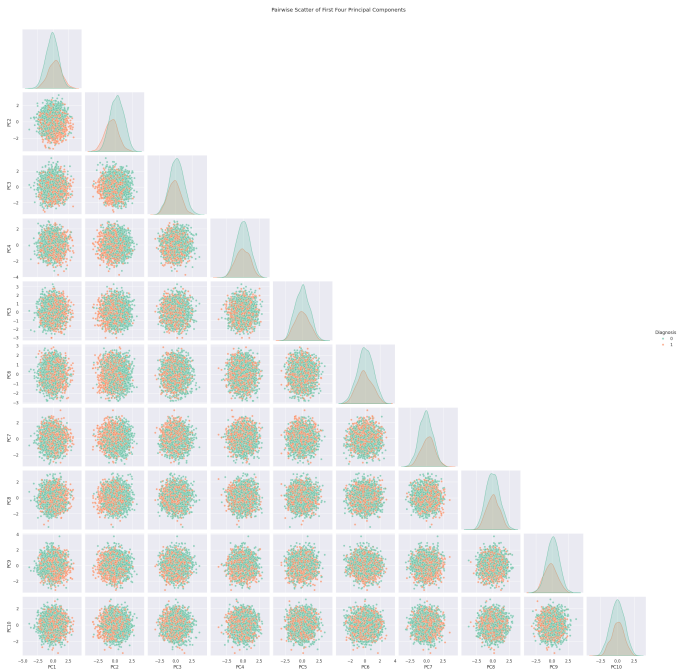
Hình: Barplot

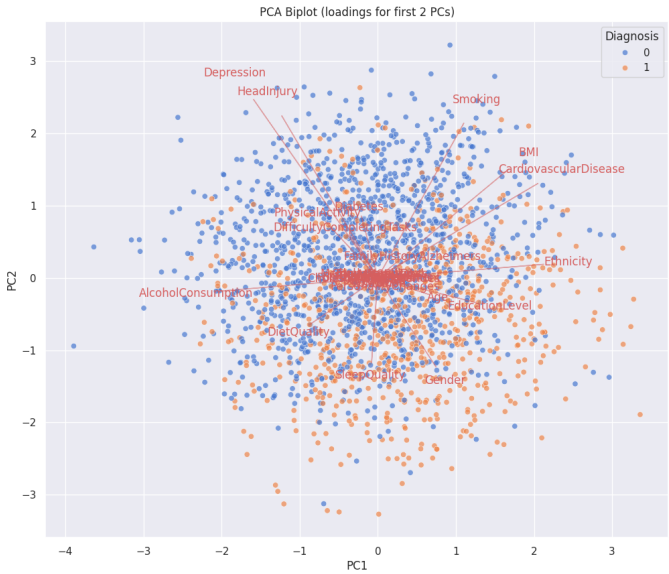
PCA



Hình: PCA results

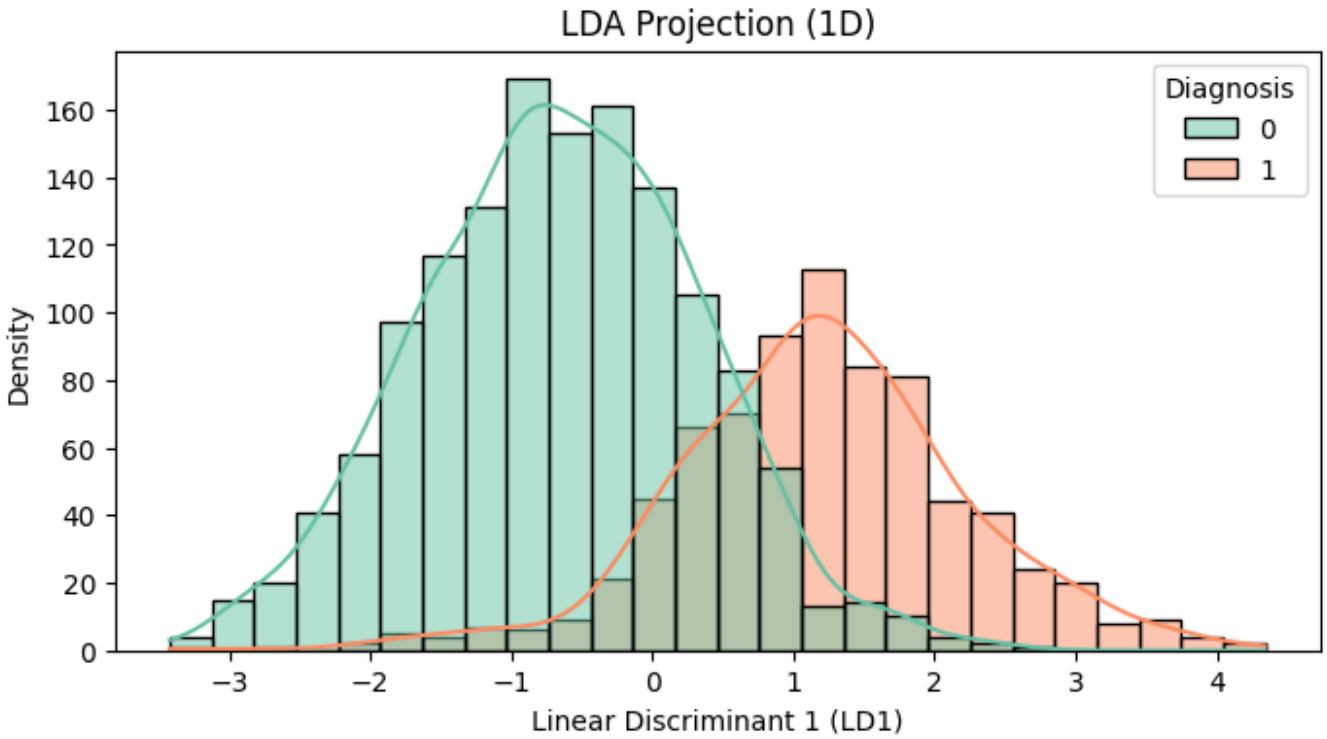
PCA





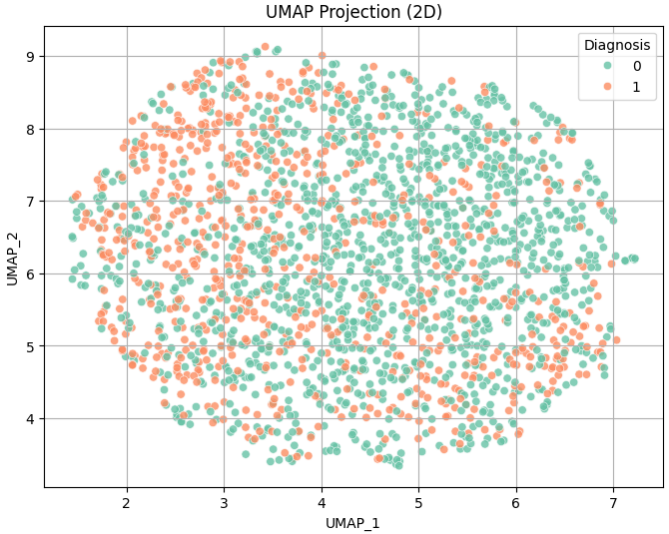
Hình: PCA results.

LDA

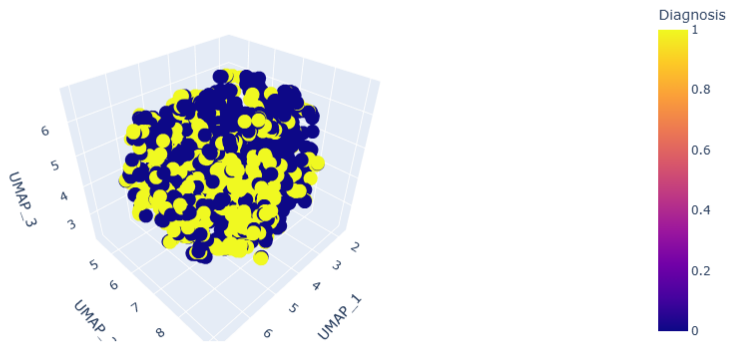


Hình: LDA result.

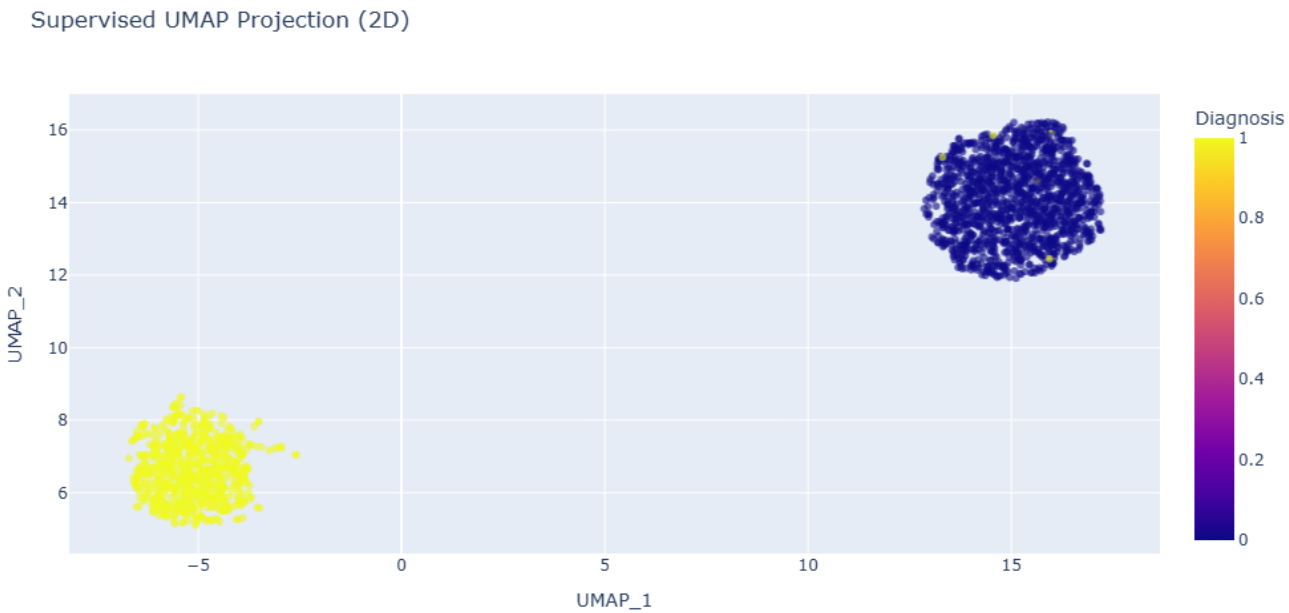
UMAP



UMAP Projection (3D) - n_neighbors=5, min_dist=0.05



Supervised UMAP



Hình: UMAP results

Nội dung

Mô hình K-NN

- **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: Bảng kết quả phân loại cho phân chia 4:1

Mô hình K-NN

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
Accuracy			0.89
Macro Avg	0.87	0.89	0.88
Weighted Avg	0.89	0.89	0.89

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

Mô hình K-NN

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
Accuracy			0.86
Macro Avg	0.84	0.85	0.85
Weighted Avg	0.86	0.86	0.86

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

Mô hình KNN 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. Tuy nhiên, độ chính xác tổng thể của mô hình vẫn ổn định ở mức cao (khoảng 89%).

Mô hình K-NN

- **Trên tệp dữ liệu 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: Bảng kết quả phân loại cho phân chia 4:1

Mô hình K-NN

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

Mô hình K-NN

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
Accuracy			0.69
Macro Avg	0.69	0.70	0.68
Weighted Avg	0.73	0.69	0.69

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

Việc giảm chiều dữ liệu bằng PCA đã không làm giảm quá nhiều hiệu suất của mô hình. Mô hình KNN cho thấy khả năng phân loại ổn định đối với lớp "0", tuy nhiên cần cải thiện khả năng phân loại lớp "1".

Mô hình K-NN

- **Trên tệp dữ liệu 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
Accuracy			0.81
Macro Avg	0.80	0.79	0.80
Weighted Avg	0.81	0.81	0.81

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

Mô hình K-NN

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
Accuracy			0.83
Macro Avg	0.81	0.81	0.81
Weighted Avg	0.83	0.83	0.83

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

Mô hình K-NN

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
Accuracy			0.83
Macro Avg	0.82	0.81	0.81
Weighted Avg	0.83	0.83	0.83

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

Việc giảm chiều dữ liệu bằng LDA giúp mô hình duy trì hiệu suất phân loại ổn định, tuy nhiên vẫn cần cải thiện khả năng phân loại lớp "1", đặc biệt khi có sự chồng chéo giữa các lớp. Độ chính xác tổng thể nhỏ hơn một chút so với dữ liệu gốc.

Mô hình Softmax regression

- **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
Accuracy			0.83
Macro Avg	0.81	0.82	0.81
Weighted Avg	0.83	0.83	0.83

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

Mô hình Softmax regression

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
Accuracy			0.84
Macro Avg	0.83	0.83	0.83
Weighted Avg	0.84	0.84	0.84

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

Mô hình Softmax regression

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
Accuracy			0.84
Macro Avg	0.82	0.82	0.82
Weighted Avg	0.84	0.84	0.84

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

Trên dữ liệu gốc, mô hình Softmax Regression đạt độ chính xác cao và ổn định (83-84%) ở các tỷ lệ chia khác nhau. Các chỉ số Precision, Recall và F1-score giữa hai lớp có sự chênh lệch nhẹ nhưng mô hình vẫn duy trì độ chính xác tổng thể tốt và khả năng phân loại ổn định.

Mô hình Softmax regression

- **Trên tệp dữ liệu 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
Accuracy			0.82
Macro Avg	0.80	0.80	0.80
Weighted Avg	0.82	0.82	0.82

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

Mô hình Softmax regression

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
Accuracy			0.84
Macro Avg	0.82	0.82	0.82
Weighted Avg	0.84	0.84	0.84

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

Mô hình Softmax regression

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
Accuracy			0.83
Macro Avg	0.82	0.82	0.82
Weighted Avg	0.83	0.83	0.83

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

Mặc dù hiệu suất của mô hình Softmax Regression sau khi giảm chiều bằng PCA có giảm nhẹ so với dữ liệu gốc, nhưng các chỉ số vẫn duy trì ở mức ổn định (82-84%) trên các tỷ lệ chia khác nhau.

Mô hình Softmax regression

- **Trên tệp dữ liệu giảm chiều LDA**

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

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

Mô hình Softmax regression

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

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

Mô hình Softmax regression

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

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

Sau khi giảm chiều bằng LDA, mô hình Softmax Regression vẫn giữ được hiệu suất gần tương đương với dữ liệu gốc, với độ chính xác ổn định khoảng 83-84%.