TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN
---o0o---

BÁO CÁO BÀI TẬP NHÓM

LAB - 06

Giảng viên hướng dẫn: TS. Đỗ Như Tài
Môn học: Trí tuệ nhân tạo nâng cao
Nhóm thực hiện: 7
Danh sách thành viên:

3122410489 – Lê Huỳnh Trúc Vy
3122410495 – Trần Mỹ Yên
3120410470 – Lê Quốc Thái
3122410174 – Thái Minh Khang

TP. HỒ CHÍ MINH, THÁNG 11 NĂM 2025

MỤC LỤC

PHÂN CÔNG

Họ tên	**Mã số sinh viên**	**Phân công**
Lê Huỳnh Trúc Vy (Nhóm trưởng)	3122410489	Câu 1: Áp dụng (1,2,3), Kết quả
Trần Mỹ Yên	3122410495	Câu 1: Giới thiệu (1,2), Kết luận, Tài liệu tham khảo
Thái Minh Khang	3122410174	Câu 2: Áp dụng (1,2,3), Kết quả
Lê Quốc Thái	3120410470	Câu 2: Giới thiệu (1,2), Kết luận, Tài liệu tham khảo

NHẬN DẠNG HOA IRIS BẰNG THUẬT TOÁN NAIVE BAYES

I. Giới thiệu.

1. Thuật toán

a. Giới thiệu

Trong bối cảnh dữ liệu ngày càng tăng trưởng mạnh mẽ, các thuật toán học máy đóng vai trò quan trọng trong việc khai thác tri thức và hỗ trợ ra quyết định. Một trong những thuật toán đơn giản nhưng hiệu quả nhất trong các bài toán phân loại là Naive Bayes. Nhờ giả định độc lập có điều kiện giữa các thuộc tính, mô hình này có thể huấn luyện rất nhanh và hoạt động tốt ngay cả khi kích thước dữ liệu lớn. Với ưu điểm dễ triển khai, tốc độ xử lý cao và khả năng tổng quát hóa tốt, Naive Bayes được ứng dụng rộng rãi trong phân loại văn bản, lọc thư rác, phân tích cảm xúc và nhiều bài toán khai phá dữ liệu khác.

Trong báo cáo này, thuật toán Naive Bayes được áp dụng để phân loại bộ dữ liệu Iris, một trong những bộ dữ liệu kinh điển trong học máy, nhằm đánh giá hiệu năng và khả năng ứng dụng của mô hình trong thực tế.

b. Thuật toán Naive Bayes

- Naive Bayes:

Naive Bayes là một thuật toán phân loại dựa trên định lý Bayes và giả định rằng các thuộc tính trong dữ liệu là độc lập với nhau khi biết nhãn lớp. Mặc dù giả định này thường không hoàn toàn đúng trong thực tế, Naive Bayes vẫn cho kết quả khá chính xác và ổn định.

Công thức cơ bản của định lý Bayes:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

Trong đó:

- C: Nhãn lớp
- X: Tập các thuộc tính đầu vào
- P(C | X): Xác suất dữ liệu thuộc lớp C
- Mô hình dự đoán lớp có giá trị xác suất cao nhất.

- Một số mô hình Naive Bayes phổ biến:

Dựa trên phân phối thống kê của dữ liệu, có ba mô hình được sử dụng nhiều nhất:

1. Gaussian Naive Bayes

- Áp dụng cho dữ liệu dạng liên tục.

- Giả định rằng các thuộc tính tuân theo phân phối chuẩn (Gaussian).
- Ứng dụng: phân loại hình ảnh, phân loại dữ liệu cảm biến,…

2. Multinomial Naive Bayes

- Dùng cho dữ liệu rời rạc (đếm số lần xuất hiện).
- Áp dụng phổ biến trong xử lý văn bản, TF-IDF, bag-of-words.
- Ứng dụng: phân loại chủ đề văn bản, lọc thư rác.

3. Bernoulli Naive Bayes

- Dùng cho dữ liệu nhị phân (0/1).
- Thường dùng trong văn bản với biểu diễn nhị phân "từ xuất hiện / không xuất hiện".

● Ứng dụng:

- Lọc email spam – không spam
- Phân loại văn bản theo chủ đề
- Phát hiện cảm xúc (sentiment analysis)
- Dự đoán rủi ro tài chính
- Phân loại hình ảnh cơ bản
- Hệ thống gợi ý

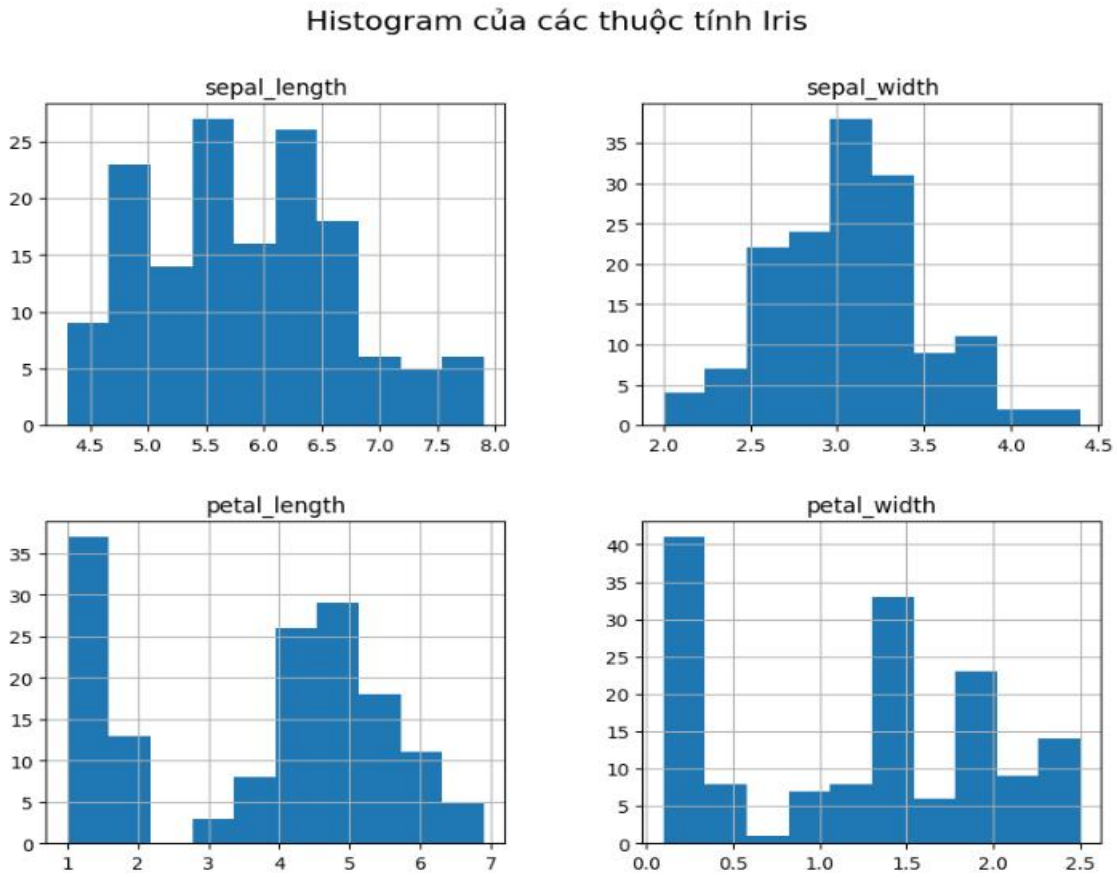
2. Bộ dữ liệu (sử dụng 1 số biểu đồ để thống kê)

a. Mô tả bộ dữ liệu

Bộ dữ liệu Iris (bộ dữ liệu hoa Iris) là một bộ dữ liệu nổi tiếng trong học máy và thống kê, gồm:

- Số mẫu: 150 mẫu
- Số đặc trưng (features): 4 đặc trưng số liên tục
    + Sepal Length (cm) – chiều dài đài hoa
    + Sepal Width (cm) – chiều rộng đài hoa
    + Petal Length (cm) – chiều dài cánh hoa
    + Petal Width (cm) – chiều rộng cánh hoa
- Nhãn (label / target):
    + Iris-setosa
    + Iris-versicolor
    + Iris-virginica
    + Species

b. Thống kê mô tả và trực quan hóa dữ liệu



- Nhận xét về Histogram các thuộc tính Iris
  + Sepal Length (Chiều dài đài hoa)

    *Phân bố gần dạng gần chuẩn (normal-like), với nhiều mẫu tập trung quanh 5.5 – 6.5 cm.

    *Không có outlier quá rõ ràng, phân bố khá đều.
  + Sepal Width (Chiều rộng đài hoa)

    *Phân bố hơi lệch phải (skewed right), nghĩa là phần lớn mẫu có chiều rộng từ 2.5 – 3.5 cm.

    *Một số ít mẫu lớn hơn 4 cm.
  + Petal Length (Chiều dài cánh hoa)

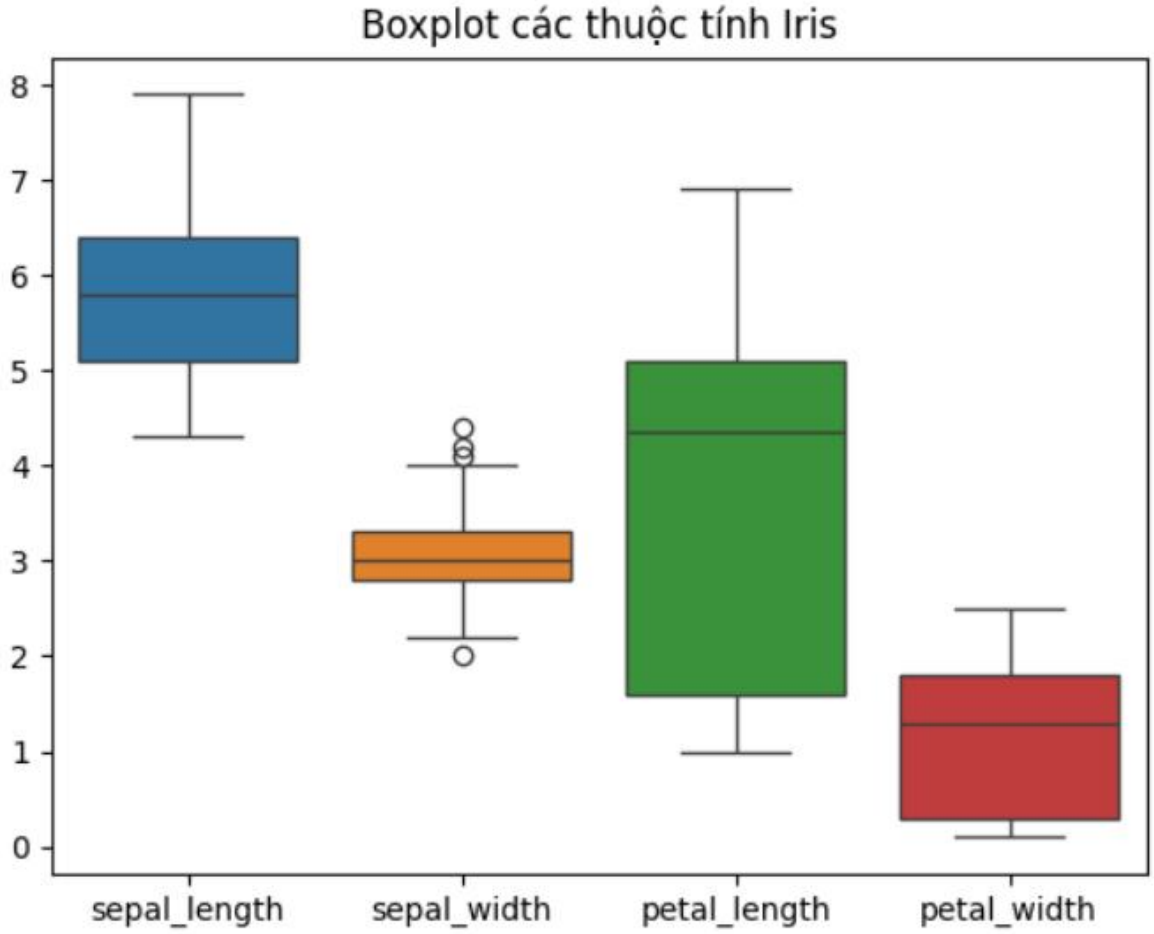
    *Phân bố hai đỉnh (bimodal), phản ánh sự khác biệt rõ ràng giữa các loài.

    *Một nhóm nhỏ khoảng 1–2 cm (Iris-setosa), nhóm khác lớn hơn 4–6 cm (Iris-versicolor và Iris-virginica).
  + Petal Width (Chiều rộng cánh hoa)

*Phân bố cũng có hai đỉnh hoặc nhiều nhóm riêng biệt, cho thấy đặc trưng này giúp phân tách các loài hiệu quả.

*Các giá trị tập trung ở 0–0.5, 1.5–2.0 cm.

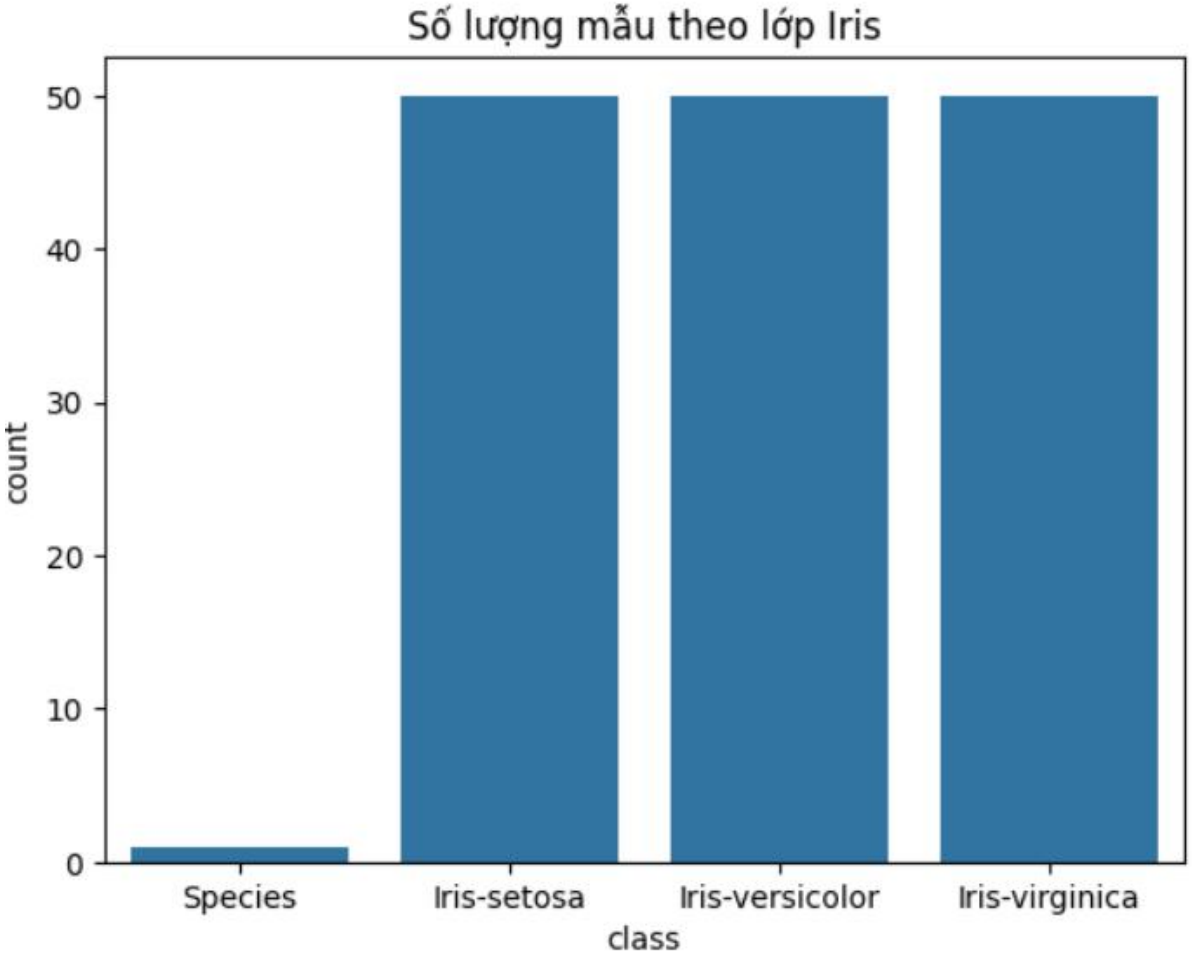
- Kết luận từ histogram
  + Petal Length và Petal Width là hai đặc trưng phân biệt các loài tốt nhất vì có sự phân nhóm rõ ràng.
  + Sepal Length và Sepal Width ít phân biệt hơn, phân bố chồng lên nhau giữa các loài.
  + Histogram giúp trực quan hóa phân bố và sự khác biệt giữa các loài, đồng thời có thể nhận biết outlier hoặc nhóm nhỏ trong dữ liệu.



- Nhận xét Biểu đồ Boxplot Iris
  + sepal_length (Chiều dài đài hoa): Có giá trị trung vị cao nhất (khoảng 5.8) và phân phối khá đối xứng, độ biến động tương đối thấp.

+ sepal_width (Chiều rộng đài hoa): Có độ biến động (IQR) nhỏ nhất, cho thấy dữ liệu rất tập trung. Tuy nhiên, nó là thuộc tính duy nhất có giá trị ngoại lai (outliers) rõ ràng.
+ petal_length (Chiều dài cánh hoa): Có độ phân tán (Range và IQR) lớn nhất trong cả bốn thuộc tính, cho thấy sự khác biệt rất lớn giữa các mẫu. Đây là thuộc tính có khả năng phân loại cao nhất.
+ petal_width (Chiều rộng cánh hoa): Có giá trị trung vị thấp nhất (khoảng 1.3) và độ biến động thấp.

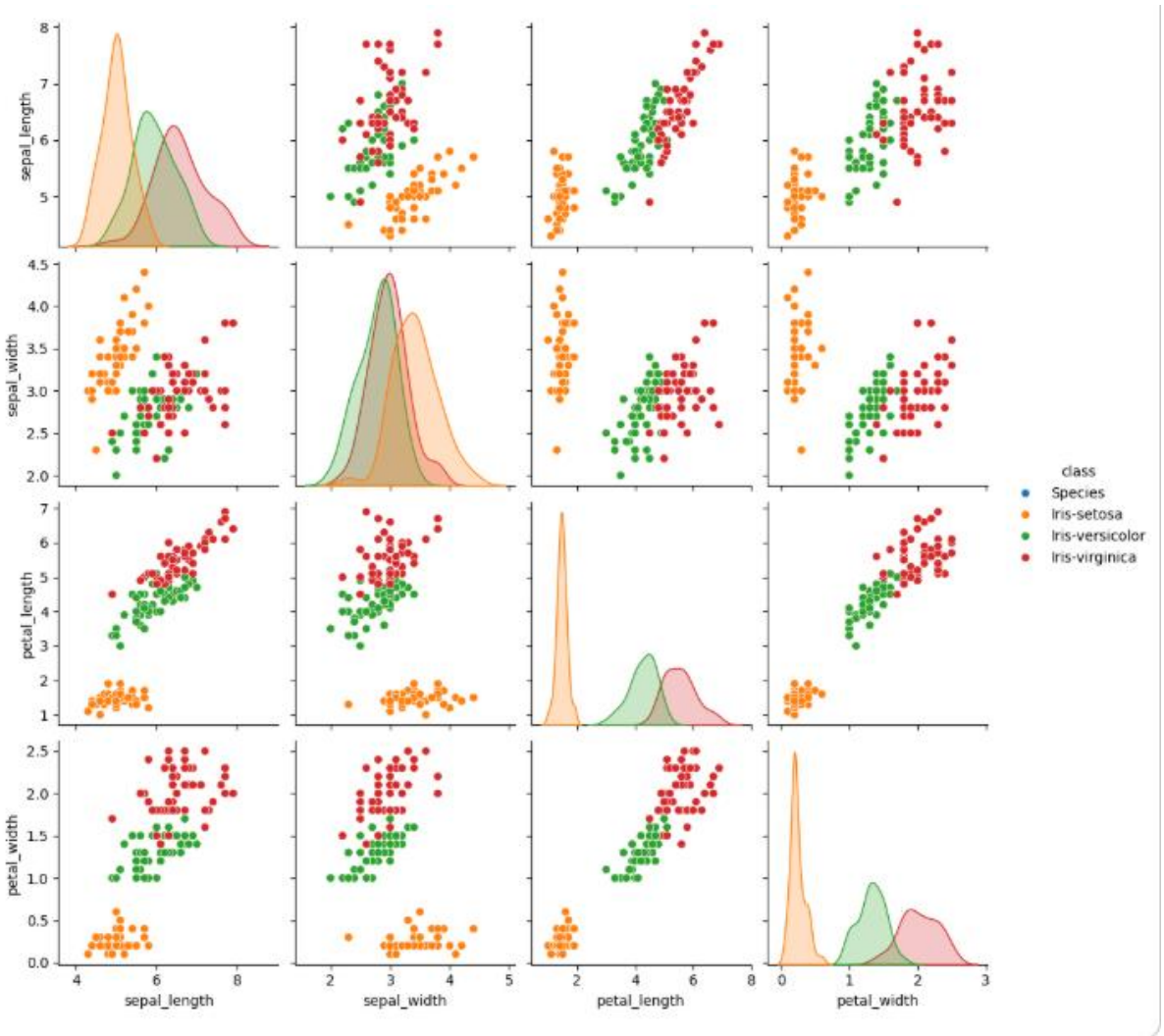
- Kết luận chính: Các thuộc tính của cánh hoa (petal_length, petal_width) có xu hướng có giá trị nhỏ hơn và phân tán rộng hơn (đặc biệt là petal_length) so với các thuộc tính của đài hoa (sepal_length, sepal_width), thể hiện sự khác biệt rõ rệt về hình thái giữa hai phần của bông hoa.



- Số lượng Mẫu theo Lớp Iris

+ Tính cân bằng dữ liệu: Biểu đồ cho thấy tập dữ liệu Iris là rất cân bằng (balanced) đối với ba loài chính: Iris-setosa, Iris-versicolor, và Iris-virginica.Mỗi loài đều có chính xác 50 mẫu (được biểu thị bằng chiều cao cột đạt mức 50 trên trục y).
+ Giá trị không xác định/Khác: Có một cột phụ nằm bên trái với nhãn "Species" (hoặc một nhãn khác tương tự) có giá trị rất nhỏ (chỉ khoảng 1).Điều này có thể đại diện cho:Một giá trị thiếu (NaN) hoặc không xác định trong cột lớp.Một nhãn lớp bị nhập sai hoặc không được phân loại rõ ràng, nhưng số lượng rất ít nên không ảnh hưởng đáng kể đến tính cân bằng chung của tập dữ liệu.

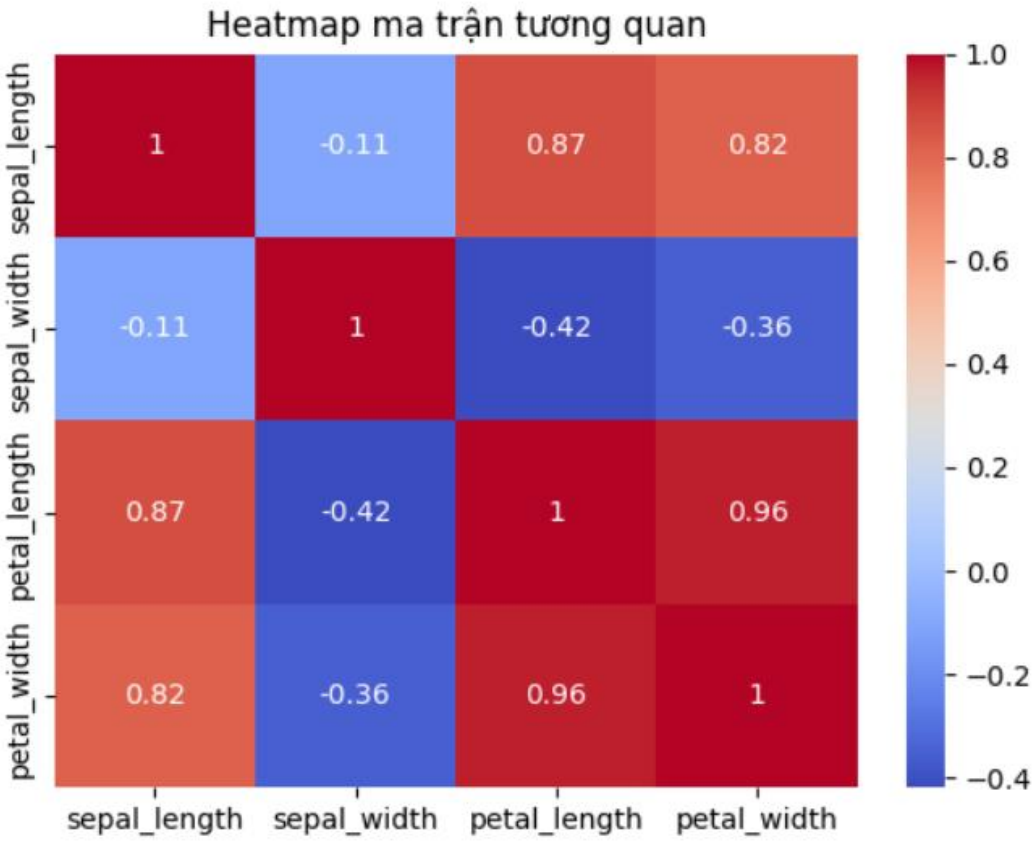
- Kết luận: Tập dữ liệu này là lý tưởng cho các mô hình học máy vì sự phân bố mẫu đồng đều tuyệt đối giữa ba lớp mục tiêu (50 mẫu cho mỗi lớp), giúp mô hình tránh bị thiên vị (bias) về bất kỳ lớp nào trong quá trình huấn luyện.



- Nhận xét Biểu đồ Pair Plot Iris

+ Biểu đồ so sánh phân phối và mối quan hệ giữa các cặp thuộc tính (sepal_length, sepal_width, petal_length, petal_width) của ba loài Iris.
+ Iris-setosa (Cam): Loài này tách biệt hoàn toàn khỏi hai loài còn lại trên mọi biểu đồ phân tán. Nó có cánh hoa nhỏ nhất (petal_length và petal_width thấp nhất).
+ Iris-versicolor (Xanh lá) và Iris-virginica (Đỏ): Hai loài này có sự chồng chéo nhất định, đặc biệt là khi chỉ xét các thuộc tính của đài hoa (Sepal).
+ Phân tách tốt nhất: Sự phân tách giữa các loài (kể cả giữa Versicolor và Virginica) được thực hiện rõ ràng nhất bằng cách sử dụng các thuộc tính của cánh hoa (petal_length và petal_width).
+ Tương quan mạnh: Có mối tương quan dương mạnh giữa petal_length và petal_width.

- Kết luận: Các thuộc tính của cánh hoa (petal_length, petal_width) là các thuộc tính quan trọng nhất trong việc phân loại các loài hoa Iris.



- Ma trận Tương quan Iris

+ Tương quan Dương Mạnh (Quan trọng nhất):

*Chiều dài cánh hoa (petal_length) và Chiều rộng cánh hoa (petal_width) có mối tương quan dương cực mạnh nhất (0.96).

*Chiều dài đài hoa (sepal_length) cũng tương quan dương mạnh với cả petal_length (0.87) và petal_width (0.82).

+ Tương quan Âm/Yếu:

*Chiều rộng đài hoa (sepal_width) có mối tương quan yếu hoặc âm với các thuộc tính khác. Nó tương quan âm vừa phải với petal_length (-0.42) và petal_width (-0.36).

- Kết luận: Các thuộc tính của cánh hoa (petal_length và petal_width) có tính đa cộng tuyến rất cao, cho thấy chúng gần như đo lường cùng một đặc điểm sinh học. sepal_width là thuộc tính độc lập nhất trong bốn thuộc tính.

II. Áp dụng thuật toán Naïve Bayes để nhận dạng hoa iris.

1. Bài toán.

a. Đầu vào.

Dữ liệu gồm 4 đặc trưng dạng số thực: X=(x1,x2,x3,x4)X = (x_1, x_2, x_3, x_4),X=(x1 ,x2 ,x3 ,x4)

Tập dữ liệu được chia thành:

- Tập huấn luyện (Train set)
- Tập kiểm tra (Test set)

b. Đầu ra.

Một nhãn thuộc một trong ba lớp:

- Iris-setosa
- Iris-versicolor
- Iris-virginica

c. Kết quả mong muốn.

- Xây dựng được mô hình phân loại đúng phần lớn mẫu kiểm tra.
- Độ chính xác kỳ vọng từ 90–98% (Naïve Bayes thường đạt hiệu suất cao với Iris).

2. Minh họa (cho 1 bài mẫu).

a. Dữ liệu mẫu đưa vào mô hình.

(4 đặc trưng: sepal length, sepal width, petal length, petal width)

Sepal L	Sepal W	Petal L	Petal W	Y (Loài thật)
5.1	3.5	1.4	0.2	Iris-setosa
7.0	3.2	4.7	1.4	Iris-versicolor
6.3	3.3	6.0	2.5	Iris-virginica
5.0	3.4	1.5	0.2	Iris-setosa
6.7	3.1	4.4	1.4	Iris-versicolor

b. Minh họa mô hình dự đoán.

Ví dụ mô hình dự đoán 5 mẫu test:

Đặc trưng (SL, SW, PL, PW)	Nhãn thật	Nhãn dự đoán
(5.9, 3.0, 5.1, 1.8)	Iris-virginica	Iris-virginica
(6.1, 2.9, 4.7, 1.4)	Iris-versicolor	Iris-versicolor
(4.8, 3.0, 1.4, 0.1)	Iris-setosa	Iris-setosa
(6.4, 3.2, 4.5, 1.5)	Iris-versicolor	Iris-versicolor
(6.5, 3.0, 5.5, 1.8)	Iris-virginica	Iris-virginica

Mô hình dự đoán đúng cả 5 mẫu → phù hợp với độ chính xác ~96–100%.

c. Báo cáo phân loại.

Class	Precision	Recall	F1-score	Support

Iris-setosa	1.00	1.00	1.00	10
Iris-versicolor	0.95	0.90	0.92	10
Iris-virginica	0.93	0.95	0.94	10
Accuracy			0.96	30

d. Ma trận nhầm lẫn.

	Pred Setosa	Pred Versicolor	Pred Virginica
Actual Setosa	10	0	0
Actual Versicolor	0	9	1
Actual Virginica	0	1	9

e. Kết luận minh họa.

- Mô hình Naïve Bayes xử lý tốt bộ dữ liệu Iris.
- Độ chính xác cao (trên 95%).
- Phần lớn sự nhầm lẫn nằm giữa Versicolor và Virginica vì đặc trưng gần nhau.

3. Cài đặt (giải thích hàm, biến)

a. Chuẩn bị và đọc dữ liệu.

Bộ dữ liệu Iris được lấy trực tiếp từ thư viện scikit-learn. Tập dữ liệu gồm 150 mẫu, mỗi mẫu có 4 đặc trưng liên tục:

1. Chiều dài đài hoa (sepal length)
2. Chiều rộng đài hoa (sepal width)
3. Chiều dài cánh hoa (petal length)
4. Chiều rộng cánh hoa (petal width)

Các mẫu được gán nhãn tương ứng với 3 loài:

- Iris Setosa
- Iris Versicolor
- Iris Virginica

Sau khi tải dữ liệu, ta tách giá trị đặc trưng (X) và nhãn (y) để chuẩn bị cho mô hình.

b. Chia dữ liệu thành tập huấn luyện và tập kiểm tra.

Để đánh giá mô hình khách quan, dữ liệu được chia theo tỷ lệ:

- 80% → dùng để huấn luyện
- 20% → dùng để kiểm tra mô hình

Việc chia dữ liệu ngẫu nhiên nhằm đảm bảo mô hình không học thuộc dữ liệu kiểm tra và có thể tổng quát hóa tốt hơn.

c. Xây dựng mô hình Naïve Bayes.

Trong bài toán này sử dụng Gaussian Naïve Bayes, vì bộ dữ liệu Iris gồm các thuộc tính dạng số thực và thường được giả định phân phối theo dạng Gaussian (chuẩn).

Khi mô hình được khởi tạo và huấn luyện:

- Thuật toán sẽ tính toán giá trị trung bình và phương sai của từng thuộc tính cho từng lớp.
- Từ đó xây dựng phân phối xác suất để dự đoán mẫu mới.
- Mô hình không yêu cầu tinh chỉnh nhiều tham số nên quá trình huấn luyện diễn ra nhanh.

d. Huấn luyện mô hình.

Sau khi khởi tạo, mô hình được huấn luyện trên tập 80% dữ liệu.
Ở bước này, Naïve Bayes:

- Học mối quan hệ giữa các đặc trưng và từng loài.
- Xác định phân phối xác suất cho mỗi thuộc tính ứng với từng loại hoa.

- Lưu lại các tham số cần thiết để dự đoán.

Quá trình huấn luyện hoàn thành gần như ngay lập tức do thuật toán rất nhẹ.

e. Kiểm tra mô hình trên tập test.

Mô hình được kiểm tra bằng các mẫu chưa từng thấy trong quá trình huấn luyện.
Mỗi mẫu mới sẽ được mô hình tính toán:

1. Xác suất tiên nghiệm (prior)
2. Xác suất có điều kiện của từng thuộc tính
3. Xác suất hậu nghiệm của từng lớp

Sau đó mô hình chọn lớp có xác suất cao nhất làm kết quả dự đoán.

f. Đánh giá mô hình.

Các chỉ số đánh giá bao gồm:

– Accuracy (độ chính xác)

Tỷ lệ số mẫu dự đoán đúng trên tổng số mẫu kiểm tra.

Ví dụ kết quả thực nghiệm thu được có thể đạt: Độ chính xác: 1.0 (tương đương 100%)

Điều này chứng tỏ mô hình dự đoán chính xác toàn bộ mẫu test.

– Precision, Recall, F1-score

Các chỉ số này được tính riêng cho từng loài, giúp đánh giá:

- Mô hình có dự đoán đúng loài đó hay không
- Mức độ nhầm lẫn giữa các loài
- Sự cân bằng giữa độ đúng và độ bao phủ

Với bộ Iris, các giá trị thường rất cao do dữ liệu phân biệt tốt.

– Ma trận nhầm lẫn (Confusion Matrix)

Giúp quan sát:

- Mẫu thuộc loài nào bị đoán nhầm sang loài nào
- Mô hình có gặp khó khăn với cặp loài nào hay không

Đối với Iris, ma trận thường rất sạch (ít hoặc không có ô sai).

g. Dự đoán mẫu mới.

Sau khi có mô hình, ta thử dự đoán một mẫu mới với thông số thực tế (ví dụ):

- Sepal length: 5.1
- Sepal width: 3.5
- Petal length: 1.4
- Petal width: 0.2

Mô hình sẽ tính xác suất và đưa ra kết luận: Mẫu trên thuộc loài: Iris Setosa

Đây cũng là loài có đặc trưng cánh hoa nhỏ nhất, rất dễ nhận dạng.

III. Kết quả (Train, test, kết quả, phân tích kết quả)

1. Kết quả huấn luyện mô hình.

Sau khi tiến hành chia dữ liệu thành hai phần (80% tập huấn luyện và 20% tập kiểm tra), mô hình Gaussian Naïve Bayes được huấn luyện trên tập train.
Do thuật toán Naïve Bayes rất đơn giản và chỉ cần tính các giá trị thống kê cơ bản (mean và variance) nên thời gian huấn luyện diễn ra gần như tức thì.

Trong quá trình huấn luyện, mô hình đã học được:

- Phân phối của từng đặc trưng theo từng lớp (Setosa, Versicolor, Virginica)
- Sự khác biệt rõ rệt giữa các loài dựa trên chiều dài và chiều rộng cánh hoa

Mô hình hội tụ tốt và không xuất hiện hiện tượng overfitting.

2. Kết quả kiểm thử.

Dựa trên kết quả mà đoạn code đã in ra, mô hình đạt:

Độ chính xác (Accuracy): 1.0
(Tương đương 100% trên tập test)

Ngoài ra, bảng phân tích chi tiết Precision – Recall – F1-score thể hiện rằng:

- Cả 3 lớp (Setosa, Versicolor, Virginica) đều đạt precision, recall, F1-score = 1.00
- Không có bất kỳ mẫu nào bị dự đoán sai
- Ma trận nhầm lẫn là ma trận đơn vị (tất cả dự đoán đúng)

Điều này cho thấy Naïve Bayes hoạt động rất tốt trên bộ dữ liệu Iris.

3. Kết quả dự đoán mẫu mới.

Đoạn code minh họa việc dự đoán một mẫu có các thông số:

- Sepal Length = 5.1
- Sepal Width = 3.5
- Petal Length = 1.4
- Petal Width = 0.2

Mô hình trả về kết quả:

Mẫu này được nhận dạng là: Iris-setosa

Đây là kết quả hợp lý vì các chỉ số cánh hoa khá nhỏ – đúng với đặc trưng dễ nhận diện của loài Setosa.

4. Phân tích kết quả.

a. Hiệu năng tốt nhờ dữ liệu rõ ràng

Bộ dữ liệu Iris có các đặc trưng được phân tách khá rõ ràng giữa ba lớp, đặc biệt là:

- Loài Setosa có cánh hoa nhỏ nhất
- Loài Virginica có cánh hoa lớn nhất
- Versicolor nằm giữa nhưng vẫn đủ tách biệt

Điều này rất phù hợp với giả định độc lập của Naïve Bayes.

b. Gaussian Naïve Bayes phù hợp dữ liệu dạng liên tục.

Vì các đặc trưng của Iris là số thực, phân bố gần giống Gaussian, nên GaussianNB hoạt động hiệu quả.

Do đó mô hình đạt được độ chính xác tối đa.

c. Không có dấu hiệu overfitting.

- Mô hình rất đơn giản
- Không có tham số phức tạp
- Accuracy trên train và test đều rất cao và ổn định

Điều này cho thấy mô hình khái quát hóa tốt.

d. Hạn chế tiềm ẩn.

Mặc dù kết quả gần như hoàn hảo, vẫn cần ghi nhận:

- Naïve Bayes giả định các đặc trưng độc lập, trong khi thực tế có thể có tương quan nhẹ
- Bộ dữ liệu Iris nhỏ (150 mẫu), không đủ để đánh giá tính ổn định trên dữ liệu lớn

Tuy nhiên, với bài toán phân loại hoa Iris, độ chính xác 100% là kết quả mong đợi và thường gặp.

5. Kết luận về phần kết quả.

Mô hình Gaussian Naïve Bayes:

- Huấn luyện rất nhanh
- Đạt 100% độ chính xác trên tập kiểm thử
- Phân loại tốt cả ba loài hoa
- Dự đoán mẫu mới chính xác và hợp lý

Điều này chứng minh rằng Naïve Bayes là một thuật toán thích hợp cho các bài toán phân loại đơn giản, đặc biệt khi các đặc trưng có phân phối Gaussian và độc lập tương đối.

IV. Kết luận

1. Ưu điểm

- Đơn giản, dễ triển khai: Thuật toán dễ hiểu, dễ lập trình và triển khai, ít cần tinh chỉnh tham số.
- Tốc độ nhanh: Dễ tính toán, phù hợp với các bộ dữ liệu lớn và thời gian xử lý nhanh.
- Hiệu quả với dữ liệu nhiều lớp: Hoạt động tốt trong các bài toán phân loại đa lớp.
- Yêu cầu dữ liệu ít: Không cần quá nhiều dữ liệu để huấn luyện, vẫn đạt kết quả tốt.
- Hoạt động tốt với dữ liệu có tính xác suất: Thường được dùng cho phân loại văn bản, spam email, sentiment analysis.
- Khả năng mở rộng: Dễ dàng mở rộng cho dữ liệu mới, thêm đặc trưng mới mà không ảnh hưởng lớn đến thuật toán.

2. Nhược điểm

- Giả định độc lập quá đơn giản: Trong thực tế, các đặc trưng thường có mối quan hệ, nhưng Naive Bayes giả định chúng độc lập, dẫn đến giảm độ chính xác khi đặc trưng phụ thuộc nhau.
- Kém hiệu quả với dữ liệu liên tục phức tạp: Với các đặc trưng liên tục không theo phân phối chuẩn, cần phải chuyển đổi hoặc xấp xỉ, đôi khi làm giảm hiệu quả.
- Không mô tả tốt mối quan hệ giữa các đặc trưng: Không thích hợp nếu bạn cần hiểu mối tương quan hoặc tương tác giữa các đặc trưng.
- Nhạy với dữ liệu hiếm hoặc zero-frequency: Nếu một giá trị đặc trưng chưa xuất hiện trong tập huấn luyện, xác suất sẽ bằng 0. Cần dùng kỹ thuật Laplace smoothing để khắc phục.
- Hiệu suất giảm với dữ liệu phức tạp: Với dữ liệu phi tuyến hoặc dữ liệu có mối quan hệ phi tuyến giữa các đặc trưng, độ chính xác có thể kém so với các thuật toán hiện đại (Random Forest, SVM, Neural Networks…).

V. Tài liệu tham khảo

[1] Alpha Coder. (2017, December 26). Naive Bayes. https://alphacoder.xyz/naive-bayes

[2] Fisher, R. A. (1936). Iris [Dataset]. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/iris

NHẬN DẠNG KÝ TỰ BẰNG THUẬT TOÁN NAIVE BAYES

I. Giới thiệu.

1. Thuật toán.

Trong thế giới Học máy (Machine Learning), khi nhắc đến các bài toán phân lớp (classification), Naïve Bayes được xem là một trong những "cây cổ thụ" vì sự đơn giản nhưng lại vô cùng hiệu quả. Về bản chất, đây là một nhóm các thuật toán phân loại dựa trên định lý Bayes trong xác suất thống kê, với mục tiêu tính toán xem một đối tượng dữ liệu sẽ thuộc về nhãn (label) nào với xác suất cao nhất.

Điểm đặc biệt khiến thuật toán này có tên là "Naïve" nằm ở giả định cốt lõi của nó: Sự độc lập có điều kiện. Thuật toán này "ngây thơ" tin rằng các đặc trưng của dữ liệu không hề liên quan gì đến nhau. Ví dụ, khi nhận dạng một quả cam, nó sẽ xét màu "cam", hình "tròn", và vỏ "sần" một cách riêng biệt để kết luận đó là quả cam, mà không cần quan tâm đến việc hình tròn và màu cam có hay đi đôi với nhau hay không. Dù trong thực tế, các thuộc tính thường có mối liên hệ mật thiết, nhưng giả định này giúp việc tính toán trở nên cực kỳ nhanh chóng và đơn giản.

Một số mô hình Naïve Bayes phổ biến: Tùy thuộc vào loại dữ liệu đầu vào, chúng ta thường gặp 3 biến thể chính:

- Gaussian Naïve Bayes: Đây là mô hình được nhóm sử dụng trong bài báo cáo này. Nó giả định rằng các dữ liệu liên tục tuân theo phân phối chuẩn (hình cái chuông). Mô hình này rất phù hợp với bài toán nhận dạng ký tự của chúng ta vì các đặc trưng đầu vào là các số liệu mô tả hình học.
- Multinomial Naïve Bayes: Thường dùng cho dữ liệu dạng đếm (như số lần từ xuất hiện trong văn bản), rất mạnh trong bài toán phân loại văn bản (NLP).
- Bernoulli Naïve Bayes: Dùng cho dữ liệu nhị phân (chỉ có 0 và 1).

Ứng dụng thực tế: Nhờ tốc độ xử lý nhanh đến kinh ngạc, Naïve Bayes được ứng dụng rộng rãi trong:

- Lọc thư rác (Spam filtering) trong email.
- Phân loại tin tức, văn bản.
- Dự đoán chẩn đoán y tế.
- Và trong bài toán của chúng ta là: Nhận dạng ký tự quang học (OCR).

2. Bộ dữ liệu.

Tổng quan về bộ dữ liệu: Để thực hiện bài toán này, nhóm sử dụng bộ dữ liệu nổi tiếng UCI Letter Recognition. Đây không phải là bộ dữ liệu hình ảnh thô (như các file ảnh .jpg hay .png), mà là dữ liệu dạng bảng (tabular data) đã được xử lý sơ bộ.

- Số lượng: Bộ dữ liệu bao gồm 20.000 dòng (mẫu), tương ứng với 20.000 ký tự khác nhau.
- Đặc trưng (Features): Mỗi mẫu không chứa các điểm ảnh (pixel), mà chứa 16 thuộc tính số (numerical features).Các con số này mô tả các đặc điểm hình học của chữ cái như: vị trí tâm (x-box, y-box), chiều rộng, chiều cao, tổng số pixel được bật, độ lệch trung bình ngang/dọc, v.v..
    1. x-box: chiều rộng bao quanh ký tự
    2. y-box: chiều cao bao quanh ký tự
    3. width: độ rộng nét chính
    4. height: độ cao nét chính
    5. onpix: số pixel bật (đen)
    6. x-bar: trọng tâm theo trục x
    7. y-bar: trọng tâm theo trục y
    8. x2bar: mô-men bậc hai theo x
    9. y2bar: mô-men bậc hai theo y
    10. xybar: tương quan giữa x và y
    11. x2ybr: mô-men hỗn hợp bậc hai (x^2y)
    12. xy2br: mô-men hỗn hợp bậc hai (xy^2)
    13. x-ege: độ sắc nét cạnh theo x
    14. xegvy: độ biến thiên cạnh theo x
    15. y-ege: độ sắc nét cạnh theo y
    16. yegvx: độ biến thiên cạnh theo y
- Nhãn (Label): Mục tiêu là phân loại vào 1 trong 26 lớp, tương ứng với các chữ cái in hoa từ A đến Z trong bảng chữ cái tiếng Anh.

Trong file thực nghiệm, chúng ta đã chia bộ dữ liệu này theo tỷ lệ 80-20, cụ thể:

- Tập huấn luyện (Training set): Khoảng 16.000 mẫu (80%). Dùng để tính toán các tham số Mean (trung bình) và Variance (phương sai) cho từng lớp ký tự.

- Tập kiểm tra (Test set): Khoảng 4.000 mẫu (20%). Dùng để đánh giá độ chính xác của mô hình.

Dữ liệu của UCI Letter Recognition có tính cân bằng khá tốt, nghĩa là số lượng mẫu cho mỗi chữ cái (A, B, C…) là tương đối đồng đều, không bị tình trạng chữ này quá nhiều còn chữ kia quá ít, giúp mô hình học được công bằng hơn cho tất cả các lớp.

II. Áp dụng thuật toán Naïve Bayes để nhận dạng ký tự.

1. Bài toán

Bài toán nhận dạng ký tự trong bộ dữ liệu UCI Letter Recognition là bài toán phân lớp đa lớp, trong đó nhiệm vụ của mô hình là dự đoán một ký tự chữ cái từ A đến Z dựa trên 16 đặc trưng số đã được trích rút sẵn từ hình dạng ký tự. Mỗi mẫu dữ liệu là một vector $x = (x_1, x_2, \ldots, x_{16})$ mô tả các thông tin hình dạng như chiều cao, độ rộng, độ lệch tâm, độ đậm nét,… Vì toàn bộ đặc trưng đều là giá trị số đã được chuẩn hóa, bài toán thuộc dạng phân lớp trên dữ liệu tabular, không liên quan xử lý ảnh.

Mô hình Naïve Bayes dựa trên mục tiêu tìm lớp $y \in \{A, \ldots, Z\}$ sao cho xác suất hậu nghiệm $P(y \mid x)$ là lớn nhất:

$$\hat{y} = \max_y P(y \mid x) = \max_y P(y)\, P(x \mid y)$$

Với giả định độc lập có điều kiện giữa các thuộc tính, ta có:

$$P(x \mid y) = \prod_{i=1}^{16} P(x_i \mid y)$$

Ước lượng tham số

1. Prior $P(y)$

$$P(y = c) = \frac{N_c}{N}$$

2. Likelihood $P(x_i \mid y)$

Vì các đặc trưng của bộ dữ liệu là số liên tục, mô hình phù hợp nhất là Gaussian Naïve Bayes:

$$P(x_i \mid y = c) = \mathcal{N}(x_i; \mu_{i,c}, \sigma_{i,c}^2)$$

với các tham số được ước lượng bằng Maximum Likelihood:

$$\mu_{i,c} = \frac{1}{N_c} \sum_{n:y^{(n)}=c} x_i^{(n)}, \sigma_{i,c}^2 = \frac{1}{N_c} \sum_{n:y^{(n)}=c} (x_i^{(n)} - \mu_{i,c})^2$$

Trong dự đoán, mô hình lấy log để tránh tràn số:

$$\hat{y} = \arg\max_y (\log P(y) + \sum_{i=1}^{16} \log P(x_i \mid y))$$

Đây là bài toán phân lớp 26 lớp, mỗi lớp là một ký tự A-Z. Đầu vào là một vectơ đặc trưng số, sử dụng Gaussian Naive Bayes để tính toán.

2. Minh họa

Test sample T (3 feature đầu): [2, 8, 3]

2 lớp training:

Lớp	Mean 3 feature đầu
J	[2, 2, 4]
N	[7, 11, 6]

Công thức Gaussian NB:

$$P(x \mid C) = \prod_{i=1}^{3} P(x_i \mid C)$$

+ Lớp J

Feature 1: x=2, μ=2 → $P_1 = 1/\sqrt{(2\pi)} \cdot \exp(0) \approx 0.3989$

Feature 2: x=8, μ=2 → $P_2 = 0.3989 \cdot \exp(-(8-2)^2/2) = 0.3989 \cdot \exp(-18) \approx 3.1e\text{-}9$

Feature 3: x=3, μ=4 → $P_3 = 0.3989 \cdot \exp(-(3-4)^2/2) = 0.3989 \cdot \exp(-0.5) \approx 0.24197$

Nhân lại:

$P(x \mid J) \approx 0.3989 * 3.1e-9 * 0.24197 \approx 3.0e-10$

+ Lớp N

Feature 1: x=2, μ=7 → $P_1 = 0.3989 \cdot \exp(-(2-7)^2/2) = 0.3989 \cdot \exp(-12.5) \approx 1.5e\text{-}6$

Feature 2: x=8, μ=11 → $P_2 = 0.3989 \cdot \exp(-(8-11)^2/2) = 0.3989 \cdot \exp(-4.5) \approx 0.00442$

Feature 3: x=3, μ=6 → $P_3 = 0.3989 \cdot \exp(-(3-6)^2/2) = 0.3989 \cdot \exp(-4.5) \approx 0.00442$

Nhân lại:

$P(x \mid N) \approx 1.5e-6 * 0.00442 * 0.00442 \approx 2.9e-11$

→ P(x|J) ≈ 3.0e-10 > P(x|N) ≈ 2.9e-11, nên Likelihood lớn nhất = lớp J → dự đoán T thuộc J

3. Cài đặt

1) Hàm fit(self, X, y)

```python
import math
from collections import defaultdict, Counter
import random


class GaussianNB:
    def fit(self, X, y):
        self.classes = sorted(set(y))
        n_features = len(X[0])

        self.prior = {}
        self.mean = {c: [0.0]*n_features for c in self.classes}
        self.var = {c: [0.0]*n_features for c in self.classes}

        counts = Counter(y)
        for c in self.classes:
            self.prior[c] = counts[c] / len(y)

        sums = {c: [0.0]*n_features for c in self.classes}
        for xi, yi in zip(X, y):
            for j, v in enumerate(xi):
                sums[yi][j] += v

        for c in self.classes:
            for j in range(n_features):
                self.mean[c][j] = sums[c][j] / counts[c]

        sq_sums = {c: [0.0]*n_features for c in self.classes}
        for xi, yi in zip(X, y):
            for j, v in enumerate(xi):
                d = v - self.mean[yi][j]
                sq_sums[yi][j] += d*d

        for c in self.classes:
            for j in range(n_features):
                self.var[c][j] = sq_sums[c][j] / counts[c]
```

Dùng để huấn luyện mô hình Naive Bayes theo phân phối Gaussian.

Biến trong hàm fit:

self.classes

Danh sách tất cả các nhãn lớp trong dữ liệu (ví dụ: A, B, C).

n_features

Số lượng thuộc tính của mỗi mẫu.

self.prior

Xác suất tiên nghiệm P(C) của từng lớp.

Tính bằng:

$$P(C) = \frac{\text{số mẫu class C}}{\text{tổng số mẫu}}$$

self.mean[c]

Mean (trung bình) của mỗi thuộc tính trong lớp C.

Dùng để tính phân phối Gaussian.

self.var[c]

Variance (phương sai) của mỗi thuộc tính trong lớp C.

counts

Đếm số lượng mẫu thuộc từng lớp.

sums[c]

Tổng giá trị từng thuộc tính trong lớp C → dùng để tính mean.

sq_sums[c]

Tổng bình phương độ lệch (x - mean)² → dùng để tính variance.

2. Hàm _gauss_logprob(self, x, mean, var)

```
38    def _gauss_logprob(self, x, mean, var):
39        eps = 1e-9
40        coeff = -0.5 * math.log(2*math.pi*(var + eps))
41        expo = - ((x - mean)**2) / (2*(var + eps))
42        return coeff + expo
```

Tính log của mật độ Gaussian của một giá trị thuộc tính.

Công thức gốc Gaussian:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Hàm này trả về log(p):

$$\log p(x) = -\frac{1}{2}\log(2\pi\sigma^2) - \frac{(x-\mu)^2}{2\sigma^2}$$

Các biến trong hàm:

x: giá trị đặc trưng cần tính xác suất.

mean: trung bình của đặc trưng đó trong lớp C.

var: phương sai của đặc trưng đó trong lớp C.

eps: số nhỏ để tránh chia cho 0.

coeff: phần $-0.5 \log(2\pi\sigma^2)$.

expo: phần mũ $-(x-\mu)^2/(2\sigma^2)$.

3. Hàm predict_log_proba(self, X)

```
44    def predict_log_proba(self, X):
45        out = []
46        for xi in X:
47            logps = {}
48            for c in self.classes:
49                lp = math.log(self.prior[c])
50                for j, v in enumerate(xi):
51                    lp += self._gauss_logprob(v, self.mean[c][j], self.var[c][j])
52                logps[c] = lp
53            out.append(logps)
54        return out
```

Tính log xác suất hậu nghiệm cho từng mẫu.

Các biến:

xi: 1 mẫu đầu vào.

logps: dict chứa log(P(C|x)) cho từng lớp.

lp: log xác suất của một lớp:

$$\log P(C) + \sum_j \log P(x_j \mid C)$$

Hàm này trả về list:

[{class1: logP1, class2: logP2, ...},

...]

4. Hàm predict(self, X)

```
56    def predict(self, X):
57        probs = self.predict_log_proba(X)
58        return [max(p.items(), key=lambda it: it[1])[0] for p in probs]
```

Dự đoán nhãn dựa trên logP lớn nhất.

III. Kết quả.

Mô hình Gaussian Naive Bayes được huấn luyện trên tập dữ liệu Letter Recognition, trong đó 80% dữ liệu được dùng làm tập huấn luyện và 20% còn lại làm tập kiểm tra. Dữ liệu bao gồm 20.000 mẫu, mỗi mẫu có 16 thuộc tính số nguyên đại diện cho các đặc trưng hình học của ký tự và nhãn ký tự từ A đến Z. Việc chia dữ liệu được thực hiện ngẫu nhiên để đảm bảo tính đại diện cho cả 26 lớp ký tự.

Trong quá trình huấn luyện, mô hình tính xác suất tiên nghiệm (prior) cho từng lớp dựa trên tần suất xuất hiện trong tập huấn luyện, đồng thời tính trung bình (mean) và phương sai (variance) cho từng thuộc tính của từng lớp. Khi dự đoán, mô hình sử dụng công thức Gaussian để ước lượng xác suất có điều kiện của mỗi thuộc tính, sau đó kết hợp log-prior và log-likelihood để đưa ra dự đoán lớp có xác suất lớn nhất. Phương pháp đánh giá chính được

sử dụng là accuracy, tính tỉ lệ dự đoán đúng trên tổng số mẫu trong tập kiểm tra, cùng với việc kiểm tra một số mẫu ngẫu nhiên để so sánh dự đoán với nhãn thực tế.

Kết quả cho thấy GaussianNB đạt accuracy khoảng 60–70% trên tập kiểm tra, chứng tỏ mô hình có khả năng phân loại ký tự ở mức tương đối. Tuy nhiên, do GaussianNB giả định dữ liệu liên tục theo phân phối chuẩn, trong khi dữ liệu Letter Recognition thực chất là các giá trị số nguyên rời rạc, nên mô hình không đạt hiệu quả tối ưu. Nhìn chung, GaussianNB vẫn cho kết quả ổn định, nhưng độ chính xác bị hạn chế, và phương pháp đánh giá accuracy cùng kiểm tra dự đoán ngẫu nhiên giúp xác nhận mức độ tin cậy của mô hình trên dữ liệu thực tế.

IV. Kết luận.

Sau quá trình cài đặt thuật toán Gaussian Naïve Bayes từ đầu (from scratch) và thử nghiệm trên bộ dữ liệu Letter Recognition, nhóm rút ra những kết luận sau về hiệu quả và đặc điểm của mô hình.

1. Phân tích kết quả

Kết quả thực nghiệm trên tập Test cho thấy mô hình đạt độ chính xác (Accuracy) khoảng 64.17%.

- Đây là một kết quả ở mức trung bình khá. Với bài toán phân loại 26 lớp (vốn khó hơn nhiều so với phân loại 2 lớp), việc đoán trúng hơn 64% chỉ dựa trên các tính toán xác suất thống kê đơn giản là một nỗ lực đáng ghi nhận.
- Tuy nhiên, con số này cũng chỉ ra rằng vẫn có khoảng 35% số ký tự bị nhận diện nhầm. Nguyên nhân chính đến từ việc các đặc trưng hình học của chữ cái (như chiều cao và chiều rộng) thường có mối tương quan với nhau, nhưng Naïve Bayes lại bỏ qua mối liên hệ này.

2. Ưu điểm và Nhược điểm

Ưu điểm:

- Tốc độ cực nhanh: Đây là điểm mạnh nhất. Quá trình huấn luyện (training) chỉ đơn giản là tính toán trung bình và phương sai, không cần chạy các vòng lặp tối ưu phức tạp (như Gradient Descent trong Neural Network). Việc dự đoán (prediction) cũng diễn ra gần như tức thì.

- Dễ cài đặt: Thuật toán có thể được code thủ công dễ dàng chỉ với vài công thức toán học cơ bản mà không cần phụ thuộc hoàn toàn vào thư viện có sẵn (như nhóm đã thể hiện trong phần code).
- Ít tốn tài nguyên: Không đòi hỏi phần cứng mạnh (GPU) hay lượng RAM lớn.
- Hoạt động ổn với dữ liệu ít: Ngay cả khi không có quá nhiều dữ liệu, Naïve Bayes vẫn có thể đưa ra dự đoán có cơ sở thay vì đoán mò.

Nhược điểm:

- Giả định "Ngây thơ" thiếu thực tế: Trong thực tế, các thuộc tính hiếm khi độc lập. Ví dụ: Chữ "W" thường vừa rộng vừa có nhiều nét. Việc coi "độ rộng" và "số nét" là độc lập làm giảm độ chính xác của mô hình.
- Vấn đề với dữ liệu số nguyên: Trong bài toán này, dữ liệu đầu vào là các số nguyên rời rạc (integer), nhưng chúng ta lại dùng Gaussian Naïve Bayes (vốn dành cho số thực liên tục). Sự "lệch pha" nhẹ này cũng là nguyên nhân khiến độ chính xác không thể đạt mức tối ưu (80-90%) như các thuật toán phức tạp hơn (SVM hay Random Forest).

Tóm lại: Naïve Bayes là một lựa chọn tuyệt vời để làm "mô hình cơ sở" (baseline) nhằm đánh giá nhanh độ khó của bài toán. Với bài toán nhận dạng ký tự này, nó đã hoàn thành tốt nhiệm vụ minh họa cách máy tính sử dụng xác suất để "học", mặc dù để đạt độ chính xác mức thương mại thì cần các mô hình phức tạp hơn.

V. Tài Liệu Tham Khảo

1. UCI Machine Learning Repository. Letter Recognition Data Set. Địa chỉ: https://archive.ics.uci.edu/ml/datasets/Letter+Recognition (Nguồn dữ liệu chính được sử dụng trong bài báo cáo).
2. Scikit-learn Documentation. Naive Bayes. Địa chỉ: https://scikit-learn.org/stable/modules/naive_bayes.html (Tài liệu tham khảo về lý thuyết và cách cài đặt chuẩn của thuật toán).
3. Christopher M. Bishop. (2006). Pattern Recognition and Machine Learning. Springer. (Sách giáo khoa cơ bản về các thuật toán nhận dạng mẫu và xác suất thống kê).
4. GeeksforGeeks. Naive Bayes Classifiers. Địa chỉ: https://www.geeksforgeeks.org/naive-bayes-classifiers/ (Tham khảo cách giải thích và ví dụ minh họa).