Trường Đại học Khoa học tự nhiên – Khoa Công nghệ thông tin.

Đồ án thực hành số 02

Trực quan hóa dữ liệu – Data Visualization.

Nhóm 16
Tháng 11, 2024.

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN THỰC HÀNH SỐ 02

Bộ môn: Trực quan hóa dữ liệu.

Tên đề tài: *"Linear Discriminant Analysis."*

STT nhóm: 16.

Thành viên thực hiện:

22120412 – Nguyễn Anh Tường.

Thông tin chung:

1. **Bộ môn:** Trực quan hóa dữ liệu.
2. **Giảng viên lý thuyết:** Thầy Bùi Tiến Lên.
3. **Giảng viên thực hành:** Thầy Lê Nhựt Nam.
4. **Mã lớp:** 22_21.
5. **STT nhóm:** 16.
6. **Danh sách thành viên:**
    a. 22120378 – Nguyễn Ngọc Khánh Trân.
    b. 22120384 – Nguyễn Đình Trí.
    c. 22120387 – Trần Đức Trí.
    d. 22120412 – Nguyễn Anh Tường.
7. **Thành viên thực hiện:**
    a. 22120412 – Nguyễn Anh Tường.

MỤC LỤC

ĐỒ ÁN THỰC HÀNH SỐ 02 2

Thông tin chung: 3

Section 0: *Đánh giá kết quả.* 6

I. Bảng công việc và tiến độ hoàn thành: 7

II. Đánh giá tiến độ: 7

III. Hướng dẫn sử dụng chương trình: 7

Section 1: *Tìm hiểu về LDA.* 8

I. Động lực phát triển và định nghĩa LDA: 9

II. Đặc điểm: 9

III. Vì sao nên sử dụng phương pháp LDA? 10

IV. Những hạn chế của phương pháp LDA? 10

V. Mô tả phương pháp LDA: 10

Section 2: *Thuật toán LDA và cài đặt mẫu.* 11

I. Thuật toán LDA: 12

II. Cài đặt thuật toán bằng python và thư viện Numpy: 13

1. Hàm fit() dùng để xây dựng một không gian vector mới với chiều thấp hơn. 13

2. Hàm tranform() dùng để chiếu dữ liệu lên chiều không gian đại diện mới vừa được tìm thấy. 13

3. Hàm fit_and_tranform() dùng để thực hiện LDA. 13

III. Áp dụng cho tập dữ liệu số random: 14

IV. Áp dụng cho tập dữ liệu– Wine.csv: 15

1. Mô tả tập dữ liệu: 15

2. Kết quả áp dụng LDA với n_components = 2: 15

Section 3: *Bản chất toán học của LDA.* 16

I. Bài toán: 17

II. Giải quyết bài toán: 18

Section 4: *So sánh LDA và PCA.* 20

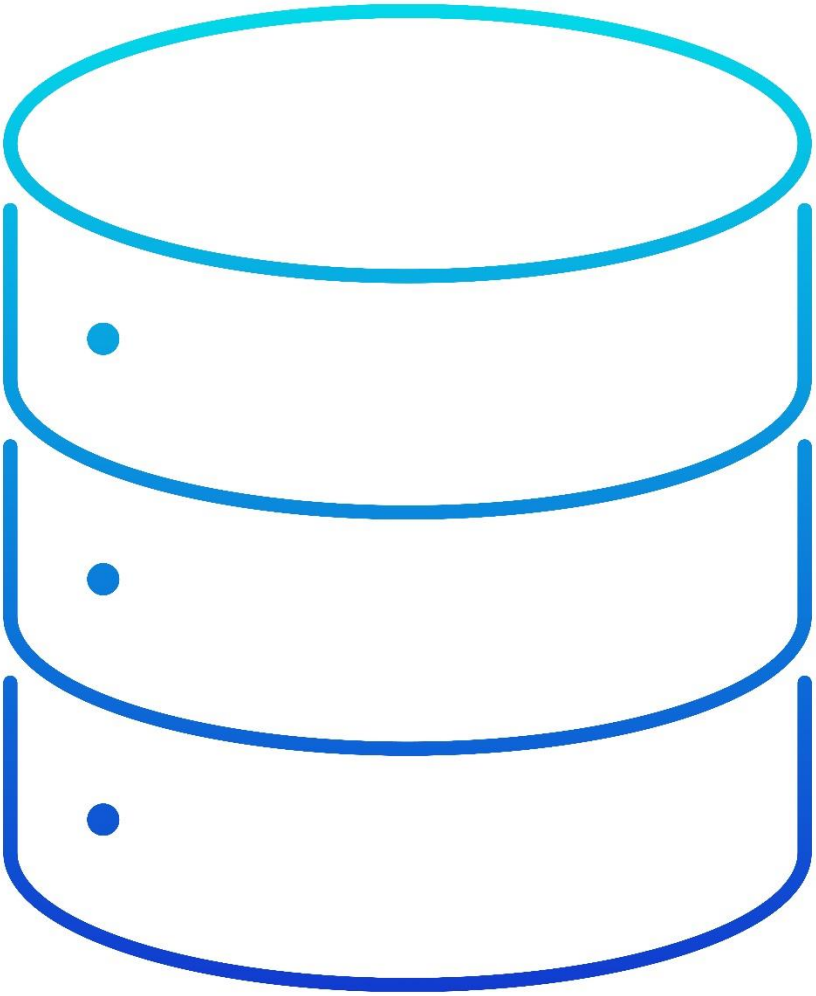
1. Mục tiêu: 21

2. Cách thức hoạt động: 21

3. Yêu cầu về nhãn (labels): 21

4. Số lượng thành phần tối đa: 22
5. Ứng dụng: 22
6. Ví dụ về cách sử dụng PCA và LDA: 22

Section 0: *Đánh giá kết quả.*



I. Bảng công việc và tiến độ hoàn thành:

Số thứ tự	Điểm	Tên công việc	Tiến độ
1	45%	Studying PCA	100%
2	45%	Implementation PCA	100%
3	10%	Overall comprehension of the submitted source code.	100%
4	10%	Bonus points	100%

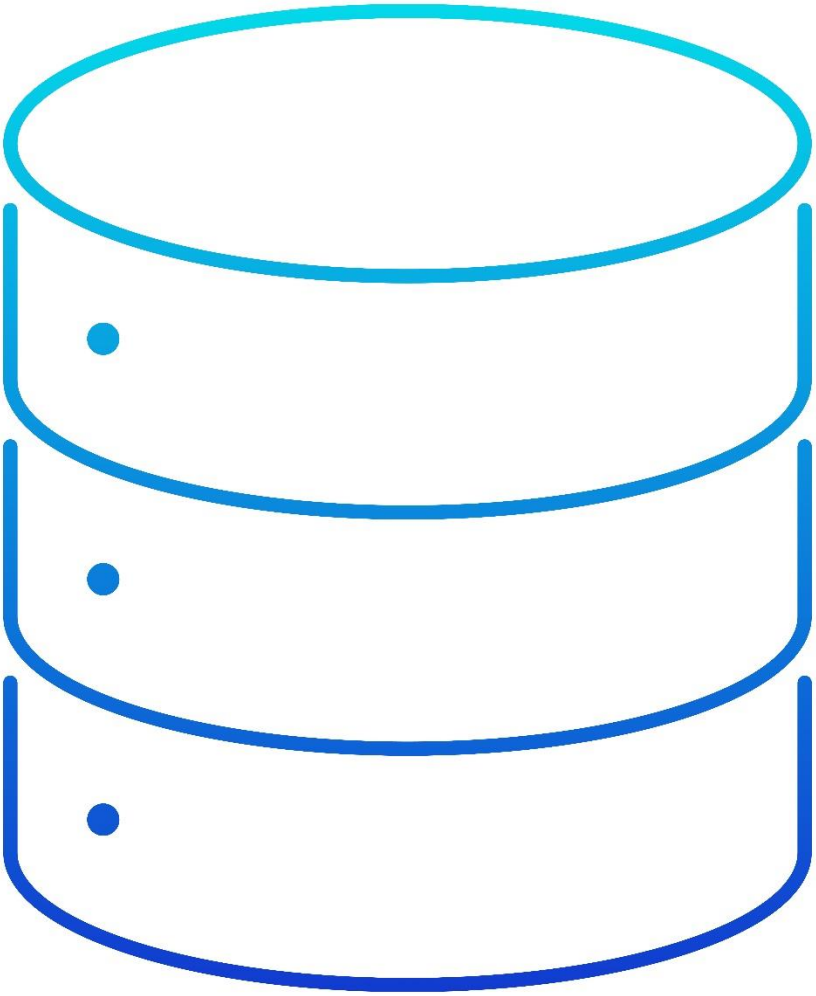
II. Đánh giá tiến độ:

110%

III. Hướng dẫn sử dụng chương trình:

- Thực thi file Main.py

Section 1: *Tìm hiểu về LDA.*



I. Động lực phát triển và định nghĩa LDA:

LDA hay **NDA** – Normal Discriminant Analysis – Discriminant Function Analysis đều là một dạng của Fisher's Linear Discriminant – Phương pháp tìm một tổ hợp tuyến tính các đặc trưng hoặc tách biệt hai hay nhiều lớp của đối tượng hoặc sự kiện nào đó.

Định nghĩa về LDA: là một kỹ thuật thống kê để phân loại dữ liệu thành các nhóm. Nó xác định các mẫu trong các tính năng để phân biệt giữa các lớp khác nhau.

Diễn giải: nhằm mục đích tìm một đường thẳng hoặc mặt phẳng phân tách tốt nhất các nhóm này trong khi giảm thiểu sự chồng chéo trong mỗi lớp. Bằng cách tối đa hóa sự phân tách giữa các lớp, nó cho phép phân loại chính xác các điểm dữ liệu mới. Nói một cách đơn giản hơn, LDA giúp hiểu dữ liệu bằng cách tìm ra cách hiệu quả nhất để phân tách các danh mục khác nhau.

II. Đặc điểm:

Trong các bài toán phân loại và giảm chiều dữ liệu, đặc biệt là đối với các bài toán học có giám sát thì việc tìm ra một không gian chiếu tối ưu là việc rất quan trọng. Sở dĩ như vậy là vì con người chỉ có thể biểu diễn dữ liệu dưới dạng một số chiều hạn chế nhất định – thường là 2 – 3 chiều.

Chiều không gian tối ưu này sẽ đại diện cho cả thẩy dữ liệu của chúng ta, giúp chúng ta phân biệt các lớp với nhau, tách biệt các lớp này và tối ưu hóa khoảng cách giữa chúng để giảm thiểu tối đa sự nhầm lẫn.

LDA còn giúp chúng ta lượt bỏ những chiều dữ liệu không thật sự quan trọng và giữ lại những thông tin cần thiết cho việc phân loại và lọc bỏ nhiễu.

LDA là một phương pháp được xây dựng trên các khái niệm toán học về phân phối Gaussian và phương sai.

III. Vì sao nên sử dụng phương pháp LDA?

Hồi quy logistic là một trong những mô hình phân loại tuyến tính phổ biến nhất, hoạt động tốt đối với phân loại nhị phân nhưng lại không hiệu quả trong trường hợp có nhiều vấn đề phân loại với các lớp được phân tách tốt. Trong khi LDA xử lý những vấn đề này khá hiệu quả.

Phân tích phân biệt tuyến tính (LDA) cũng làm giảm số lượng tính năng trong quá trình xử lý dữ liệu trước, tương tự như Phân tích thành phần chính (PCA), do đó làm giảm đáng kể chi phí tính toán.

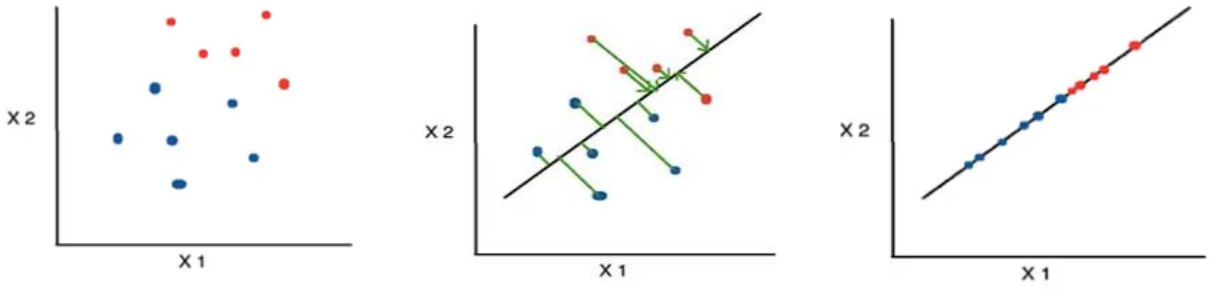
LDA cũng được sử dụng trong các thuật toán phát hiện khuôn mặt. Trong Fisherfaces, LDA được sử dụng để trích xuất dữ liệu hữu ích từ các khuôn mặt khác nhau. Kết hợp với eigenfaces, nó tạo ra kết quả hiệu quả.

IV. Những hạn chế của phương pháp LDA?

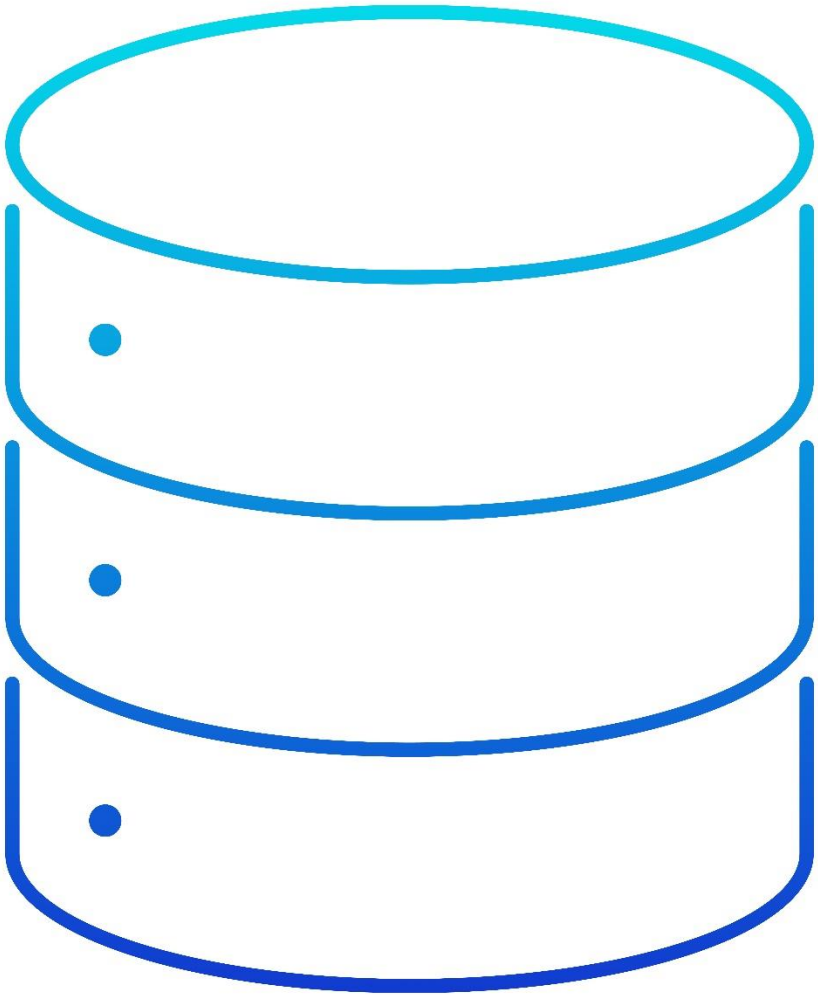
Ranh giới quyết định tuyến tính (chiều dữ liệu đại diện) có thể không phân tách hiệu quả các lớp không thể phân tách tuyến tính. Cần có ranh giới linh hoạt hơn.

Trong trường hợp số lượng quan sát vượt quá số lượng features, LDA có thể không hoạt động như mong muốn. Đây được gọi là vấn đề kích thước mẫu nhỏ (SSS). Cần phải có sự chuẩn hóa thì mới hoạt động hiệu quả hơn.

V. Mô tả phương pháp LDA:



Section 2: *Thuật toán LDA và cài đặt mẫu.*



I. Thuật toán LDA:

Nền tảng: LDA được phát triển dựa trên các khái niệm toán học liên quan đến phân phối Gaussian, ma trận hiệp phương sai (covarience), …

Mục tiêu: Nhằm tối ưu hóa tỉ lệ Reyleigh (Reyleigh Quotient), là tỉ lệ giữa phương sai giữa các lớp (giữa các trung bình lớp) và phương sai trong lớp.

Công thức cho tỉ lệ Reyleigh:

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

Trong đó:

- w là vector chiếu mà chúng ta cần tìm.
- S_B là **ma trận phương sai giữa các lớp** (between-class scatter matrix).
- S_W là **ma trận phương sai trong lớp** (within-class scatter matrix).

Trình bày thuật toán:

Đầu tiên: Tính toán khả năng tách biệt giữa các lớp, tức là khoảng cách giữa các giá trị trung bình của các lớp khác nhau. Hay còn được gọi là phương sai giữa các lớp.

$$S_B = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T$$

Trong đó:

- *c* là số lượng các lớp.
- Ni là số lượng mẫu trong lớp i.
- μ là vector trung bình của lớp i.

Tiếp theo: Tính khoảng cách giữa giá trị trung bình và mẫu của mỗi lớp. Nó cũng được gọi là phương sai trong lớp.

$$S_W = \sum_{i=1}^{c} \sum_{x \in X_i} (x - \mu_i)(x - \mu_i)^T$$

Trong đó:

- *c* là số lượng các lớp.
- *Xi* là tập hợp các mẫu thuộc lớp *i*.
- μi là vector trung bình của lớp *i*.

Cuối cùng: Xây dựng một không gian mới với chiều thấp hơn để đối đa hóa phương sai giữa các lớp và tối thiểu hóa phương sai trong lớp. Quá trình này còn được gọi là tìm phép chiếu không gian chiều thấp hơn, còn được gọi là tiêu chuẩn Fisher.

$$P_{lda} = \arg\max_{P} \frac{|P^T S_b P|}{|P^T S_w P|}$$

Trong đó:

$P^T S_b P$ và $P^T S_w P$: Tương ứng là ma trận được chiếu theo P cho phương sai giữa các lớp và trong lớp.

II. Cài đặt thuật toán bằng python và thư viện Numpy:

1. Hàm fit() dùng để xây dựng một không gian vector mới với chiều thấp hơn.

a. **Đầu vào:** Ma trận dữ liệu X, Vector nhãn dán y.

b. **Kiểu trả về:** Vector riêng đại diện cho chiều không gian đó.

c. **Cài đặt:**

Bước 1: lấy ra các lớp có trong tập nhãn dán.

Bước 2: tính trung bình của cả tập dữ liệu X.

Bước 3: Tính S_W, S_B bằng các công thức ở mục I. thông qua vòng lặp.

Bước 4: Tính trị riêng và vector riêng cho S_W^(-1) * S_B.

Bước 5: Sắp xếp các trị riêng và vector riêng này theo thứ tự giảm dần để lấy ra đối lượng lớn nhất (maximum).

Bước 6: Chọn ra *n_components* thành phần trong vector riêng lớn nhất.

2. Hàm tranform() dùng để chiếu dữ liệu lên chiều không gian đại diện mới vừa được tìm thấy.

3. Hàm fit_and_tranform() dùng để thực hiện LDA.

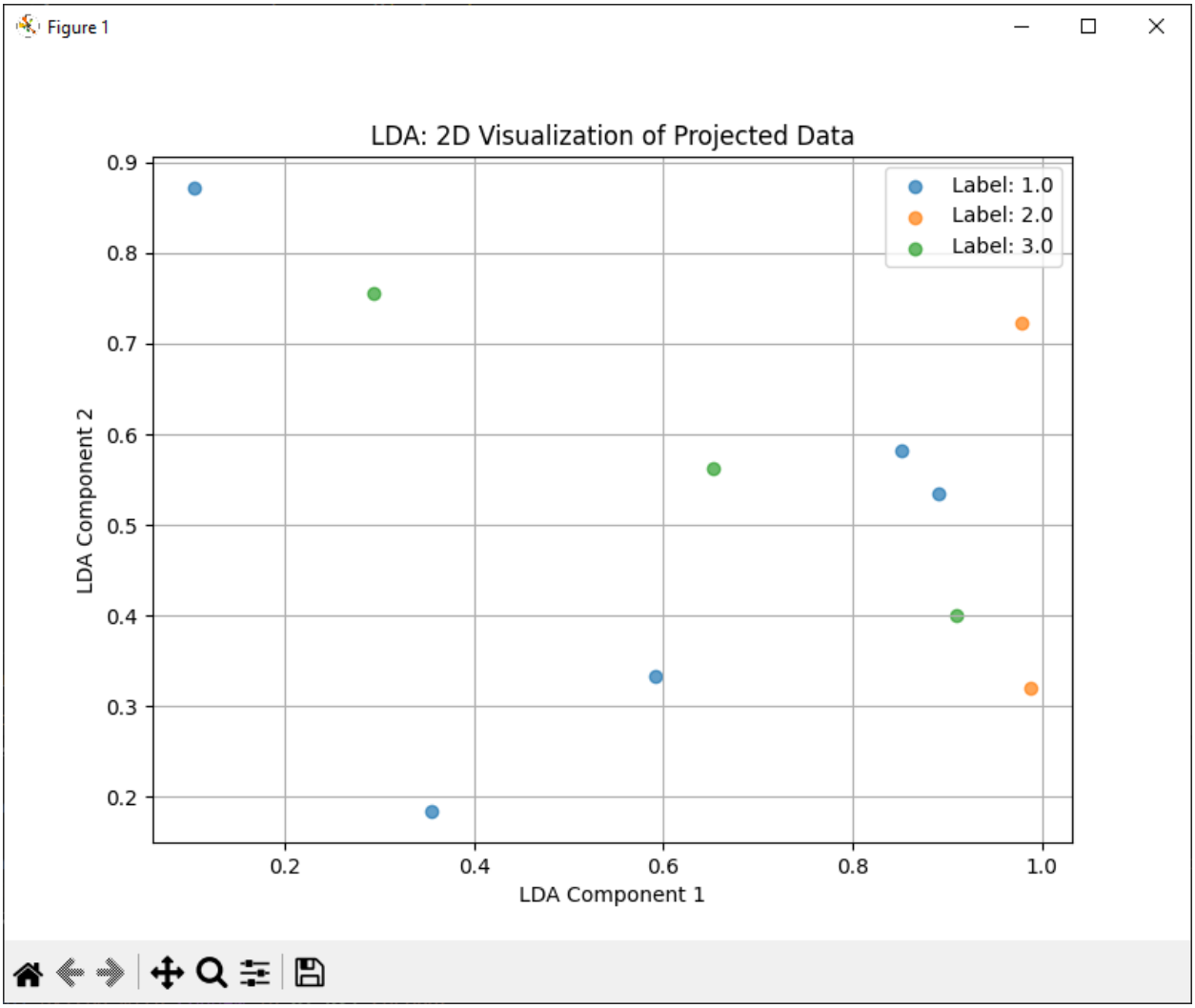
III. Áp dụng cho tập dữ liệu số random:

Số cột: 10.

Số hàng: 50.

Giá trị: Được sinh ngẫu nhiên từ hàm random.

Số label: 3 – có giá trị lần lượt là 1.0 – 2.0 – 3.0



IV. Áp dụng cho tập dữ liệu– Wine.csv:

1. Mô tả tập dữ liệu:

a. **Số cột:** 14 cột.

i. 13 cột thuộc tính mô tả mùi vị, thành phần,… của rượu.

ii. 1 cột phân khúc khách hàng.

b. **Số hàng:** 178 hàng.

2. Kết quả áp dụng LDA với n_components = 2:

Lý do chọn n_components = 2: Vì trong tập Wine.csv có tổng cộng 3 phân khúc khách hàng lần lượt là 1 – 2 – 3.

Do đó khi ta sử dụng công thức n_components = C – 1 thì kết quả sẽ ra 2.

Biểu đồ thể hiện kết quả:

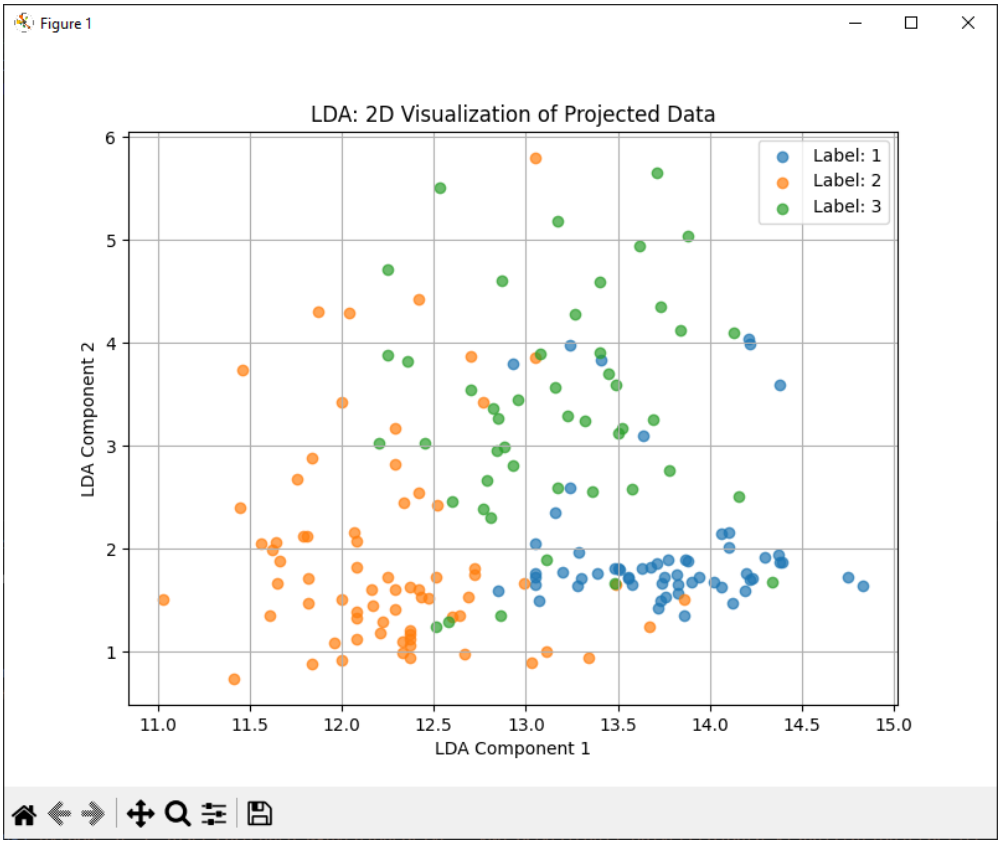
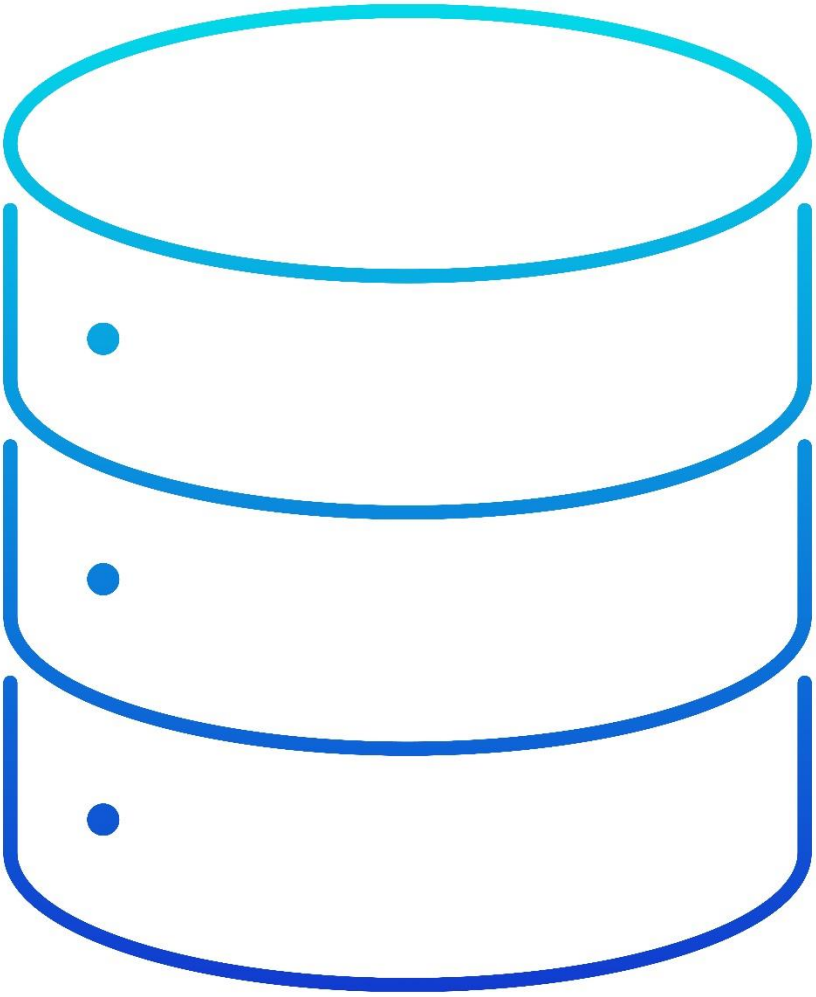
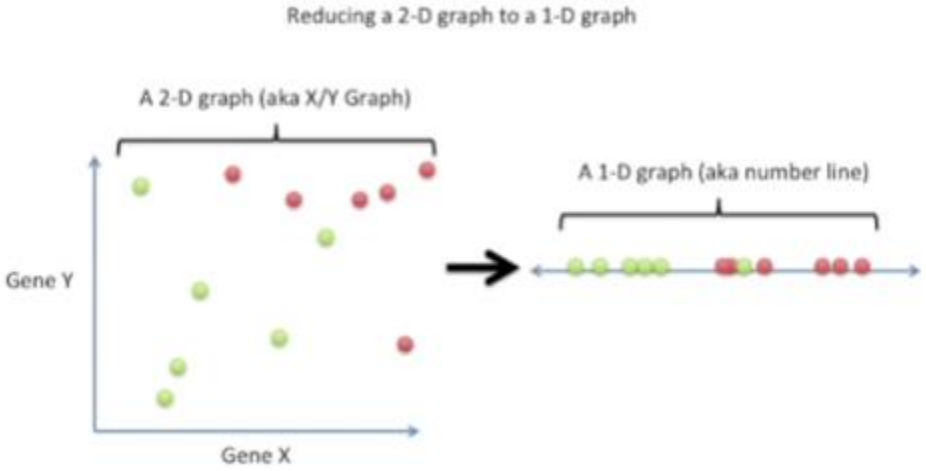


Figure 1. Kết quả thực hiện LDA với tập Wine.csv

Section 3: *Bản chất toán học của LDA.*



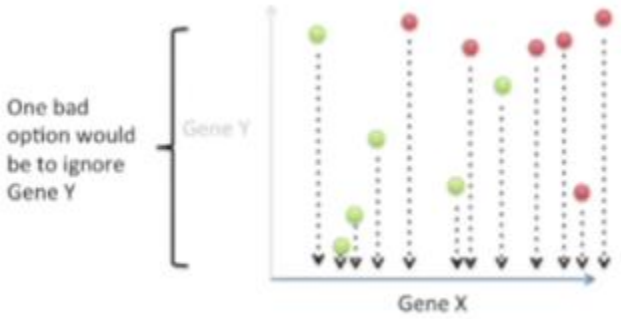
I. Bài toán:



Ta có một biểu diễn dữ liệu 2 chiều, ta đang muốn chuyển nó về một chiều bằng cách chiếu các điểm dữ liệu này lên một chiều đại diện nào đó.

Trường hợp 1:

Bỏ qua trục Y và chiếu các điểm dữ liệu xuống trục X.



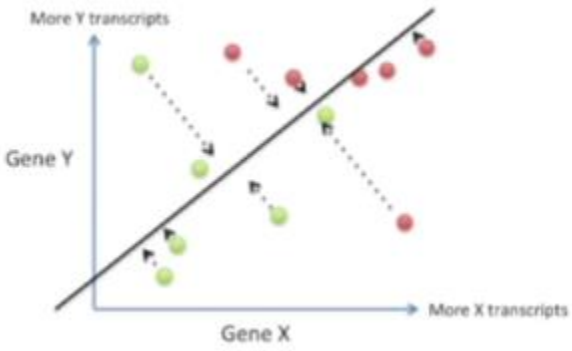
Nhận xét: Trường hợp này không tốt, vì đã bỏ qua những thông tin chính và quan trọng từ Y.

Trường hợp 2:

Tương tự nhưng ta chiếu các điểm dữ liệu lên trục Y và bỏ qua trục X.

Nhận xét chung:

Cả hai trường hợp trên đều tệ, vì thế ta cần tìm một chiều dữ liệu chưa biết trước nào đó có thể đại diện rõ nét cho cả 2 trục X và Y này.



Khi này, trục nào đen sẽ là trục ta cần tìm để biểu diễn điểm chiếu của các điểm dữ liệu.

II. Giải quyết bài toán:

Để tìm được một trục *'z nào đó'* đại diện thì ta cần làm các bước như em đã trình bày ở trên phần thuật toán LDA.

Đầu tiên: Tính toán khả năng tách biệt giữa các lớp, tức là khoảng cách giữa các giá trị trung bình của các lớp khác nhau. Hay còn được gọi là phương sai giữa các lớp.

$$S_B = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T$$

Trong đó:

- c là số lượng các lớp.
- Ni là số lượng mẫu trong lớp i.
- μ là vector trung bình của lớp i.

Tiếp theo: Tính khoảng cách giữa giá trị trung bình và mẫu của mỗi lớp. Nó cũng được gọi là phương sai trong lớp.

$$S_W = \sum_{i=1}^{c} \sum_{x \in X_i} (x - \mu_i)(x - \mu_i)^T$$

Trong đó:

- c là số lượng các lớp.

- Xi là tập hợp các mẫu thuộc lớp i.
- μi là vector trung bình của lớp i.

Cuối cùng: Xây dựng một không gian mới với chiều thấp hơn để đối đa hóa phương sai giữa các lớp và tối thiểu hóa phương sai trong lớp. Quá trình này còn được gọi là tìm phép chiếu không gian chiều thấp hơn, còn được gọi là tiêu chuẩn Fisher.

$$P_{lda} = \arg\max_{P} \frac{|P^T S_b P|}{|P^T S_w P|}$$

Trong đó:

$P^T S_b P$ và $P^T S_{\mathrm{w}} P$: Tương ứng là ma trận được chiếu theo P cho phương sai giữa các lớp và trong lớp.

Và để tìm được phép chiếu này thì cần thực hiện tiếp các bước sau:

Bước 1: Tính các trị riêng và vector riêng

Sau khi có S_W và S_B, chúng ta cần tối đa hóa tỷ lệ phân tán giữa các lớp và phân tán trong lớp. Vấn đề này được giải bằng cách giải bài toán giá trị riêng (eigenvalue problem) của ma trận kết hợp:

$$S_W^{-1} S_B$$

Lúc này, chúng ta tìm các trị riêng và vector riêng của $S_W^{-1}\, S_B$. Vector riêng tương ứng với các trị riêng lớn nhất sẽ tạo thành các phương hướng mà khi chiếu dữ liệu lên, chúng ta đạt được sự phân tách lớp tối ưu.

Bước 2: Chọn các thành phần LDA (số lượng thành phần)

Số lượng thành phần (dimensions) tối đa mà LDA có thể tạo ra là k-1, với k là số lượng lớp.

Do đó, chúng ta chọn k-1 vector riêng tương ứng với k - 1 trị riêng lớn nhất.

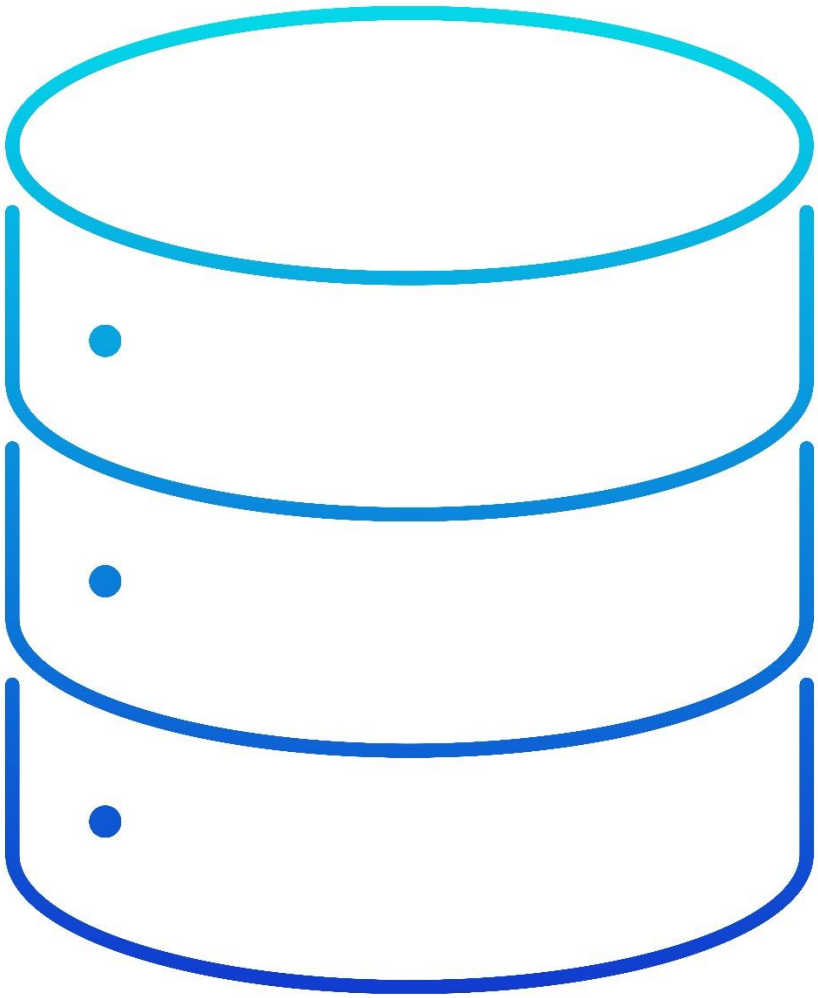
Bước 3: Chiếu dữ liệu lên không gian mới

Sau khi chọn các vector riêng, ta chiếu dữ liệu XXX lên không gian mới với số chiều là k-1:

$$X_{\mathrm{LDA}} = X \times W$$

trong đó W là ma trận chứa các vector riêng đã chọn. Không gian này giữ lại được sự phân tách tối ưu giữa các lớp.

Section 4: *So sánh LDA và PCA.*



1. Mục tiêu:

- **PCA**: PCA tập trung vào **biến thiên tổng thể** của dữ liệu mà không cần biết nhãn của dữ liệu. Mục tiêu chính của PCA là tìm ra các trục chính để **giảm chiều dữ liệu** bằng cách giữ lại các thành phần có phương sai lớn nhất, nhằm bảo toàn lượng thông tin nhiều nhất có thể.
- **LDA**: LDA hướng tới việc **phân tách các lớp** trong dữ liệu và do đó cần biết nhãn của từng mẫu (supervised). Mục tiêu chính của LDA là tìm các trục tối ưu để **tối đa hóa sự phân tách giữa các lớp**, giúp phân loại các lớp hiệu quả hơn trong không gian mới.

2. Cách thức hoạt động:

- **PCA**:
    - Tính phương sai của từng đặc trưng và tính toán ma trận hiệp phương sai của dữ liệu.
    - Tìm các trị riêng và vector riêng của ma trận hiệp phương sai.
    - Chọn các vector riêng tương ứng với các trị riêng lớn nhất để tạo thành các thành phần chính, giữ lại phần lớn phương sai.
    - Chiếu dữ liệu lên các trục chính này để giảm chiều.
- **LDA**:
    - Tính trung bình của từng lớp và trung bình tổng thể của dữ liệu.
    - Tính ma trận phân tán trong lớp (Within-Class Scatter Matrix) và ma trận phân tán giữa các lớp (Between-Class Scatter Matrix).
    - Giải bài toán trị riêng của ma trận kết hợp $S_W^{-1}S_B$ (ma trận phân tán trong lớp nghịch đảo nhân với ma trận phân tán giữa các lớp).
    - Chọn các vector riêng với trị riêng lớn nhất để tạo các thành phần LDA và chiếu dữ liệu lên các trục này để tối đa hóa phân tách giữa các lớp.

3. Yêu cầu về nhãn (labels):

- **PCA**: Không yêu cầu nhãn của dữ liệu vì PCA là phương pháp **không giám sát (unsupervised)**. PCA chỉ phân tích cấu trúc nội tại của dữ liệu mà không quan tâm đến thông tin phân loại.
- **LDA**: Yêu cầu nhãn của dữ liệu vì LDA là phương pháp **có giám sát (supervised)**. Thông tin nhãn giúp LDA tính toán được các ma trận phân tán để tối ưu hóa phân tách lớp.

4. Số lượng thành phần tối đa:

- **PCA**: Số lượng thành phần tối đa mà PCA có thể tạo ra phụ thuộc vào số lượng đặc trưng ban đầu của dữ liệu (số chiều ban đầu). Nếu dữ liệu có nnn đặc trưng, PCA có thể tạo tối đa nnn thành phần chính.
- **LDA**: Số lượng thành phần tối đa của LDA bị giới hạn bởi $C-1$, với C là số lượng lớp trong dữ liệu. Ví dụ, nếu có 3 lớp, LDA chỉ có thể tạo ra tối đa 2 thành phần, vì chỉ có 2 trục là tối ưu cho phân tách giữa các lớp.

5. Ứng dụng:

- **PCA**: PCA phù hợp để giảm chiều dữ liệu nhằm tối ưu hóa hiệu suất tính toán và giữ lại lượng thông tin lớn nhất mà không phân biệt lớp. Do đó, PCA thường được sử dụng trong **nhiệm vụ phát hiện mẫu, phân cụm và biểu diễn dữ liệu**.
- **LDA**: LDA thường được sử dụng trong các bài toán **phân loại có nhãn**, đặc biệt khi có nhiều lớp và ta muốn tăng khả năng phân biệt của mô hình bằng cách chiếu dữ liệu xuống không gian tối ưu cho phân tách lớp.

6. Ví dụ về cách sử dụng PCA và LDA:

- **PCA**: Nếu ta có một tập dữ liệu với 100 đặc trưng và không có nhãn, ta có thể sử dụng PCA để giảm xuống khoảng 10-20 thành phần chính, giúp tối ưu hóa lưu trữ và xử lý dữ liệu trong các bài toán như **nén dữ liệu, tìm mẫu ẩn trong dữ liệu phi cấu trúc**.
- **LDA**: Nếu ta có dữ liệu về các loại rượu (như trong ví dụ Wine.csv) và muốn phân loại các loại rượu, ta có thể sử dụng LDA để giảm dữ liệu xuống không gian 2 hoặc 3 chiều để tăng khả năng phân biệt giữa các loại rượu trong bài toán phân loại.