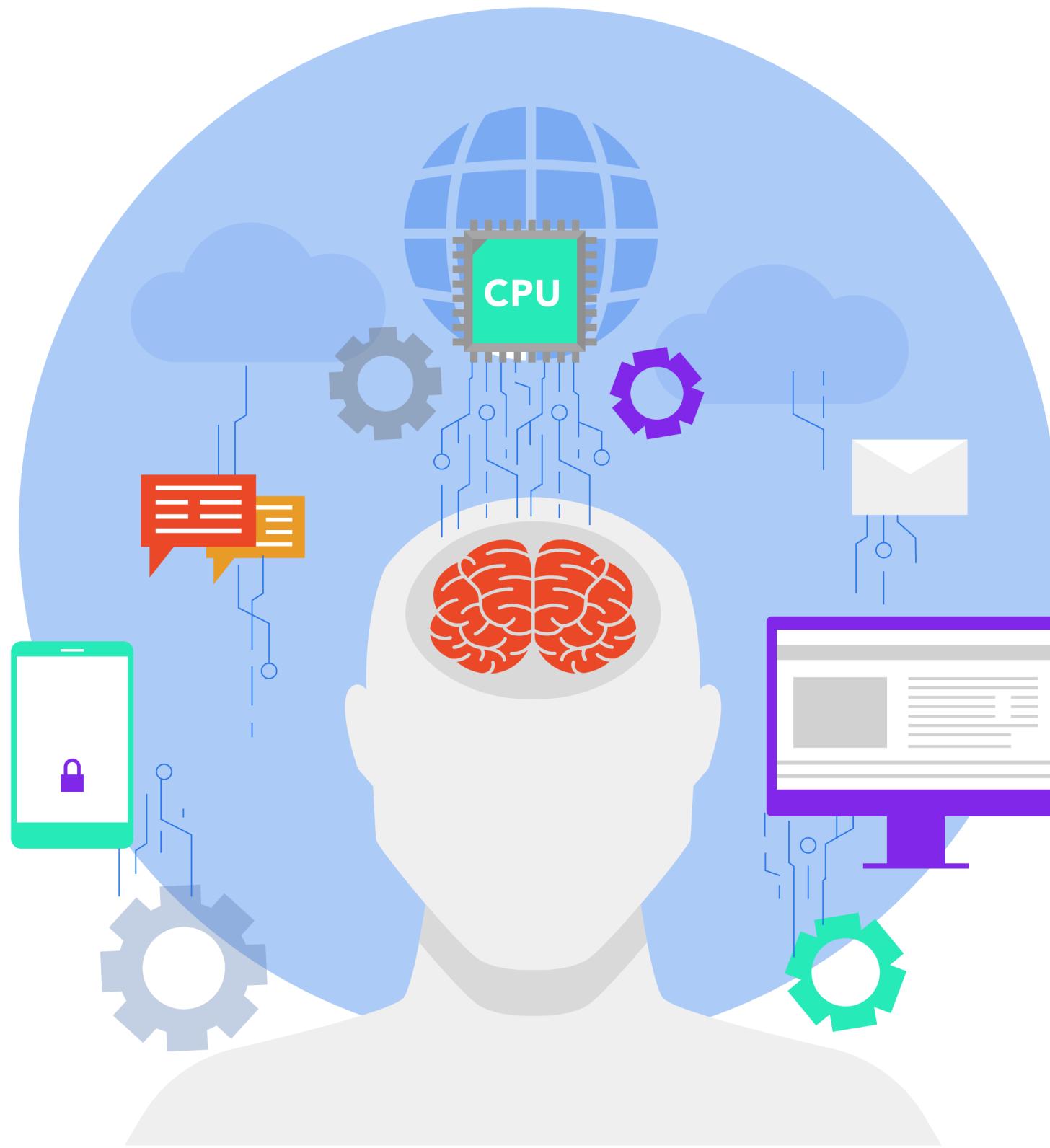


Dự đoán khả năng mắc rối loạn tự kỷ



| Thành viên:

Vũ Đình Hoàng - 23001881

Nguyễn Nhật Đan - 23001861

Hoàng Mạnh Duy - 23001852

Tổng quan về Rối loạn Tự kỷ (ASD)

ASD là gì?



Giao tiếp xã hội

Khó giao tiếp bằng mắt,
biểu cảm hạn chế, ít
chia sẻ cảm xúc

Hành vi lặp lại

Thường xuyên lặp động
tác (vỗ tay, đung đưa),
sắp xếp đồ vật cố định.

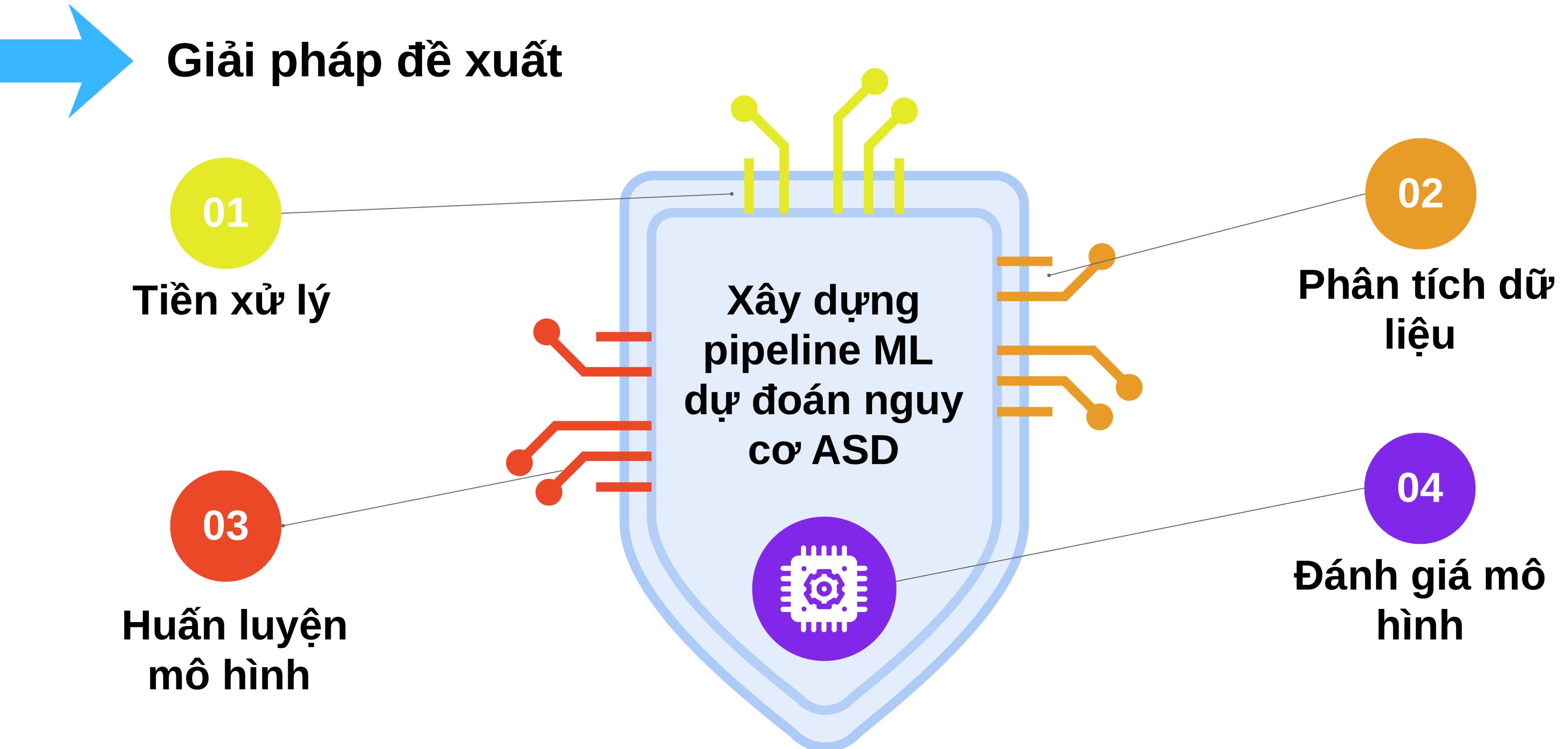
Kháng cự thay đổi

Phản ứng mạnh khi môi
trường hoặc thói quen
bị thay đổi.

Giác quan bất
thường

Quá nhạy hoặc kém
nhạy với âm thanh, ánh
sáng, nhiệt độ.

Tổng quan về Rối loạn Tụ kỷ (ASD)



Tập dữ liệu

01

Nguồn

Bộ dữ liệu ASD Screening Adults gồm 800 bản ghi, được thu thập từ bảng khảo sát sàng lọc tự kỷ cho người trưởng thành.

02

Cấu trúc & Đặc trưng

22 cột → 21 đặc trưng đầu vào (câu hỏi trắc nghiệm AQ, thông tin cá nhân, tiền sử bệnh, hành vi) + 1 nhãn mục tiêu (ASD / không ASD).

```
# đọc dữ liệu
df = pd.read_csv("../data/raw/train.csv")

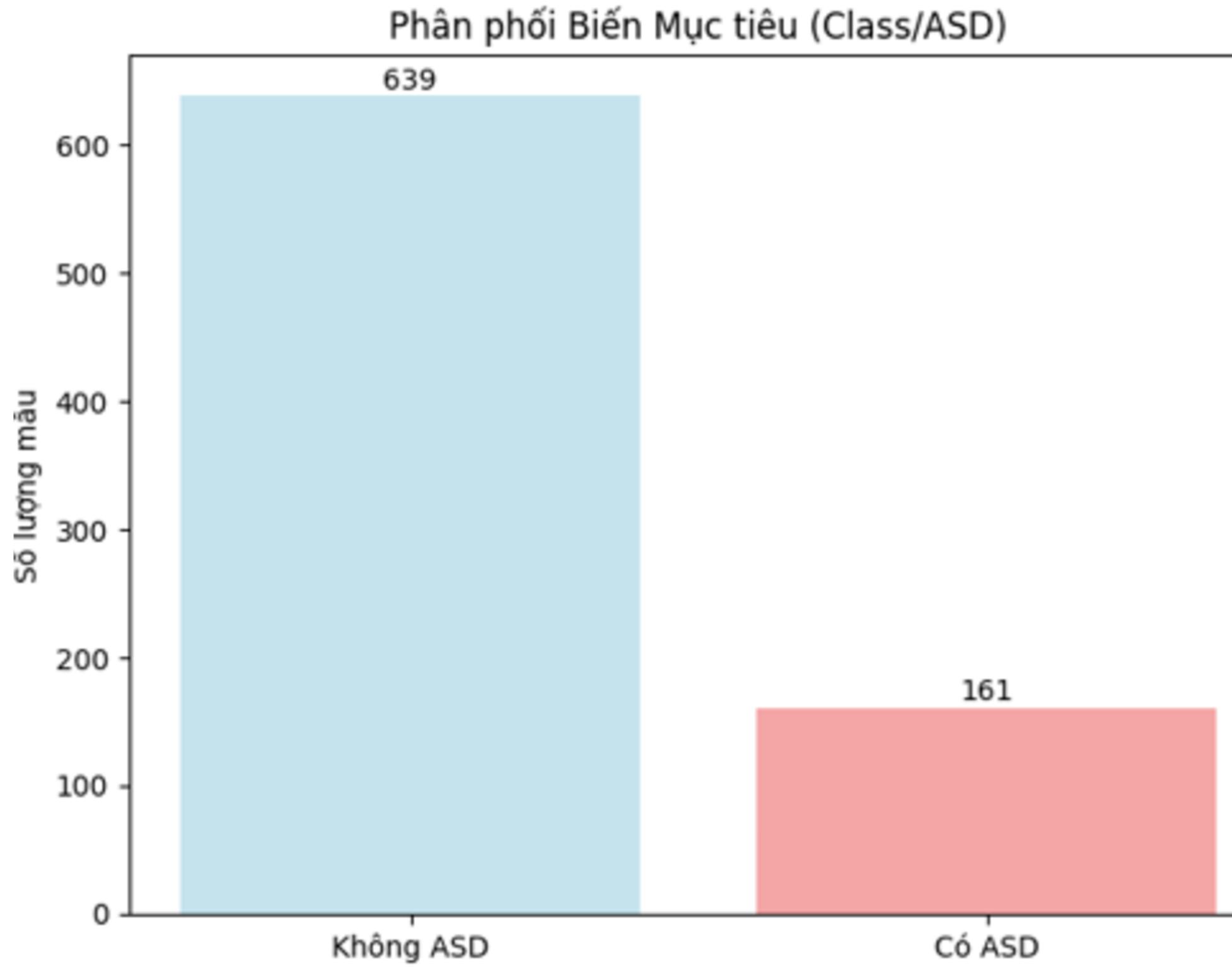
# Hiển thị thông tin cơ bản về dataset
print("\n1. THÔNG TIN CƠ BẢN VỀ DATASET")
print("-" * 40)
print(f"Kích thước dataset: {df.shape}")
print(f"Số lượng mẫu (samples): {df.shape[0]}")
print(f"Số đặc trưng (features): {df.shape[1]}")
```

✓ 0.0s

1. THÔNG TIN CƠ BẢN VỀ DATASET

Kích thước dataset: (800, 22)
Số lượng mẫu (samples): 800
Số đặc trưng (features): 22

Tập dữ liệu



Features:

- **AQ Score Features (10 features):** ['A1_Score', 'A2_Score', 'A3_Score', 'A4_Score', 'A5_Score', 'A6_Score', 'A7_Score', 'A8_Score', 'A9_Score', 'A10_Score']
- **Demographic Features (10 features):** ['age', 'gender', 'ethnicity', 'jaundice', 'austim', 'contry_of_res', 'used_app_before', 'result', 'age_desc', 'relation']
- **Target Feature:** ['Class/ASD']

Xử lý Missing Values đặc biệt hoặc dữ liệu sai, không đúng định dạng

Dữ liệu cần tiền xử lý:

- ethnicity: chứa giá trị '?', 'Others', 'others' → cần chuẩn hóa
- jaundice, austim, used_app_before: giá trị 'yes' / 'no'
- contry_of_res: nhiều tên quốc gia, có lỗi chính tả như 'Viet Nam', 'AmericanSamoa' → cần đồng nhất
- age_desc: phân nhóm 'child', 'adult', 'teen', 'elderly'
- relation: có '?', 'Others', 'Self', 'Parent' → cần thay thế hoặc gộp nhóm



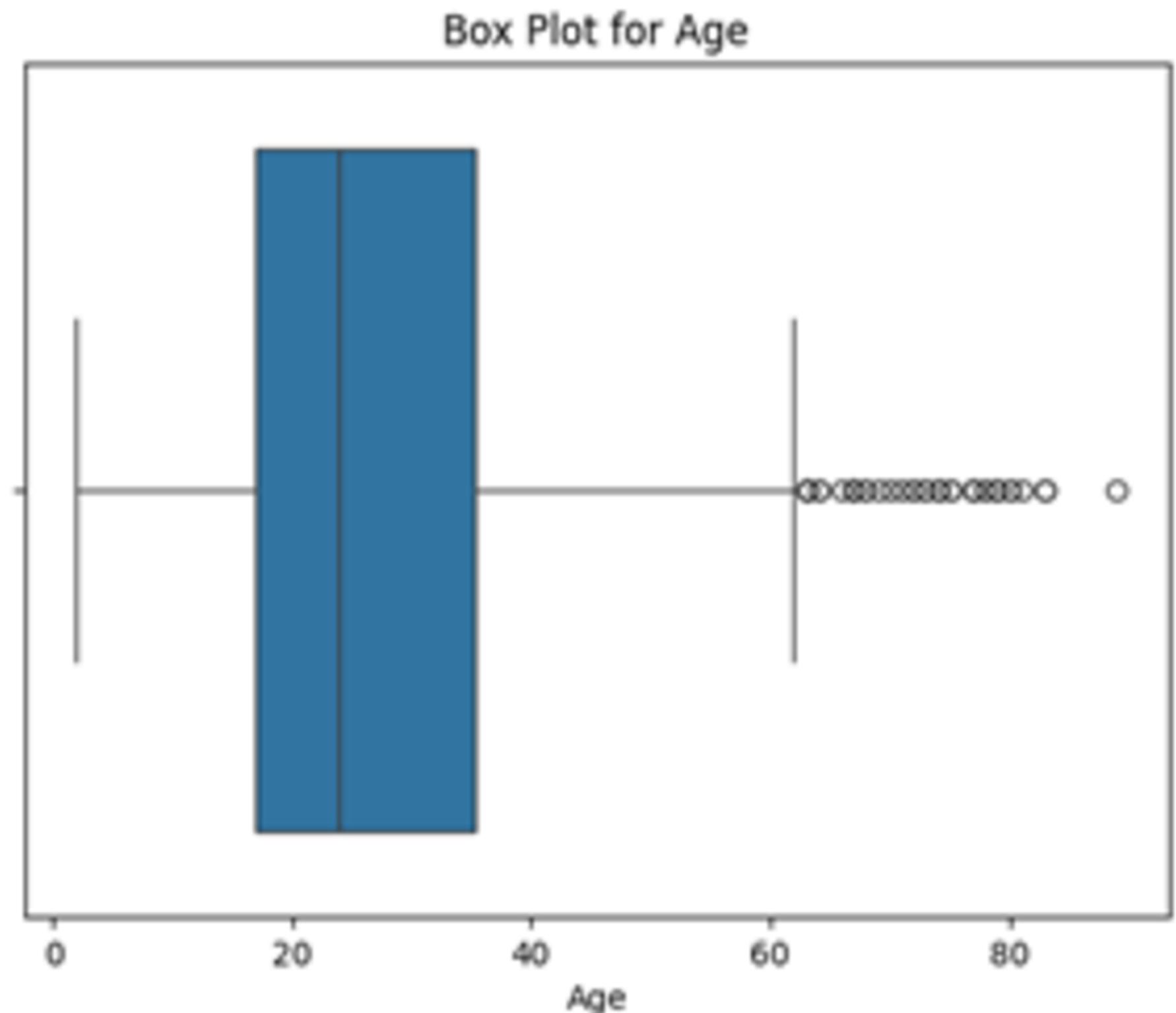
```
# Thay thế các giá trị sai định dạng hoặc đồng nghĩa trong các cột
df["ethnicity"] = df["ethnicity"].replace({"?": "Others", "others": "Others"})
df["relation"] = df["relation"].replace({"?": "Others"})

mapping = {
    "Viet Nam": "Vietnam",
    "AmericanSamoa": "United States",
    "Hong Kong": "China"
}
df["contry_of_res"] = df["contry_of_res"].replace(mapping)
```

Tiền xử lý dữ liệu

Xử lý các giá trị ngoại lai (outliers)

- Khoảng giá trị chính (IQR) nằm chủ yếu từ khoảng ~18 đến ~40 tuổi.
- Có **nhiều điểm nằm phía bên phải**, xấp xỉ trong khoảng 60–90 tuổi, bị đánh dấu là **outliers**.
- Những outliers này **xuất hiện tập trung**, không rải rác đơn lẻ → cho thấy đây có thể là **nhóm người lớn tuổi hợp lệ**.



Biến đổi dữ liệu

Attribute classification

- Numerical: age, result
- Binary: A1–A10, jaundice, austim, used_app_before, Class/ASD
- Categorical: gender, ethnicity, country_of_res, relation, age_desc

Categorical Features

- Ordinal features: age_desc sử dụng ordinal encode
- Nominal Categorical: ethnicity, contry_of_res, relation sử dụng target encode còn gender, jaundice, austim, used_app_before còn sử dụng ordinal encode
- Mục tiêu: mã hóa dữ liệu chữ thành số, giữ thông tin phân biệt

Binary Features

- Biến nhị phân (0/1, Yes/No)
- Giữ nguyên, không cần mã hóa → passthrough

01

Numerical features

- Bổ khuyết giá trị thiếu bằng median
- Chuẩn hóa dữ liệu bằng Z-score (StandardScaler)
- Mục tiêu: giảm ảnh hưởng ngoại lệ, làm mô hình hội tụ nhanh hơn

03

Preprocessor

- Dùng ColumnTransformer tích hợp toàn bộ pipeline:
- Median + StandardScaler
- Ordinal / Target Encoding
- Passthrough cho Binary
- Chỉ fit trên tập train → tránh data leakage

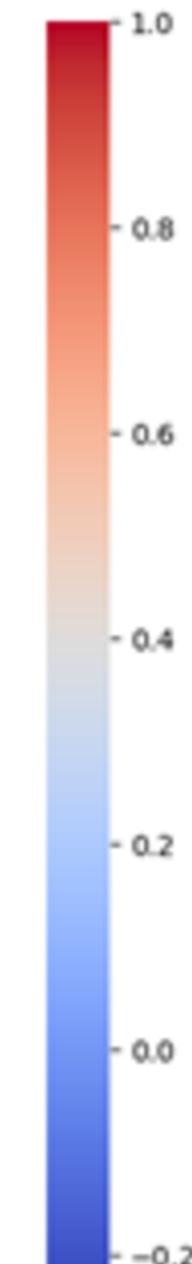
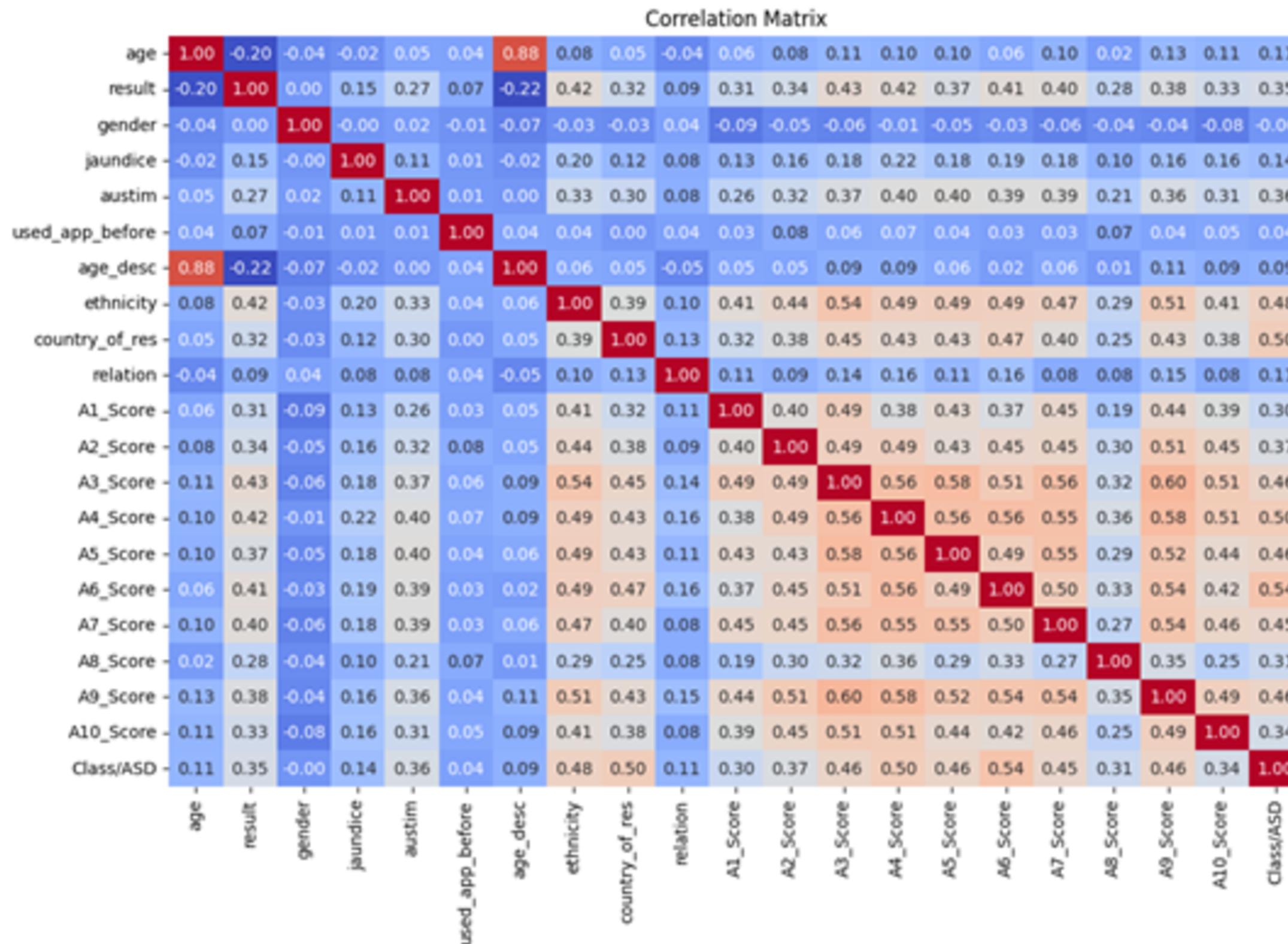
05

02

04

Phân tích và trực quan hóa dữ liệu

Trực quan hóa một số thông tin trước khi giảm chiều



Ma trận tương quan giữa các features và target

Phân tích và trực quan hóa dữ liệu

Nhận xét

A1–A10 tương quan cao với nhau (0.4–0.7) và với Class/ASD (0.3–0.55) → nhóm câu hỏi có ý nghĩa phân loại tốt.

age ↔ age_desc tương quan 0.88 → trùng thông tin.

ethnicity ↔ country_of_res tương quan ~0.5 → có thể dữ thừa.

Giải pháp

Giữ A1–A10 nhưng có thể giảm chiều (PCA).
Bỏ age_desc nếu đã có age.

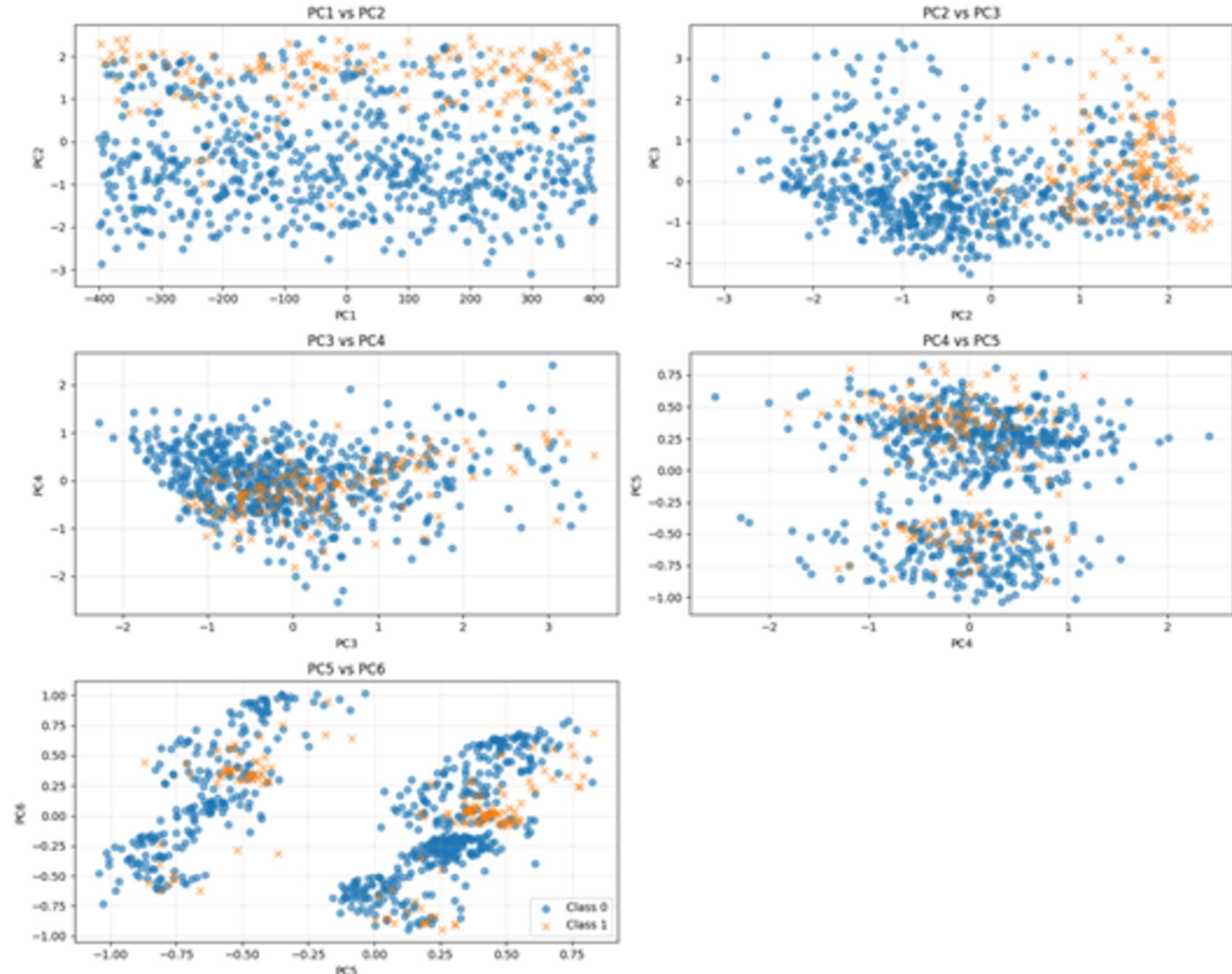
Phân tích và trực quan hóa dữ liệu

PCA – Giảm chiều dữ liệu không giám sát

PC1 vs PC2: Thể hiện sự phân tách rõ rệt nhất giữa các lớp

Các PC từ 4-6: Thể hiện sự chồng chập đáng kể giữa các lớp

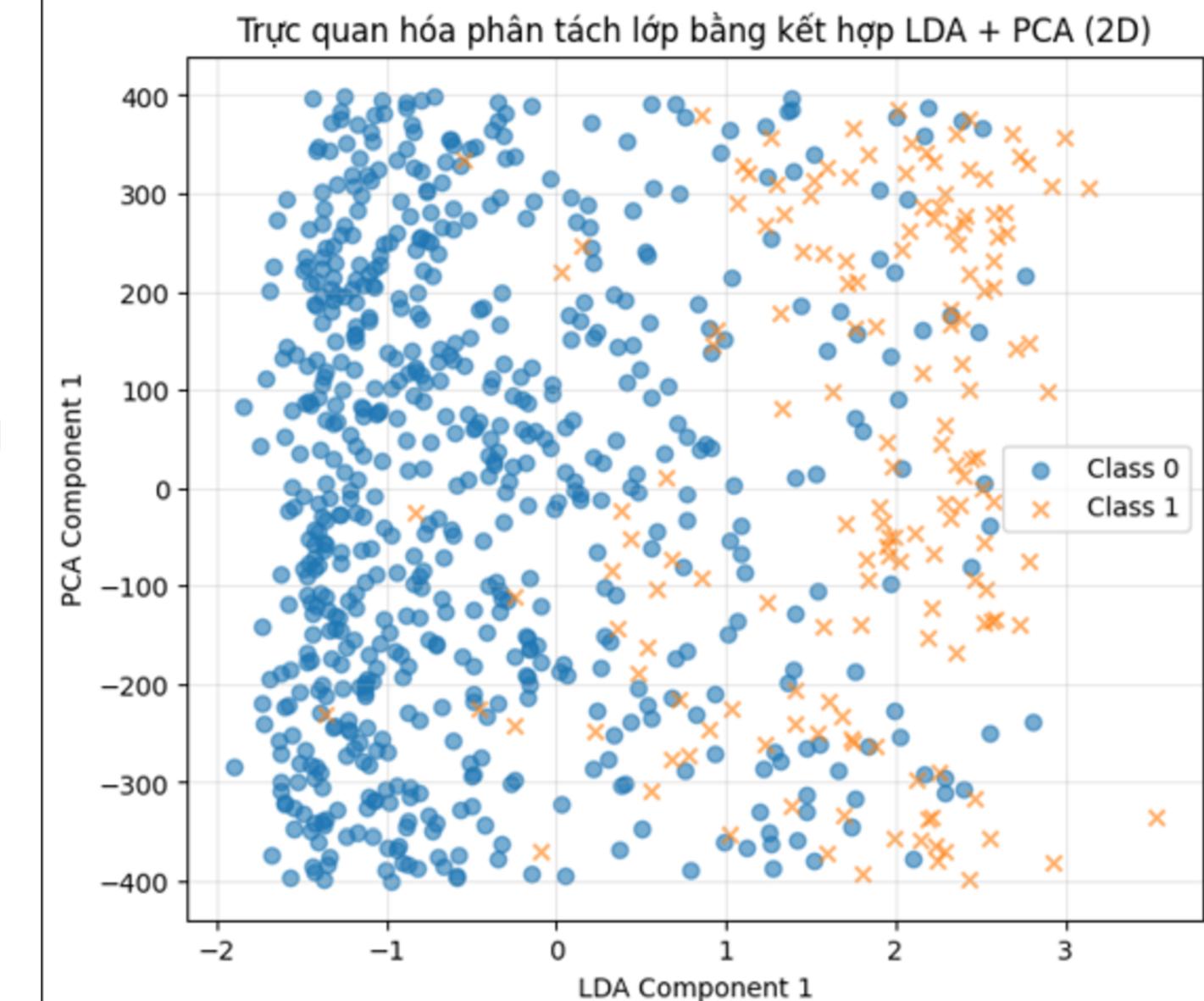
Tuy nhiên, sự phân tách giữa hai lớp vẫn chưa rõ ràng, → cần dùng LDA để cải thiện.



Phân tích và trực quan hóa dữ liệu

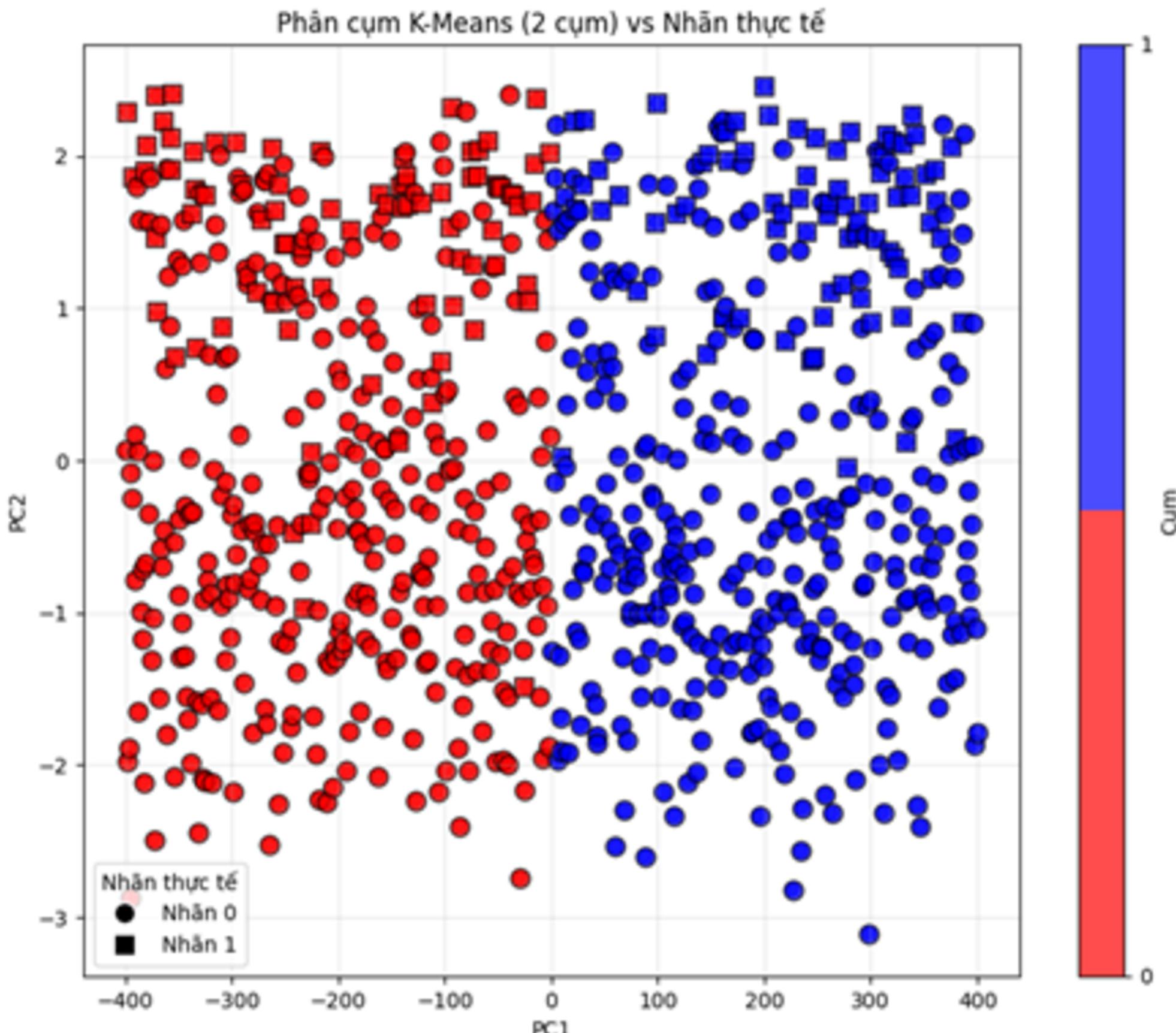
LDA – Giảm chiều dữ liệu có giám sát

- Mô hình chỉ tạo ra một thành phần phân biệt (LD1) do dữ liệu có hai lớp phân loại.
- Thành phần này giữ lại 100% thông tin phân lớp (explained variance = 1.0), cho thấy LDA đã tối ưu hóa khả năng phân tách giữa nhóm “Class 0 (Không ASD)” và “Class 1 (ASD)”.
- Giá trị trung bình xấp xỉ bằng 0 và độ lệch chuẩn khoảng 1.34 chứng tỏ dữ liệu được chuẩn hóa tốt và có sự phân tách rõ rệt trên trục LDA, làm cho LDA trở thành phương pháp hiệu quả hơn PCA trong việc quan sát cấu trúc phân loại của bộ dữ liệu này.



Phân cụm dữ liệu

- Áp dụng thuật toán **K-Means** để chia dữ liệu thành 2 cụm ($k = 2$).
- **Elbow Method** chọn $k = 2$ là tối ưu → phù hợp với số lớp thực tế (ASD / Không ASD).
- **Silhouette Score** = 0.6255 → cho thấy **cấu trúc cụm tách biệt tương đối tốt**.
- Hai cụm có kích thước cân bằng, phản ánh **hai nhóm hành vi rõ rệt**.
- Tuy không trùng hoàn toàn với nhãn Class/ASD, nhưng cho thấy dữ liệu có **phân tách tự nhiên**.



Phân loại

Khái niệm

- Là thuật toán học có giám sát, dựa trên khoảng cách giữa các điểm dữ liệu.
- Mỗi điểm mới được gán nhãn theo đa số K láng giềng gần nhất.

Cách hoạt động

- Tính khoảng cách Euclidean giữa điểm mới và các điểm huấn luyện.
- Chọn $K = 20$, trọng số theo khoảng cách (điểm gần → ảnh hưởng lớn hơn).

K-nearest neighbor

```
# Sử dụng KNN với k=20, p=2 chuẩn L2 norm, weights='distance'
knn_model = KNeighborsClassifier(n_neighbors=20, p=2, weights='distance')

# Tỉ lệ 4:1
model_with_and_without_LDA(X, y, 0.2, knn_model)
print()
# Tỉ lệ 7:3
model_with_and_without_LDA(X, y, 0.3, knn_model)
print()
# Tỉ lệ 6:4
model_with_and_without_LDA(X, y, 0.4, knn_model)

✓ 0.2s

---Chia theo tỉ lệ: 0.2---
Độ chính xác chéo với dữ liệu gốc: 0.8757281553398059
Độ chính xác với dữ liệu gốc: 0.79375
Độ chính xác với dữ liệu đã giảm chiều: 0.825

---Chia theo tỉ lệ: 0.3---
Độ chính xác chéo với dữ liệu gốc: 0.8741545747070608
Độ chính xác với dữ liệu gốc: 0.7958333333333333
Độ chính xác với dữ liệu đã giảm chiều: 0.8541666666666666

---Chia theo tỉ lệ: 0.4---
Độ chính xác chéo với dữ liệu gốc: 0.8710720651897124
Độ chính xác với dữ liệu gốc: 0.784375
Độ chính xác với dữ liệu đã giảm chiều: 0.821875
```

Thực nghiệm

- Dữ liệu được tiền xử lý và giảm chiều bằng LDA.
- Huấn luyện với tỉ lệ train/test = 0.2–0.4.
- Cross-validation ~0.87, mô hình ổn định, không overfitting.

Kết quả

- Accuracy sau LDA: ~0.85, cải thiện so với dữ liệu gốc (~0.79).
- Phân loại cân bằng hai lớp ASD / Non-ASD.
- Sai số chủ yếu ở vùng ranh giới chồng chéo đặc trưng.

Logistic regression

Khái niệm

Mô hình phân loại nhị phân (0/1), ước lượng xác suất mẫu thuộc lớp 1 qua hàm sigmoid:

$$P(y = 1|x) = 1/(1 + e^{-(w^T x + b)})$$

Hàm mất mát (Log-loss)

Đo sai khác giữa dự đoán và thực tế, tối ưu bằng cách tìm trọng số w để $J(w)$ nhỏ nhất.

```
lr_model = LogisticRegression(max_iter=1000)

# Tỉ lệ 4:1
model_with_and_without_LDA(X, y, 0.2, lr_model)
print()
# Tỉ lệ 7:3
model_with_and_without_LDA(X, y, 0.3, lr_model)
print()
# Tỉ lệ 6:4
model_with_and_without_LDA(X, y, 0.4, lr_model)
✓ 0.1s

---Chia theo tỉ lệ: 0.2---
Độ chính xác chéo với dữ liệu gốc: 0.8844660194174757
Độ chính xác với dữ liệu gốc: 0.825
Độ chính xác với dữ liệu đã giảm chiều: 0.84375

---Chia theo tỉ lệ: 0.3---
Độ chính xác chéo với dữ liệu gốc: 0.8774937769412908
Độ chính xác với dữ liệu gốc: 0.85
Độ chính xác với dữ liệu đã giảm chiều: 0.8541666666666666

---Chia theo tỉ lệ: 0.4---
Độ chính xác chéo với dữ liệu gốc: 0.8789067142008318
Độ chính xác với dữ liệu gốc: 0.84375
Độ chính xác với dữ liệu đã giảm chiều: 0.846875
```

Regularization (L1, L2)

Giúp tránh overfitting bằng cách thêm thành phần phạt vào hàm mất mát:

- L1 (Lasso): chọn lọc đặc trưng (nhiều trọng số = 0).
- L2 (Ridge): giảm độ lớn trọng số, mô hình ổn định.

Kết quả

- Accuracy ~ 0.82–0.85, cross-val ~ 0.88 → ổn định, không overfit.
- Giảm chiều bằng LDA cải thiện nhẹ hiệu suất và ranh giới phân lớp.

So sánh mô hình trước & sau giảm chiều

Mô hình	Giảm chiều	Accuracy	Precision (ASD=1)	Recall (ASD=1)	F1-score (ASD=1)
KNN	Không	0.8	0.53	0.91	0.67
KNN	LDA	0.85	0.65	0.78	0.71
Logistic Regression	Không	0.85	0.63	0.81	0.71
Logistic Regression	LDA	0.85	0.64	0.8	0.71

Nhận xét

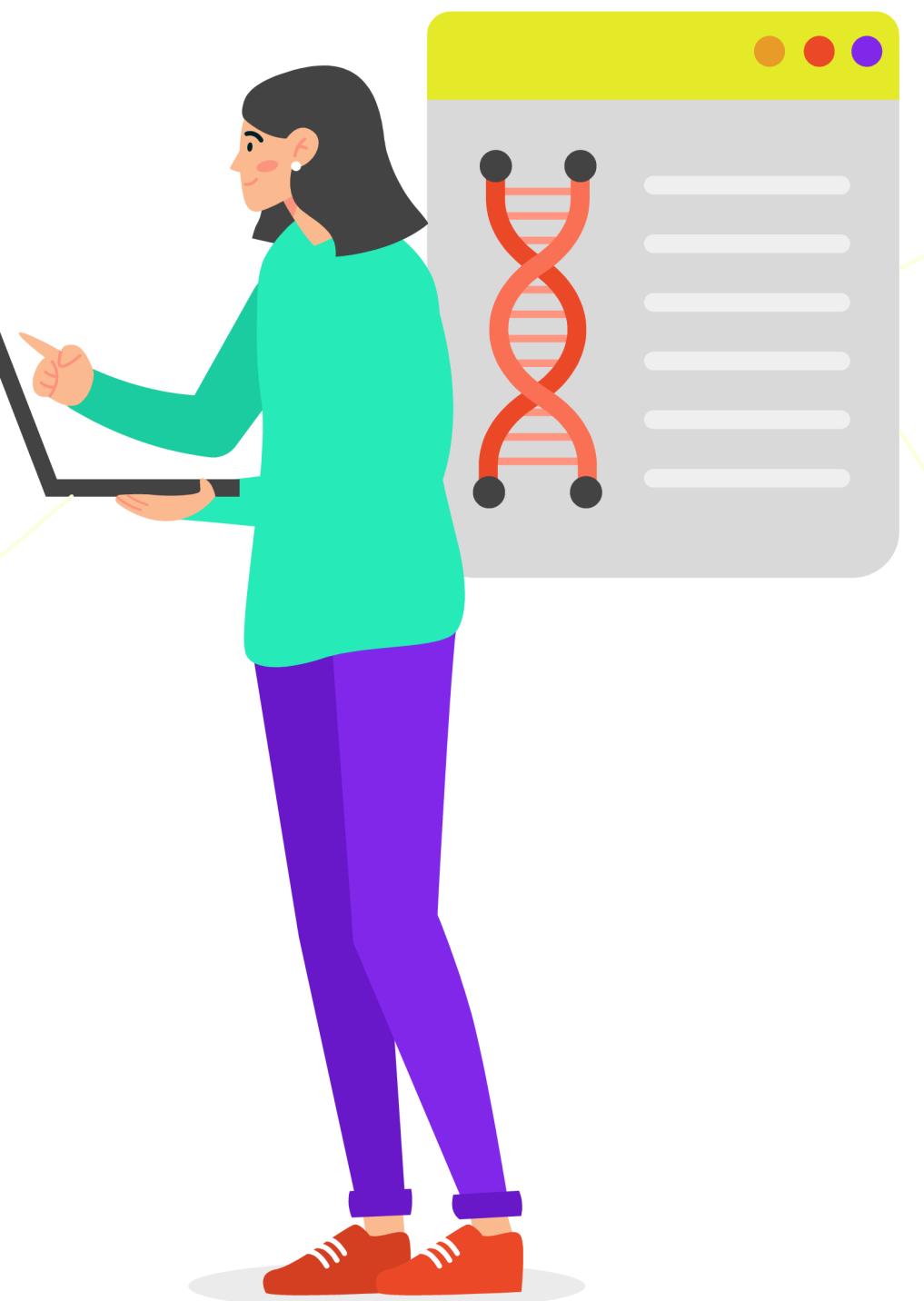
KNN (chưa giảm chiều): Recall cao (0.91) → nhận diện tốt ASD, nhưng precision thấp (0.53) → dễ dự đoán nhầm (nhiều false positive).

KNN sau LDA: Hiệu quả cân bằng hơn, accuracy tăng 0.80 → 0.85, ranh giới phân lớp rõ hơn.

Logistic Regression:
Accuracy ổn định 0.85 trước & sau LDA, precision-recall hài hòa → phù hợp với dữ liệu tuyển tính.

So sánh chung:

- KNN: phù hợp giai đoạn sàng lọc ban đầu (ưu tiên phát hiện đủ ca ASD).
- Logistic Regression: phù hợp đánh giá xác nhận, ổn định & dễ diễn giải.



Kết luận

01

Logistic Regression (Ôn định & dễ diễn giải)

Hoạt động bền vững, accuracy ~0.85; phù hợp bài toán tuyến tính và dễ hiểu trong lĩnh vực y tế.

02

KNN (Nhạy với ca mắc ASD)

Recall cao, phát hiện tốt nhóm ASD nhưng dễ nhầm lớp → thích hợp cho sàng lọc ban đầu.

03

Giảm chiều (LDA)

Cải thiện khả năng tách lớp và tăng hiệu suất mô hình, đặc biệt với KNN.

04

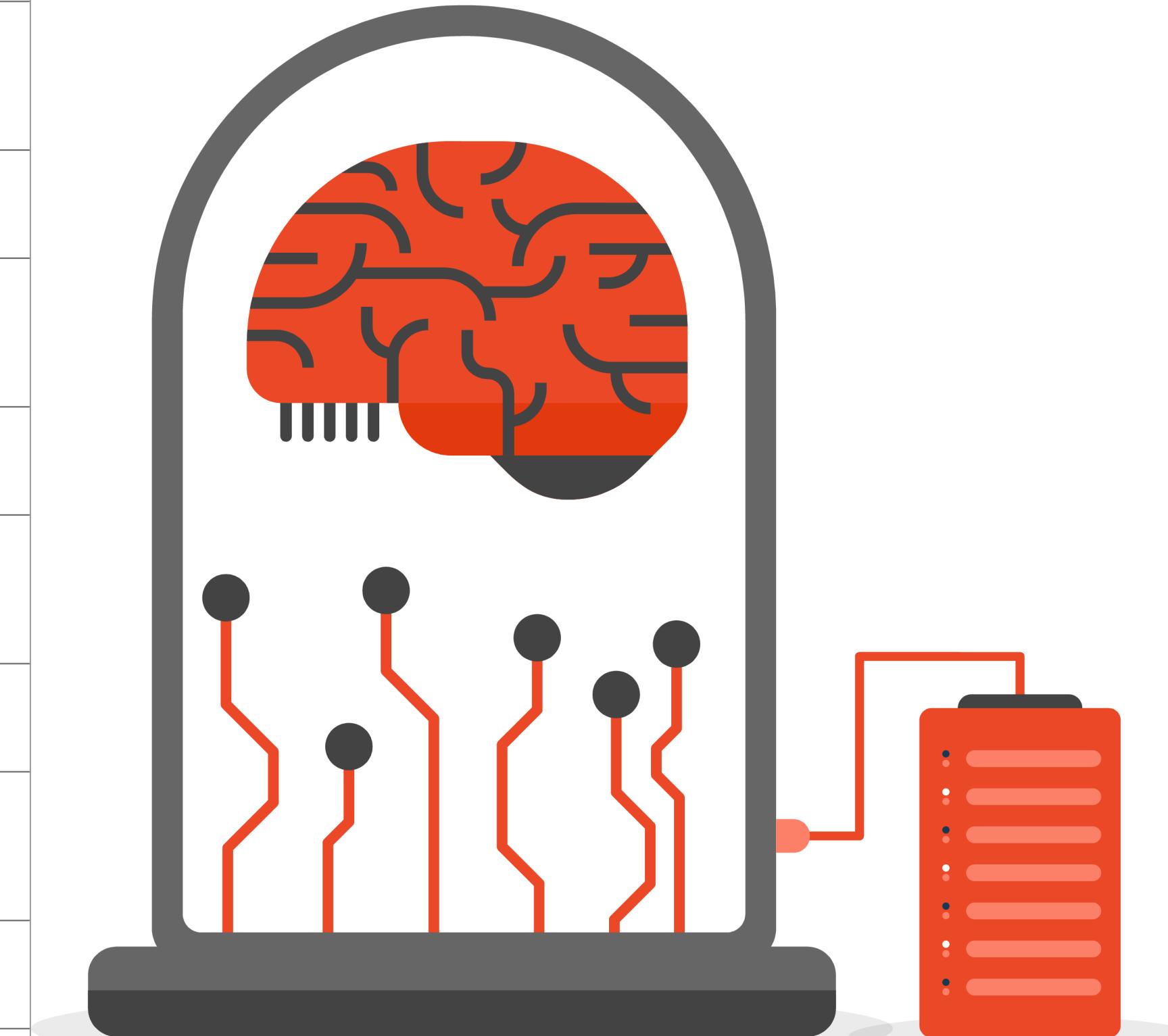
Hạn chế nghiên cứu

Dữ liệu tự khai báo, mất cân bằng lớp, thiếu yếu tố sinh học – xã hội; mô hình mới dừng ở đặc trưng tuyến tính.

05

Ý nghĩa & ứng dụng

Hỗ trợ phát hiện sớm ASD cho phụ huynh, giáo viên, bác sĩ; giúp can thiệp kịp thời, tiết kiệm chi phí chẩn đoán.



Hướng phát triển

Mô hình nâng cao

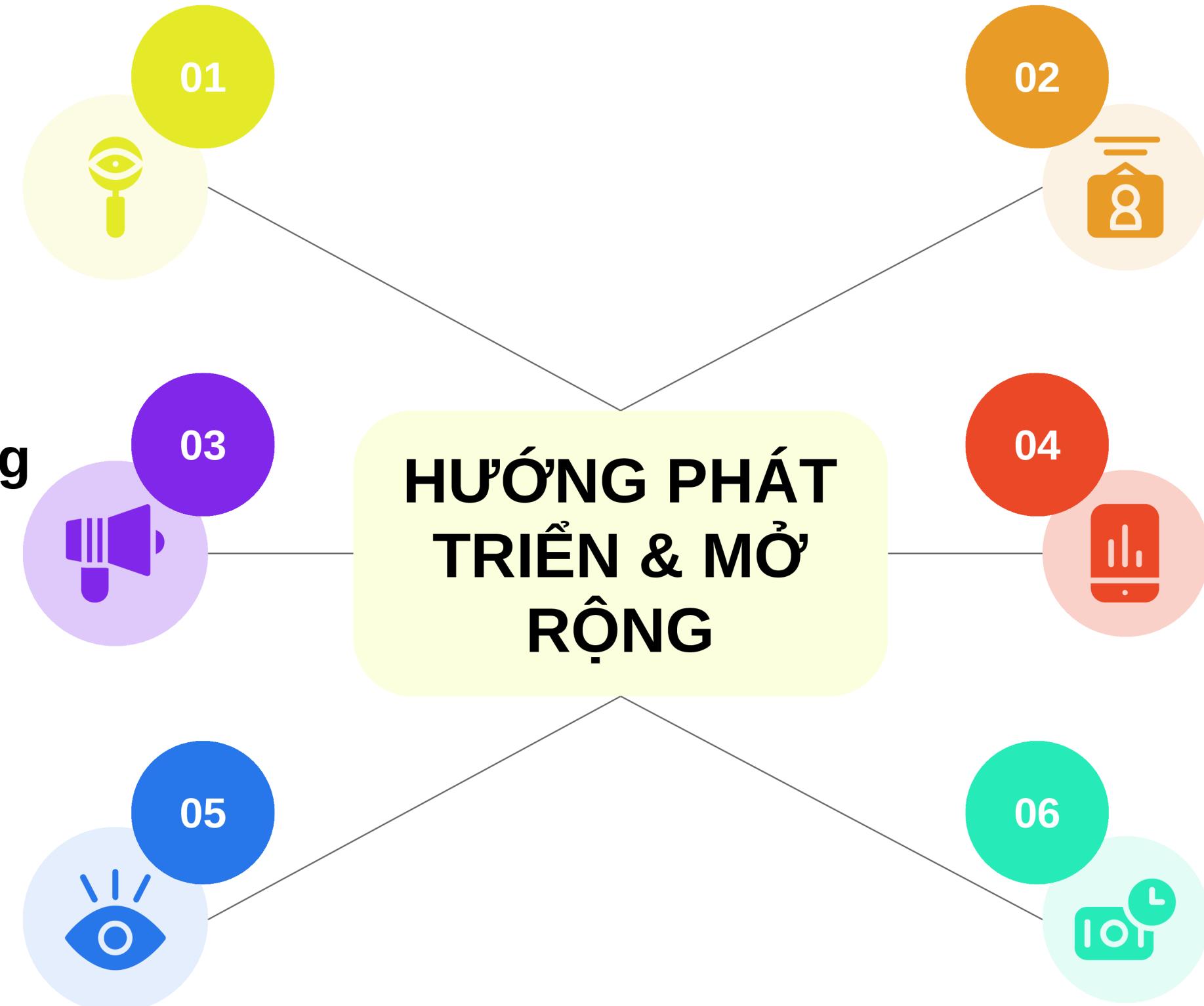
Thử nghiệm SVM, Random Forest, XGBoost, và Neural Network.

Lựa chọn đặc trưng

Xác định các hành vi quan trọng nhất thay vì chỉ giảm chiều.

Mở rộng dữ liệu

Tăng kích thước và đa dạng hóa dataset theo vùng, tuổi, văn hóa.



Dữ liệu hành vi thực tế

Kết hợp video, cử chỉ, giọng nói, ánh mắt để mô phỏng đánh giá chuyên gia.

Ứng dụng Web/Mobile

Phát triển công cụ sàng lọc tự động, giúp người dùng tự đánh giá.

Hướng nghiên cứu tiếp theo

Nghiên cứu yếu tố môi trường, di truyền, và xã hội để nâng cao độ chính xác.

Thank you

