**Phần 1\_6\_Độ đo (metrics) và cách đánh giá hiệu quả của bài toán**

**Notes:**

* *Về format cấu trúc soạn như sau:*
  + *Lý thuyết…*
  + *Bộ code mấu/ ví dụ …*
  + *Ứng dụng (nếu có)...*
* *Mems làm nhớ note tên để mn dễ contact*

**Mục lục**

[**I. Nội dung chính 1**](#_qx01752rlwy1)

[**II. Nội dung biên soạn chi tiết 1**](#_qoebufjbotbu)

[**1. Với bài toán hồi quy: MSE 1**](#_h3b6difsjw45)

[a. Lý thuyết 1](#_2c4kxtk83wae)

[b. Code ví dụ (sử dụng Python và thư viện của Python) 2](#_u0243m9sieut)

[c. Ứng dụng thực tiễn 2](#_svcvo5by07sm)

[**2. Với bài toán phân loại: confusion matrix 3**](#_md0w1oqilbzu)

[a. Lý thuyết 3](#_fvfpfbh70jrb)

[b. Code ví dụ (sử dụng Python và thư viện của Python) 3](#_r8wbfaqldvos)

[c. Ứng dụng thực tiễn 3](#_lqzsakwl4ub3)

[**3. Với bài toán phân đoạn: dice-score 4**](#_qww4ocpyics5)

[a. Lý thuyết 4](#_6heslrkwi754)

[b. Code ví dụ (sử dụng Python và thư viện của Python) 4](#_28ibozmepf3y)

[c. Ứng dụng thực tiễn 5](#_7elvofyyset5)

### **I. Nội dung chính**

Về dạng bài toán hay công việc (tasks): **phân loại**, **hồi quy**, **phát hiện**, **phân** **đoạn**, **sinh ảnh**. Cho mỗi dạng bài toán, cần hiểu rõ:

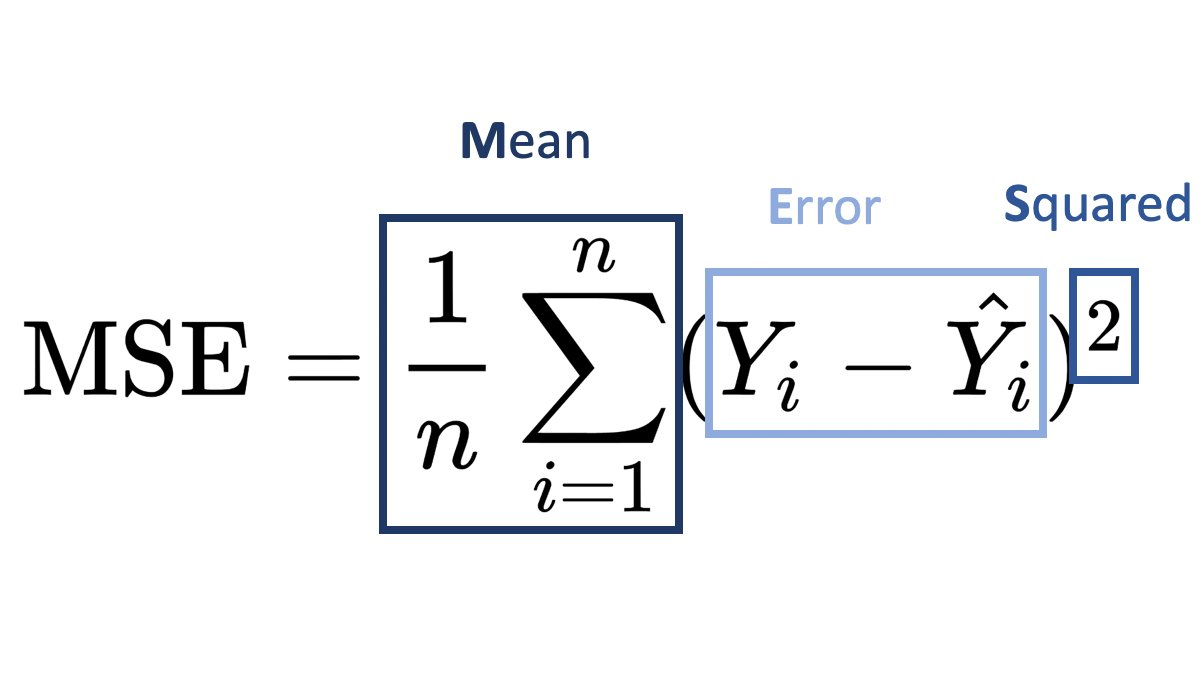
* (6) Độ đo (metrics) và cách đánh giá hiệu quả của bài toán
  + Ví dụ: với hồi quy **MSE**, **RMSE**, **MAE**, etc
  + Ví dụ: với phân loại: **confusion matrix**, **precision/recall/f1/accuracy và weighted/macro-average**
  + Ví dụ: với phân đoạn: **Dice-Score, IoU-Score**
  + Ví dụ: với phát hiện: **mAP**

### **II. Nội dung biên soạn chi tiết**

#### **1.** **Với bài toán hồi quy: MSE**

##### **a. Lý thuyết**

Khái niệm: MSE (Mean Squared Error) là một độ đo đánh giá sai số giữa giá trị dự đoán và giá trị thực tế trong học máy và thống kê. Nó đo lường trung bình của bình phương của sai số, tức là sự chênh lệch giữa giá trị dự đoán và giá trị thực tế, nhằm đánh giá hiệu suất của mô hình.

Cơ sở lý thuyết: Công thức MSE được xây dựng dựa trên việc tính tổng bình phương của các sai số rồi chia cho số lượng mẫu:

Trong đó:

● n là số lượng mẫu trong tập dữ liệu.

● Yi là giá trị dự đoán của mẫu thứ i.

● Ŷi là giá trị thực tế của mẫu thứ i.

Ứng dụng trong học máy và trí tuệ nhân tạo: MSE được sử dụng rộng rãi để đánh giá hiệu suất của mô hình trong các bài toán hồi quy. Khi huấn luyện một mô hình hồi quy, việc tối thiểu hóa MSE giữa giá trị dự đoán và giá trị thực tế thông qua việc điều chỉnh các tham số của mô hình là mục tiêu chính.

##### **b.** **Code ví dụ (sử dụng Python và thư viện của Python)**

import numpy as np

from sklearn.metrics import mean\_squared\_error

# Giá trị thực tế

actual\_values = np.array([3, 5, 7, 9, 11])

# Giá trị dự đoán

predicted\_values = np.array([2.5, 5.5, 6.8, 8.5, 10.5])

# Tính toán MSE bằng hàm mean\_squared\_error từ thư viện scikit-learn

mse = mean\_squared\_error(actual\_values, predicted\_values)

print(f"Mean Squared Error là: {mse}")

##### **c.** **Ứng dụng thực tiễn**

● Trong tài chính: Trong lĩnh vực tài chính, MSE được sử dụng để đo lường độ chính xác của các mô hình dự đoán giá cổ phiếu, giá trị tài sản, hoặc trong việc dự báo xu hướng thị trường.

● Trong y học: MSE cũng được áp dụng trong y học để đo lường độ chính xác của các mô hình dự đoán, ví dụ như dự đoán biến cố y khoa, hoặc đo lường hiệu suất của các phương pháp chẩn đoán.

#### **2.** **Với bài toán phân loại: confusion matrix**

##### **a. Lý thuyết**

Khái niệm: Confusion matrix (ma trận nhầm lẫn) là một công cụ quan trọng để đánh giá hiệu suất của một mô hình phân loại. Nó cung cấp một bức tranh chi tiết về cách mà mô hình phân loại các nhãn hoặc lớp dữ liệu.

Cấu trúc: Confusion matrix thường được biểu diễn dưới dạng bảng, trong đó hàng và cột tương ứng với nhãn thực tế và nhãn dự đoán. Các phần tử trong ma trận biểu thị số lượng mẫu thuộc từng nhóm.

Các thành phần chính:

● True Positives (TP): Số lượng mẫu thực tế thuộc lớp dương và được dự đoán đúng.

● True Negatives (TN): Số lượng mẫu thực tế thuộc lớp âm và được dự đoán đúng.

● False Positives (FP): Số lượng mẫu thực tế thuộc lớp âm nhưng được dự đoán là lớp dương (lỗi loại 1).

● False Negatives (FN): Số lượng mẫu thực tế thuộc lớp dương nhưng được dự đoán là lớp âm (lỗi loại 2).

Ứng dụng trong học máy và trí tuệ nhân tạo: confusion matrix cung cấp cái nhìn chi tiết về hiệu suất của mô hình phân loại bằng cách đo lường số lượng True Positive, True Negative, False Positive và False Negative. Thông qua confusion matrix, chúng ta có thể đánh giá hiệu suất của mô hình, điều chỉnh ngưỡng quyết định, tìm hiểu về loại lỗi mà mô hình gặp phải, tối ưu hóa mô hình phân loại và sử dụng nó trong việc đánh giá mô hình thị giác máy tính. Điều này giúp cải thiện hiệu suất và đáng tin cậy của mô hình trong việc phân loại dữ liệu.

##### **b.** **Code ví dụ (sử dụng Python và thư viện của Python)**

from sklearn.metrics import confusion\_matrix

# Giả sử y\_true là nhãn thực tế, y\_pred là nhãn dự đoán

conf\_matrix = confusion\_matrix(y\_true, y\_pred)

print(conf\_matrix)

##### **c.** **Ứng dụng thực tiễn**

Trong tài chính:

Ví dụ, trong việc dự đoán một bệnh lý dựa trên các chỉ số y khoa, mô hình phân loại có thể được sử dụng để dự đoán xác suất một bệnh nhân mắc phải bệnh hay không. Confusion matrix có thể cung cấp thông tin về số lượng các trường hợp mà mô hình dự đoán đúng bệnh nhân mắc bệnh (True Positives), số lượng bệnh nhân không mắc bệnh và được dự đoán đúng (True Negatives), số lượng bệnh nhân không mắc bệnh nhưng được dự đoán mắc bệnh (False Positives), và số lượng bệnh nhân mắc bệnh nhưng bị dự đoán không mắc bệnh (False Negatives). Thông qua confusion matrix, chúng ta có thể tính được các chỉ số như accuracy, precision, recall và F1-score để đánh giá hiệu suất của mô hình. Điều này có thể giúp trong việc hiểu rõ hơn về khả năng dự đoán của mô hình, xác định loại lỗi mà mô hình gặp phải (ví dụ như dự đoán thiên lệch về một nhóm bệnh nhân nào đó), và điều chỉnh mô hình để cải thiện hiệu suất trong việc dự đoán bệnh lý.

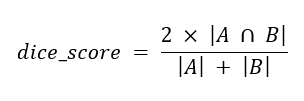
#### **3.** **Với bài toán phân đoạn: dice-score**

##### **a. Lý thuyết**

Khái niệm: Dice score, còn được gọi là Dice coefficient, là một độ đo phổ biến trong bài toán phân đoạn hình ảnh (segmentation). Nó được sử dụng để đánh giá độ chính xác của việc phân đoạn vùng quan tâm trên hình ảnh so với ground truth, tức là kết quả phân đoạn được dự đoán so với kết quả thực tế.

Cơ sở lý thuyết: Công thức Dice score:

Dice score được tính dựa trên diện tích của vùng phân đoạn dự đoán và diện tích của vùng thực tế:



Trong đó:

● |A| và |B| biểu thị số lượng phần tử của tập A và B

● |A ∩ B|: đại diện cho số phần tử có trong cả hai bộ.

Điều này có thể được hiểu như là đo lường sự đồng nhất giữa vùng phân đoạn và vùng thực tế. Giá trị của Dice score dao động từ 0 đến 1, với 1 đại diện cho việc phân đoạn hoàn toàn chính xác và 0 đại diện cho việc không có sự trùng khớp nào giữa phân đoạn dự đoán và ground truth.

##### **b.** **Code ví dụ (sử dụng Python và thư viện của Python)**

import numpy as np

def dice\_score(prediction, ground\_truth):

intersection = np.sum(prediction \* ground\_truth)

dice = (2.0 \* intersection) / (np.sum(prediction) + np.sum(ground\_truth))

return dice

# Giả sử có hai ma trận phân đoạn dự đoán và ground truth (0 và 1)

# Đây chỉ là một ví dụ giả định, thực tế ta cần tính từ hình ảnh hoặc dữ liệu thực tế

prediction = np.array([[0, 1, 1],

[1, 1, 0],

[0, 0, 1]])

ground\_truth = np.array([[0, 1, 0],

[1, 1, 0],

[0, 1, 1]])

# Tính toán Dice score

score = dice\_score(prediction, ground\_truth)

print(f"Dice score: {score}")

##### **c.** **Ứng dụng thực tiễn**

● Trong lĩnh vực y học: Trong y học hình ảnh, độ đo Dice score được sử dụng rộng rãi để đánh giá chất lượng của việc phân đoạn các cấu trúc trong hình ảnh y khoa như gan, tế bào ung thư, hoặc các cấu trúc sinh học khác. Nó cũng được áp dụng trong phân đoạn mạch máu, phân loại các bộ phận cơ thể trong y học hình ảnh và nhiều ứng dụng khác.

● Trong xử lý ảnh và computer vision: Độ đo Dice score cũng được sử dụng để đánh giá hiệu suất của các thuật toán phân đoạn hình ảnh không chỉ trong lĩnh vực y học mà còn trong xử lý ảnh, nhận diện vật thể, và các ứng dụng trong computer vision khác.