ĐẠI HỌC HUẾ

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

BÁO CÁO ĐỒ ÁN

Học kỳ II, năm học 2023 - 2024

Học phần:

Học máy 2

Chủ đề: Phân tích và thiết kế thuật toán cho mô hình Logistic Regression

Số phách
(Do hội đồng chấm thi ghi)

Thừa Thiên Huế, tháng 6 năm 2024

ĐẠI HỌC HUẾ

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

BÁO CÁO ĐỒ ÁN

Học kỳ II, năm học 2023 - 2024

Học phần:

Học máy 2

Giảng viên hướng dẫn: TS. Nguyễn Đăng Trị

Lớp: Khoa học dữ liệu và trí tuệ nhân tạo K3

Sinh viên thực hiện:

Hồ Tăng Nhật Hiếu – 22E1010006

Văn Khiêm Chương – 22E1020002

Số phách
(Do hội đồng chấm thi ghi)

Thừa Thiên Huế, tháng 6 năm 2024

LỜI CÁM ƠN

Kính gửi thầy Nguyễn Đăng Trị, chúng em xin gửi lời cám ơn sâu sắc nhất đến thầy về sự hướng dẫn và chỉ bảo mà thầy đã dành cho đồ án của chúng em. Đây là một hành trình học tập đầy thách thức và ý nghĩa, và chúng em cảm thấy may mắn được có thầy là người hướng dẫn.

Thầy đã không ngừng truyền đạt kiến thức chuyên sâu, sự tận tâm của thầy đã giúp chúng em vượt qua những khó khăn, cũng như mở mang tầm nhìn và ý thức trong lĩnh vực chuyên ngành của mình.

Chúng em muốn bày tỏ lòng biết ơn đặc biệt đến những góp ý chi tiết và xây dựng của thầy, giúp chúng em hoàn thiện đồ án một cách toàn diện hơn. Sự quan tâm và tận tâm của thầy không chỉ giúp chúng em hoàn thành đồ án mà còn giúp chúng em phát triển kỹ năng.

Chúng em cảm thấy tự hào và biết ơn vì đã có một người hướng dẫn như thầy, người luôn sẵn lòng chia sẻ kiến thức và kinh nghiệm.

Chúng em xin cảm ơn thầy một lần nữa vì sự hướng dẫn của Thầy trong suốt học kỳ này. Chúng em hi vọng sẽ tiếp tục nhận được sự hướng dẫn tận tâm của thầy trong tương lai.

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

MỤC LỤC

1: PHÁT BIỂU ĐỊNH NGHĨA

- Hồi quy logistic là một phương pháp trong thống kê và máy học được sử dụng để mô hình hóa và dự đoán xác suất của một biến phụ thuộc nhị phân (binary outcome), có nghĩa là biến phụ thuộc chỉ nhận một trong hai giá trị có thể: 0 hoặc 1, "có" hoặc "không", "thành công" hoặc "thất bại",...

- Hồi quy logistic sử dụng hàm logistic (hoặc hàm sigmoid) để chuyển đổi đầu ra của một hàm tuyến tính thành một giá trị nằm trong khoảng (0,1).

- Dựa trên các loại hồi quy logistic, có thể được phân thành ba loại:

1. **Nhị phân:** Trong hồi quy logistic nhị thức, chỉ có thể có hai loại biến phụ thuộc, chẳng hạn như 0 hoặc 1, Đạt hoặc Thất bại …
2. **Đa thức:** Trong hồi quy logistic đa thức, có thể có 3 hoặc nhiều loại biến phụ thuộc không có thứ tự, chẳng hạn như "mèo", "chó" hoặc "cừu".
3. **Thứ tự:** Trong hồi quy logistic thứ tự, có thể có 3 loại biến phụ thuộc được sắp xếp theo thứ tự trở lên, chẳng hạn như "Thấp", "Trung bình" hoặc "Cao".

2: MÔ HÌNH TOÁN

- Logistic Regression tương tự như Linear Regression ở chỗ cả hai đều tìm kiếm một hàm phi tuyến tính $\hat{y} = \mathrm{Wx} + \mathrm{b}$ để mô hình hóa dữ liệu. Tuy nhiên, điểm khác biệt quan trọng là đầu ra của Logistic Regression sẽ được đưa qua hàm sigmoid , giúp giới hạn kết quả trong khoảng (0,1).

2.1 Công thức toán học

$$\emptyset(x) = \frac{1}{1 + e^{x}}$$

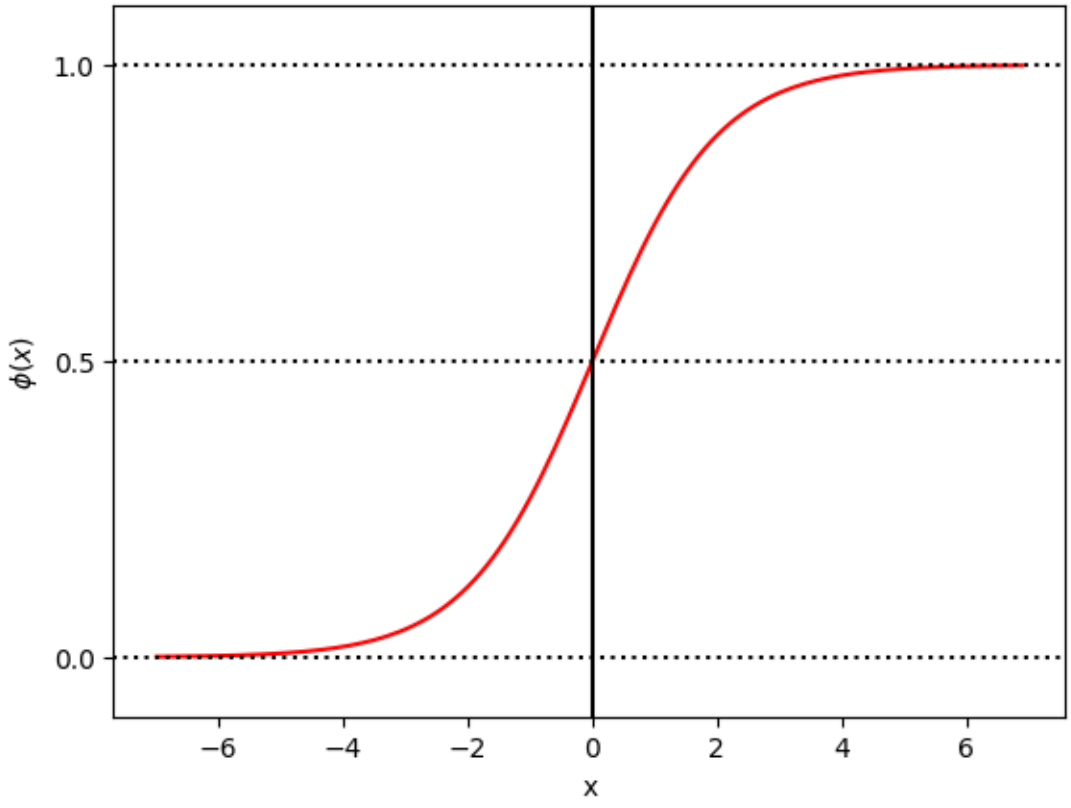
Trong đó:

$\emptyset(x)$: đầu ra trong khoảng từ 0 đến 1 (giá trị xác suất ước lượng).

x: đầu vào của hàm, (giá trị dự đoán của thuật toán, ví dụ như $mx + b$) hoặc là sự kết hợp tuyến tính của trọng số và các số tính năng mẫu và có thể được tính như sau: $z = w^{T}x = w_0 + w_1x_1 + \cdots + w_mx_m$.

e: hằng số toán học Euler, và là cơ số của logarit tự nhiên.

2.2 Đồ thị



Hình 1: Trực quan hóa đồ thị của hàm Sigmoid

Code: Hàm Sigmoid

- Hàm sigmoid có hình dạng là một đường cong chữ S và đơn điệu tăng. Vì vậy, nó còn được gọi là hàm chữ S hoặc hàm Logistic.

- Chứng minh giá trị của hàm Sigmoid nằm trong khoảng [0,1].

$$\lim_{x\to+\infty} \emptyset(x) = \lim_{x\to+\infty} \frac{1}{1+e^{-x}} = 1$$

Và

$$\lim_{x\to-\infty} \emptyset(x) = \lim_{x\to-\infty} \frac{1}{1+e^{-x}} = 0$$

Vì vậy, hàm Sigmoid rất lý tưởng để sử dụng trong việc dự đoán xác suất cho các bài toán phân loại.

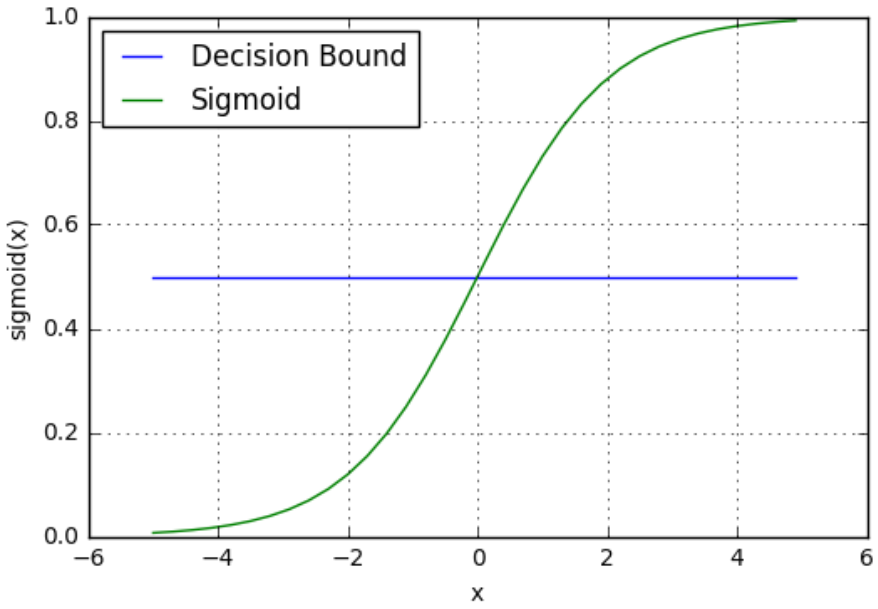
2.2 Ranh giới quyết định

- Hàm dự đoán của ta sẽ trả về giá trị xác suất trong khoảng từ 0 đến 1. Để chuyển xác suất này thành các danh mục rời rạc (đúng/sai; chó/mèo), chúng ta cần chọn một ngưỡng.

$$p \geq 0.5, class = 1$$

$$p < 0.5, class = 0$$

- Ví dụ, nếu ngưỡng là 0.5 và hàm dự đoán trả về 0.7, ta có thể phân loại điểm dữ liệu đó là 1. Nếu dự đoán là 0.2 thì ta có thể phân loại điểm dữ liệu đó là 0.



Hình 2: Ranh giới quyết định

Ranh giới quyết định (màu xanh dương). Các giá trị phía trên ranh giới là 1, còn lại là 0.

2.4 Hàm chi phí

- Chúng ta không nên sử dụng hàm trung bình bình phương sai số **MSE(L2)** như trong hồi quy tuyến tính. Nguyên nhân là do hàm dự đoán trong hồi quy logistic là phi tuyến tính (do biến đổi sigmoid).

- Việc sử dụng bình phương dự đoán như trong **MSE** sẽ tạo ra một hàm không lồi, với nhiều cực tiểu cục bộ. Khi hàm chi phí có nhiều cực tiểu cục bộ, thuật toán gradient descent sẽ gặp khó khăn trong việc tìm ra điểm tối ưu tại điểm toàn cục.

- Thay vào đó, chúng ta nên sử dụng một hàm có tên gọi là: **Binary Cross Entropy Loss – BCE.** Còn được gọi là log loss.

2.4.1 Công thức

$$BCE(y^i, \hat{y}^i) = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i))$$

2.4.2 Ví dụ 1

Nếu mô hình dự đoán là 90% cho positive class và 10% cho negative class thì log loss sẽ là bao nhiêu ?

Actual	Predicted	
1	0.9	= 1 x log(0.9) + (1 - 1) x log(1 - 0.9) = -0.10536
0	0.1	= 0 x log(0.1) + (1 - 0) x log(1 - 0.1) = -0.10536

= -((-0.10536 - 0.10536) / 2)
= 0.10536

Hình 3: Ví dụ 1

- Kết quả là **0.10536**

- Sử dụng phép chia 2 ở bước cuối cùng để lấy trung bình của hai số hạng.

- Chúng ta có thể thấy giá trị của hàm loss BCE khoảng 0.1 (khá là nhỏ). Bởi vì mô hình của chúng ta đang dự đoán đúng và tự tin với xác suất là 90%.

- Tuy nhiên nếu như mô hình tự tin nhưng dự đoán vào class negative thì sao ?

2.4.3 Ví dụ 2

Nếu mô hình dự đoán 10% cho positive class và 90% cho negative class thì log loss sẽ là bao nhiêu ?

Actual	Predicted	
1	0.1	= 1 x log(0.1) + (1 - 1) x log(1 - 0.1) = -2.30259
0	0.9	= 0 x log(0.9) + (1 - 0) x log(1 - 0.9) = -2.30259

= -((-2.30259 - 2.30259) / 2)
= 2.30259

Hình 4: Ví dụ 2

- Kết quả là **2.30259**

- Cho thấy rằng BCE có tác dụng phạt rất nặng các dự đoán sai của mô hình.

3: Ý NGHĨA CỦA BÀI TOÁN TRONG THỰC TẾ

1. Dự đoán email có phải spam hay không: dựa trên các đặc điểm của email như nội dung, tiêu đề, người gửi ...
2. Chẩn đoán y khoa: giúp phân loại bệnh nhân có bệnh hay không dựa trên các yếu tố như triệu chứng, kết quả xét nghiệm, tiền sử bệnh lý...
3. Phân tích rủi ro tài chính: dự đoán khả năng một khách hàng có khả năng không trả được nợ dựa trên lịch sử tín dụng, thu nhập, nợ hiện tại ...
4. Tiếp thị và báo cáo: xác định khả năng một khách hàng sẽ mua một sản phẩm cụ thể dựa trên hành vi mua hàng, nhân khẩu học, tương tác trên mạng xã hội ...
5. Phát hiện giao dịch ngân hàng là gian lận hay không: bằng cách phân tích các mẫu giao dịch và nhận dạng những hành vi bất thường.

4: PHÂN TÍCH VÀ CHỨNG MINH CÁCH THỨC CẬP NHẬT HỆ SỐ

4.1 Tìm kiếm nghiệm tối ưu bằng phương pháp Gradient Descent

4.1.1 Giới thiệu

Phương pháp hạ dốc (gradient descent) là một kỹ thuật quan trọng trong học máy và đặc biệt là học sâu. Nó cho phép chúng ta tiến dần tới các điểm cực trị của hàm số dựa trên gradient của nó. Trong thực tế, tìm kiếm giải pháp chính xác cho một số dạng hàm mất mát có thể rất phức tạp, đặc biệt là khi đạo hàm của chúng quá phức tạp.

Vì vậy, phương pháp hạ dốc trở thành một lựa chọn hợp lý để tiến dần tới các cực trị trong những trường hợp như vậy. Tuy nhiên, hạn chế của phương pháp này là các cực trị tìm được chỉ là các giải pháp gần đúng và không đảm bảo là các cực trị toàn cục.

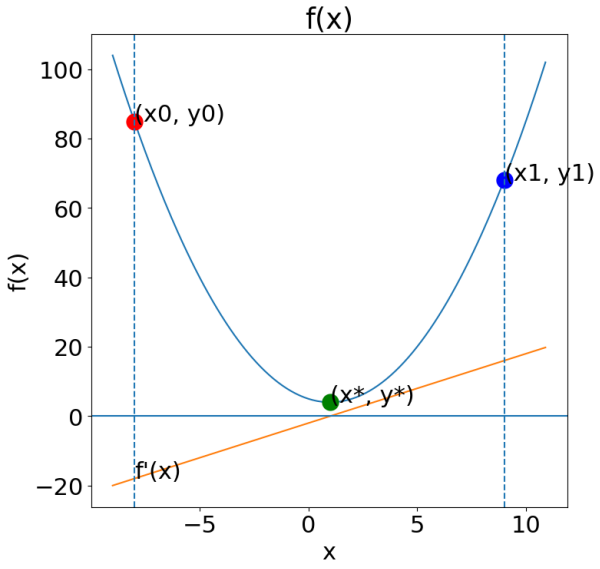
4.1.2 Ví dụ

Để hiểu về phương pháp hạ dốc là gì, chúng ta sẽ cùng phân tích một ví dụ đó là bài toán tìm cực trị của hàm $f(x) = x^2 - 2x + 5$.

Kết quả là: $f'(x) = 2x - 2$.

$f'(x)$ có nghiệm $x = 1$ và là hàm lồi tại nghiệm đó nên có cực tiểu là $(x^*, y^*) = (1,5)$

4.1.3 Vẽ đồ thị



Hình 5: Đồ thị hàm số

Code: [Đồ Thị Hàm Số]

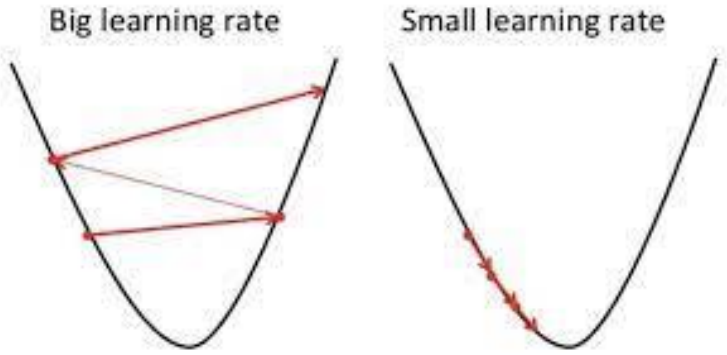
Ta nhận thấy rằng nếu điểm hiện tại nằm ở bên trái của điểm cực tiểu, thì giá trị của đạo hàm là âm. Để di chuyển tới gần cực tiểu, chúng ta cần tăng giá trị của nghiệm. Tương tự, nếu điểm hiện tại nằm bên phải của điểm cực tiểu, thì giá trị của đạo hàm là dương và chúng ta cần giảm giá trị của nghiệm.

Vì vậy, trong cả hai trường hợp, chúng ta đều cần di chuyển ngược hướng đạo hàm để tiến gần hơn tới cực tiểu. Chúng ta có thể cập nhật nghiệm dần dần sau mỗi bước bằng một hệ số học tập (learning rate) có dạng như sau:

$$x_{new} := x_0 - \alpha \nabla_x f(x0)$$

Ở mọi vị trí, chỉ cần di chuyển ngược hướng của đạo hàm một khoảng rất nhỏ, thì khả năng cao là ta sẽ thu được một giá trị nhỏ hơn.

Tuy nhiên, có trường hợp khi di chuyển ngược hướng của đạo hàm lại làm cho giá trị tăng lên. Điều này xảy ra khi chúng ta đã vượt qua cực tiểu, chẳng hạn khi đã đến gần điểm cực tiểu nhưng hệ số học tập quá lớn, dẫn đến sự biến đổi lớn ở bước tiếp theo và khiến cho nghiệm vượt quá điểm cực tiểu. Trường hợp này được gọi là nhảy dốc (Step Over).



Hình 6: Ví dụ về hệ số học tập

Hình bên phải biểu diễn learning rate được thiết lập với hệ số học tập phù hợp. Trong khi đó, hình bên trái thể hiện hiện tượng nhảy dốc. Sau mỗi lượt cập nhật nghiệm, các điểm có xu hướng nhảy qua lại hai bên xung quanh cực trị địa phương thay vì hội tụ từ từ.

Để giảm thiểu hiện tượng nhảy dốc, ta cần chọn learning rate rất nhỏ, thường trong khoảng từ 0.001 đến 0.005, và sử dụng các kỹ thuật tối ưu hóa khác nhau để điều chỉnh quá trình huấn luyện.

4.2 Chứng minh cách thức cập nhật hệ số

4.2.1 Đạo hàm cho hàm sigmoid

$$\emptyset(\mathrm{x})' = \left(\frac{1}{1+e^{-x}}\right)' \qquad (1)$$

- Theo công thức $\frac{1}{a} = a^{-1}$. Ta áp dụng cho phương trình (1), ta được:

$$\emptyset(\mathrm{x})' = [(1+e^{-x})^{-1}]' \qquad (2)$$

- Tiếp theo, ta áp dụng công thức tính đạo hàm riêng của biểu thức $U^n = nU^{n-1}U'$ vào phương trình (2), ta được:

$$\emptyset(\mathrm{x})' = -1(1+e^{-x})^{-1-1}(1+e^{-x})' \qquad (3)$$

- Rút gọn phương trình (3) ta được:

$$\emptyset(\mathrm{x})' = -(1+e^{-x})^{-2}\ (1+e^{-x})' \qquad (4)$$

- Tiếp theo, trong đạo hàm của biểu thức $(1+e^{-x})'$ thì đạo hàm của 1 sẽ là 0, đạo hàm của e^{-x} là e^{-x}, thay vào phương trình (4), ta được:

$$\emptyset(\mathrm{x})' = -(1+e^{-x})^{-2}\ (e^{-x}) \qquad (5)$$

- Chuyển mũ -2 xuống mẫu cho dễ nhân, ta được:

$$\emptyset(\mathrm{x})' = \frac{-1}{(1+e^{-x})^{2}}(-e^{-x}) \qquad (6)$$

- Tiếp tục nhân vào, ta được:

$$\emptyset(\mathrm{x})' = \frac{e^{-x}}{(1+e^{-x})^{2}} \qquad (7)$$

- Nhận thấy mẫu số là một biểu thức có dạng bình phương, ta tách:

$$\emptyset(\mathrm{x})' = \frac{1}{(1+e^{-x})} \times \frac{e^{-x}}{(1+e^{-x})} \qquad (8)$$

- Cộng trừ giá trị 1 vào biểu thức, ta được:

$$\emptyset(\mathrm{x})' = \frac{1}{(1+e^{-x})} \times \frac{(1+e^{-x})-1}{(1+e^{-x})}$$

$$= \frac{1}{(1+e^{-x})} \times \left[\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}}\right]$$

$$= \frac{1}{(1+e^{-x})} \times \left(1 - \frac{1}{1+e^{-x}}\right)$$

$$= \emptyset(x) \times \big(1 - \emptyset(x)\big) \qquad (9)$$

4.2.2 Cực tiểu hóa giá trị của hàm loss BCE

Quay trở lại vấn đề tối ưu, chúng ta nhắm đến việc giảm thiểu giá trị của hàm loss BCE. Trong hàm loss này, có sự hiện diện của hàm sigmoid, cùng với các trọng số và bias. Để tính toán đạo hàm của hàm loss theo trọng số và bias, chúng ta cần áp dụng chain rule, hay còn gọi là đạo hàm của hàm hợp.

Chain rule là một phương pháp quan trọng trong việc tính toán đạo hàm của một hàm số mà chứa các hàm số khác. Điều này có thể được mô tả qua công thức tổng quát sau:

$$[f\big(g(x)\big)]' = f'\big(g(x)\big) \times g'(x) \qquad (10)$$

- Thực hiện đạo hàm:

Bước 1: Tính $\frac{\partial L}{\partial \hat{y}}$.

$$L = -y \times \log(\hat{y}) - (1 - y) \times \log(1 - \hat{y}) \qquad (11)$$

- Xét đạo hàm của logarit, ta có:

$$[n \times \log(x)]' = \frac{n}{x} \qquad (12)$$

- Xét hai thành phần trong hàm loss, ta có:

$$[-y \times \log(\hat{y})]'_{\hat{y}} = \frac{-y}{\hat{y}} \qquad (13)$$

và

$$[-(1 - y) \times \log(1 - \hat{y})]'_{\hat{y}} = \frac{1 - y}{1 - \hat{y}} \qquad (14)$$

Bước 2: Tính $\frac{\partial \hat{y}}{\partial z}$. Ta đặt:

$$\hat{y} = sigmoid(z) = \frac{1}{1 + e^{-z}} \qquad (15)$$

Trong đó: $z = W^T X + b$

Chúng ta có đạo hàm:

$$\frac{\partial \hat{y}}{\partial x} = [\frac{1}{1 + e^{-z}}]' z = \hat{y} \times (1 - \hat{y}) \qquad (16)$$

Tổng hợp cả hai phần đạo hàm trên ta có:

$$\frac{-y}{\hat{y}} + \frac{1-y}{1-\hat{y}} \to \frac{-y}{\hat{y}} \times \left(\frac{1}{\hat{y}} - 1\right) + \frac{1-y}{1-\hat{y}} \to \frac{-y \times \frac{1}{\hat{y}} + y + 1 - y}{1-\hat{y}} \times \hat{y}$$
$$\to \frac{\hat{y} - y}{\hat{y} \times (1-\hat{y})} \qquad (17)$$

Bước 3: Tính $\frac{\partial z}{\partial W}$. Ta có:

$$z = W^T X + b = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

Vậy nên ta có đạo hàm:

$$\frac{\partial z}{\partial W} = x_i \qquad (18)$$

Tổng hợp đạo hàm của các hàm hợp (16),(17),(18), ta có:

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial \hat{z}} \times \frac{\partial z}{\partial \widehat{W}} = \frac{\hat{y} - y}{\hat{y} \times (1-\hat{y})} \times \hat{y} \times (1-\hat{y}) \times x_i = x_i \times (\hat{y} - y) \qquad (19)$$

5: PHÂN TÍCH THUẬT TOÁN

1. **Đầu vào:** $X, y, \eta, num\_iterations$: Nhận đầu vào là ma trận dữ liệu X, vector nhãn y, tốc độ học η, và số lần lặp $num\_iterations$ để cập nhật trọng số mô hình.
2. **Đầu ra:** w, b Trả về các trọng số w và độ dời b của mô hình logistic regression sau khi đã được huấn luyện.
3. **Khởi tạo** w, b Bắt đầu bằng việc khởi tạo các trọng số w và độ dời b của mô hình.
4. **repeat for num_iterations**: Lặp lại quá trình huấn luyện cho số lần đã xác định.
5. **Tính toán** $\hat{y} = sigmoid(w^T.X + b)$. Tính toán giá trị dự đoán sử dụng hàm sigmoid của tổng trọng số w nhân với ma trận dữ liệu X cộng với độ dời b.
6. **Tính toán** $dw = \frac{1}{m}.X.(\hat{y} - y)^T$. Tính toán gradient của hàm mất mát theo trọng số w.
7. **Tính toán** $db = \frac{1}{m}.\sum(\hat{y} - y)$. Tính toán gradient của hàm mất mát theo độ dời b.
8. **Cập nhật w**: $w \leftarrow w - \eta.dw$. Cập nhật trọng số w bằng cách di chuyển nó ngược lại hướng của gradient với tốc độ học η.
9. **Cập nhật b**: $b \leftarrow b - \eta * db$. Cập nhật độ dời b tương tự như trên.
10. **until convergence**: Lặp lại quá trình cập nhật trọng số và độ dời cho đến khi đạt được sự hội tụ.
11. **return w, b**: Trả về các trọng số w và độ dời b cuối cùng của mô hình logistic regression.

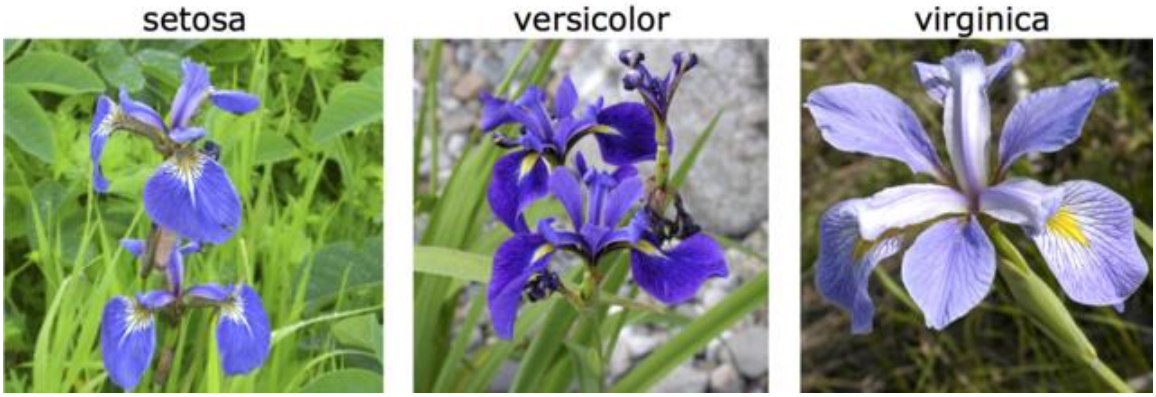
6: ĐỀ XUẤT ỨNG DỤNG

6.1 Ứng dụng đơn biến

Dự đoán khách hàng ở độ tuổi nào thì thường sẽ mua bảo hiểm nhân thọ.

6.2 Ứng dụng đa biến

Phân loại hoa Iris.



Hình 7: Hoa Iris

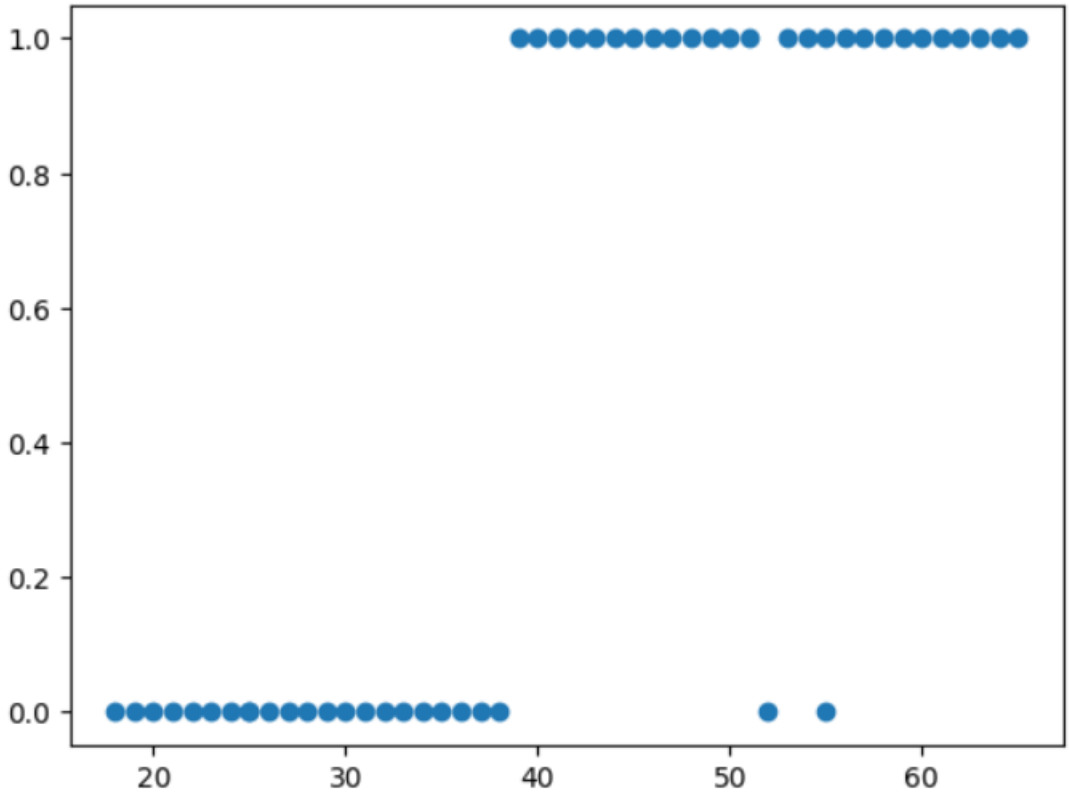
6.3 Tập dữ liệu

age	bought_insurance
22	0
25	0
47	1
52	0
46	1
56	1
55	0
60	1
62	1
61	1
20	0
28	0
27	0
29	0
49	1
55	1
25	0
58	1
19	0
18	0
21	0
26	0
40	1
45	1
50	1
54	1
23	0
24	0
30	0

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3	1.4	0.1	Iris-setosa
14	4.3	3	1.1	0.1	Iris-setosa
15	5.8	4	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
18	5.1	3.5	1.4	0.3	Iris-setosa
19	5.7	3.8	1.7	0.3	Iris-setosa
20	5.1	3.8	1.5	0.3	Iris-setosa
21	5.4	3.4	1.7	0.2	Iris-setosa
22	5.1	3.7	1.5	0.4	Iris-setosa
23	4.6	3.6	1	0.2	Iris-setosa
24	5.1	3.3	1.7	0.5	Iris-setosa
25	4.8	3.4	1.9	0.2	Iris-setosa
26	5	3	1.6	0.2	Iris-setosa
27	5	3.4	1.6	0.4	Iris-setosa
28	5.2	3.5	1.5	0.2	Iris-setosa

7: MÔ PHỎNG VÀ SO SÁNH VỚI THƯ VIỆN SKLEARN

7.1 Đơn biến



Hình 8: Biểu đồ scatter cho tập dữ liệu đơn biến

7.1.1 Code sử dụng thư viện Sklearn

Đánh giá mô hình với score:

```
0.95
```

Hình 9: Độ chính xác khi sử dụng Thư viện Sklearn

Kết quả cho thấy, độ chính xác của mô hình lên đến 95%.

	precision	recall	f1-score	support
0	1.00	0.75	0.86	4
1	0.86	1.00	0.92	6
accuracy			0.90	10
macro avg	0.93	0.88	0.89	10
weighted avg	0.91	0.90	0.90	10

Hình 10: Đánh giá mô hình khi sử dụng Thư viện Sklearn

Nhìn vào đánh giá mô hình, ta có thể nói rằng:

- Với class 0 (không mua), 100% các mẫu dự đoán là lớp 0 đều đúng, 75% các mẫu thực sự là lớp 0 được dự đoán đúng.
- Với class 1 (mua), 86% các mẫu dự đoán là lớp 1 đều đúng, 100% các mẫu thực sự là lớp 1 được dự đoán đúng.

7.1.2 Code không sử dụng thư viện Sklearn

Độ chính xác:

```
Độ chính xác của mô hình: 92.00%
```

Hình 11: Độ chính xác khi không sử dụng Thư viện Sklearn

Kết quả cho thấy, độ chính xác của mô hình lên đến 92%.

	precision	recall	f1-score	support
0	0.92	0.92	0.92	24
1	0.92	0.92	0.92	26
accuracy			0.92	50
macro avg	0.92	0.92	0.92	50
weighted avg	0.92	0.92	0.92	50

Hình 12: Đánh giá mô hình khi không sử dụng Thư viện Sklearn

Nhìn vào đánh giá mô hình, ta có thể thấy rằng:

- Với class 0 (Iris-setosa), 92% các mẫu dự đoán là lớp 0 đều đúng, 92% các mẫu thực sự là lớp 0 được dự đoán đúng.
- Với class 1 (Iris-versicolor), 92% các mẫu dự đoán là lớp 1 đều đúng, 92% các mẫu thực sự là lớp 1 được dự đoán đúng.

7.1.3 Dự đoán

Chúng ta sẽ sử dụng 2 mô hình để dự đoán nếu khách hàng ở độ tuổi là 50 tuôi thì kết quả sẽ là mua hay không mua bảo hiểm nhân thọ:

kết quả dự đoán của mô hình sử dụng sklearn:

Hình 13: Dự đoán của mô hình sử dụng thư viện Sklearn

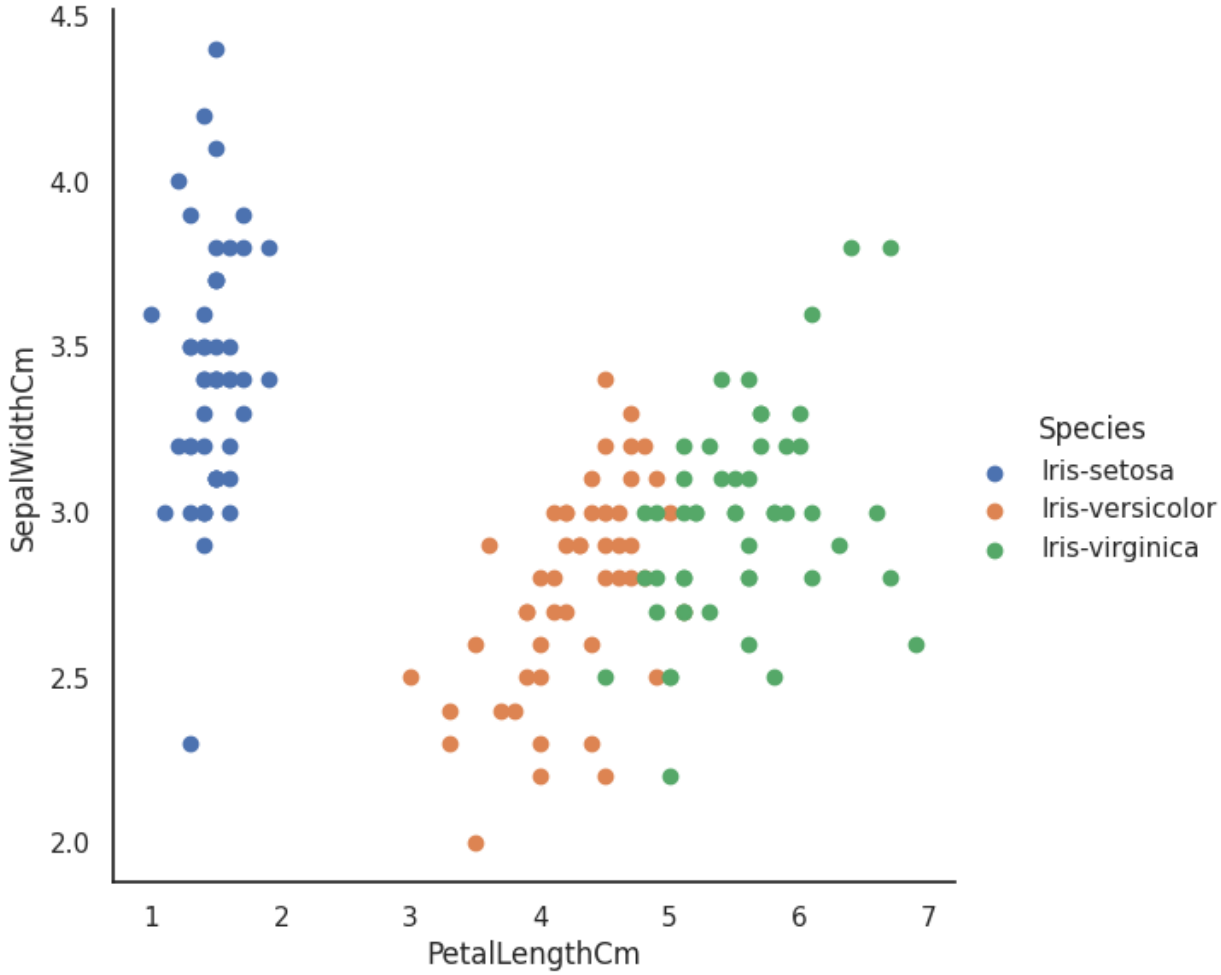
Kết quả dụ đoán của mô hình không sử dụng sklearn:

Hình 14: Dự đoán của mô hình khi không sử dụng thư viện Sklearn

Kết luận: Với kết quả dự đoán này của 2 mô hình thì cả 2 mô hình đều dự đoán rằng với khách hàng ở độ tuổi là 50 tuổi thì sẽ mua bảo hiểm nhân thọ.

Code

7.2 Đa biến



Hình 15: Biểu đồ scatter của tập dữ liệu đa biến

7.2.1 Code với thư viện Sklearn

```
0.9733333333333334
```

Hình 16: Độ chính xác khi sử dụng thư viện Sklearn

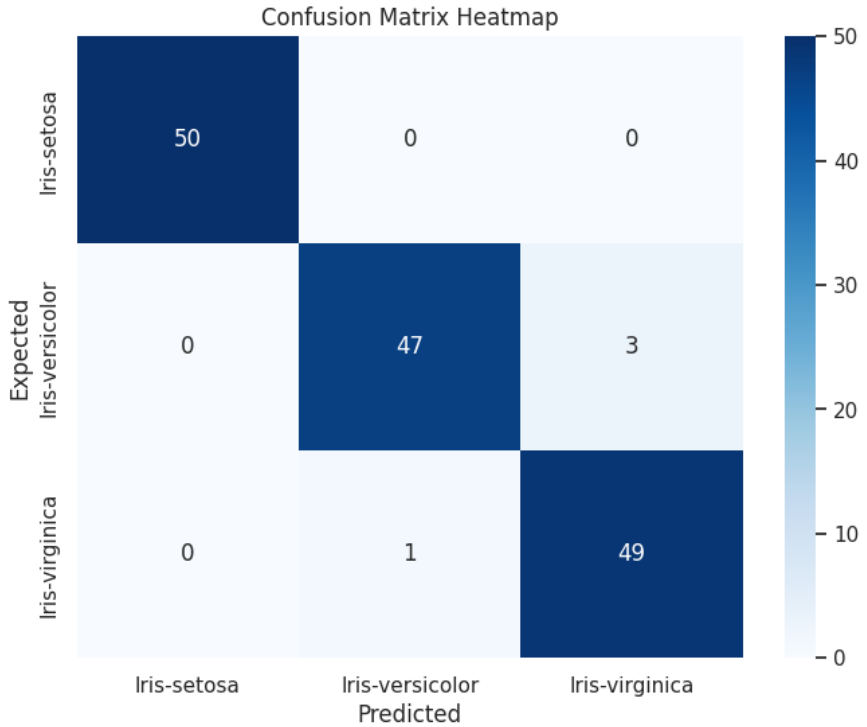
Kết quả cho thấy, độ chính xác của mô hình lên đến 97%.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	50
1	0.98	0.94	0.96	50
2	0.94	0.98	0.96	50
accuracy			0.97	150
macro avg	0.97	0.97	0.97	150
weighted avg	0.97	0.97	0.97	150

Hình 17: Đánh giá mô hình khi sử dụng thư viện Sklearn

Nhìn vào đánh giá mô hình, ta có thể nói rằng:

- Với class 0 (Iris-setosa), 100% các mẫu dự đoán là lớp 0 đều đúng, 100% các mẫu thực sự là lớp 0 được dự đoán đúng.
- Với class 1 (Iris-versicolor), 98% các mẫu dự đoán là lớp 1 đều đúng, 94% các mẫu thực sự là lớp 1 được dự đoán đúng.
- Với class 2 (Iris-virginica), 94% các mẫu dự đoán là lớp 2 đều đúng, 98% các mẫu thực sự là lớp 2 được dự đoán đúng.



Hình 18: Ma trận nhầm lẫn khi sử dụng thư viện Sklearn

Nhìn vào ma trận nhầm lẫn cho thấy rằng:

- Với class 0 (Iris-setosa) 50 số mẫu Iris-setosa được dự đoán đúng là Iris-setosa. 0 số mẫu Iris-setosa bị dự đoán nhầm là Iris-versicolor. 0 Số mẫu Iris-setosa bị dự đoán nhầm là Iris-virginica.
- Với class 1 (Iris-versicolor) 0: Số mẫu Iris-versicolor bị dự đoán nhầm là Iris-setosa. 47: Số mẫu Iris-versicolor được dự đoán đúng là Iris-versicolor. 3: Số mẫu Iris-versicolor bị dự đoán nhầm là Iris-virginica.
- Với class 2 (Iris-virginica) 0: Số mẫu Iris-virginica bị dự đoán nhầm là Iris-setosa. 1: Số mẫu Iris-virginica bị dự đoán nhầm là Iris-versicolor. 49: Số mẫu Iris-virginica được dự đoán đúng là Iris-virginica.

7.2.2 Code không sử dụng thư viện Sklearn

```
dtype: object
Model Accuracy (scratch): 96.67%
```

Hình 19: Độ chính xác khi không sử dụng thư viện Sklearn

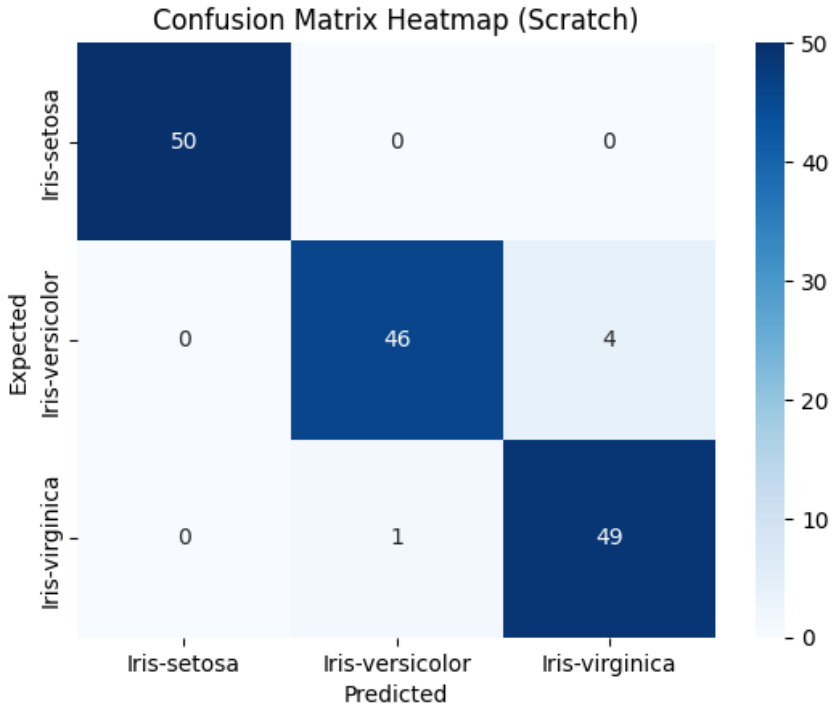
Kết quả cho thấy, độ chính xác của mô hình là 96,67%. Thấp hơn độ chính xác của mô hình khi sử dụng thư viện Sklearn.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	50
1	0.98	0.92	0.95	50
2	0.92	0.98	0.95	50
accuracy			0.97	150
macro avg	0.97	0.97	0.97	150
weighted avg	0.97	0.97	0.97	150

Hình 20: Đánh giá mô hình khi không sử dụng thư viện Sklearn

Nhìn vào đánh giá mô hình, ta có thể nói rằng::

- Với class 0 (Iris-setosa), 100% các mẫu dự đoán là lớp 0 đều đúng, 100% các mẫu thực sự là lớp 0 được dự đoán đúng.
- Với class 1 (Iris-versicolor), 98% các mẫu dự đoán là lớp 1 đều đúng, 92% các mẫu thực sự là lớp 1 được dự đoán đúng.
- Với class 2 (Iris-virginica), 92% các mẫu dự đoán là lớp 2 đều đúng, 98% các mẫu thực sự là lớp 2 được dự đoán đúng.



Hình 21: Ma trận nhầm lẫn khi không sử dụng thư viện Sklearn

Nhìn vào ma trận nhầm lẫn cho thấy rằng:

- Với class 0 (Iris-setosa) 50 số mẫu Iris-setosa được dự đoán đúng là Iris-setosa. 0 số mẫu Iris-setosa bị dự đoán nhầm là Iris-versicolor. 0 Số mẫu Iris-setosa bị dự đoán nhầm là Iris-virginica.
- Với class 1 (Iris-versicolor) 0: Số mẫu Iris-versicolor bị dự đoán nhầm là Iris-setosa. 46: Số mẫu Iris-versicolor được dự đoán đúng là Iris-versicolor. 4: Số mẫu Iris-versicolor bị dự đoán nhầm là Iris-virginica.
- Với class 2 (Iris-virginica) 0: Số mẫu Iris-virginica bị dự đoán nhầm là Iris-setosa. 1: Số mẫu Iris-virginica bị dự đoán nhầm là Iris-versicolor. 49: Số mẫu Iris-virginica được dự đoán đúng là Iris-virginica.

Có thể thấy rằng, cả hai phương pháp đều cho kết quả phân loại tương đối tốt trên tập dữ liệu, với độ chính xác cao.

Sử dụng thư viện Sklearn có vẻ có hiệu suất phân loại tốt hơn so với phương pháp không sử dụng thư viện Sklearn, đặc biệt là đối với class 1 (Iris- versicolor).

Khi phương pháp sử dụng thư viện Sklearn có 3 mẫu bị phân loại sai, thì phương pháp không sử dụng Sklearn có 4 mẫu bị phân loại sai.

Code

TÀI LIỆU THAM KHẢO

https://www.geeksforgeeks.org/understanding-logistic-regression/
https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/
https://cafedev.vn/tu-hoc-ml-hieu-hoi-quy-logistic/
https://chat.openai.com/

KẾT QUẢ KIỂM TRA ĐẠO VĂN