CHƯƠNG 3:
BÀI TOÁN PHÂN LỚP (CLASSIFICATION)

Khoa Khoa học và Kỹ thuật thông tin
Bộ môn Khoa học dữ liệu

NỘI DUNG

1. Định nghĩa bài toán phân lớp.
2. Bài toán phân lớp nhị phân.
3. Các độ đo đánh giá dùng cho bài toán phân lớp.
4. Phân tích lỗi.
5. Các dạng bài toán phân lớp khác

ĐỊNH NGHĨA BÀI TOÁN PHÂN LỚP

Định nghĩa bài toán và tác vụ phân lớp

– Tác vụ phân lớp sẽ gán cho dữ liệu đầu vào một nhãn (label) cụ thể trong tập hữu hạn các nhãn của bộ dữ liệu.

– Input: dữ liệu đầu vào.

– Output: nhãn c của dữ liệu thuộc tập nhãn C.

Ví dụ:

- Phân tích cảm xúc.
- Phân loại chủ đề cho văn bản.
- Phân loại ảnh món ăn.
- Phân loại ảnh X-Quang phổi.

Phân biệt giữa phân lớp và hồi quy

Phân lớp (classification)

Nhãn của dữ liệu đầu vào thuộc một nhóm hữu hạn (tập các nhãn).

VD: Bài toán dự đoán tin nhắn spam.

Nhãn của bài toán là: “spam” và “not-spam”.

Hồi quy (regression)

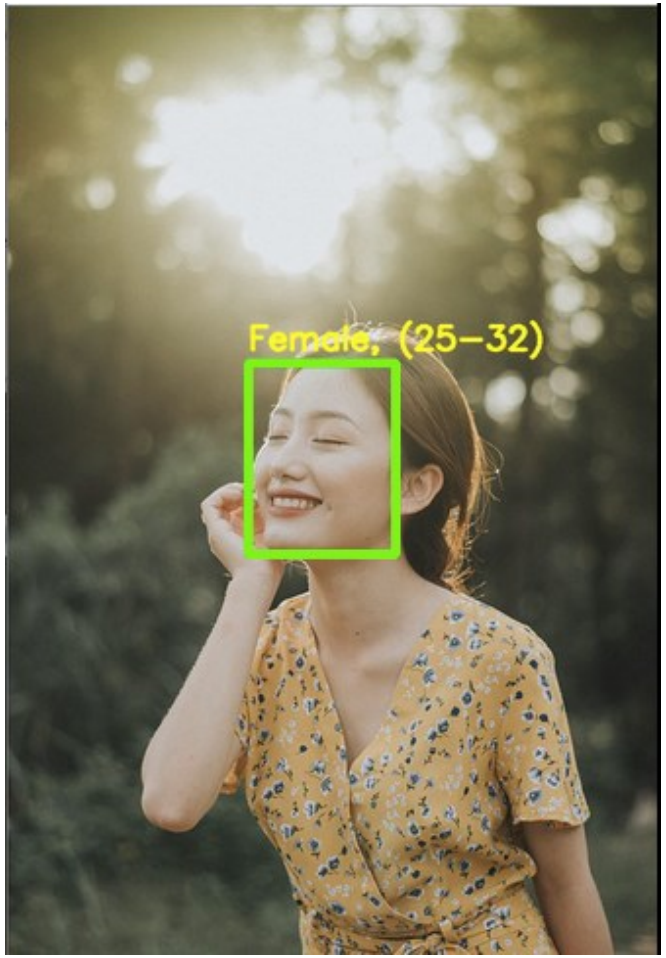
Nhãn của dữ liệu đầu vào là một giá trị thực.

VD: Bài toán dự đoán giá nhà.

Nhãn của bài toán là: một giá trị số cụ thể chỉ giá nhà.

Tác vụ phân lớp và tác vụ hồi quy thuộc về học máy có giám sát (Supervised learning)

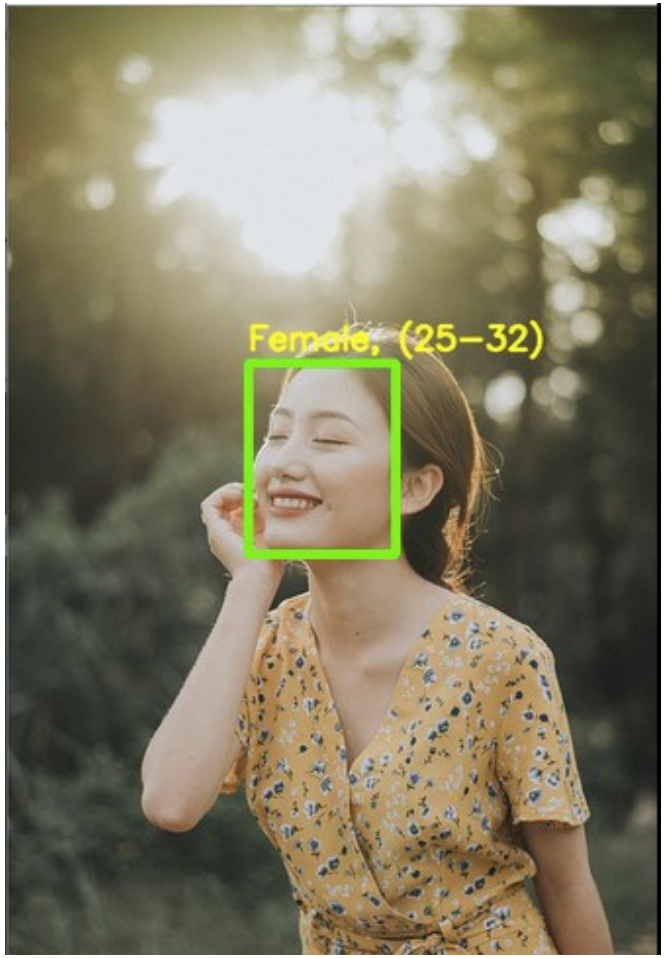
Ví dụ



Ứng dụng nhận diện giới tính và tuổi.

➔ Hãy liệt kê thử xem bài toán này có bao nhiêu tác vụ?

Ví dụ



Bài toán có 3 tác vụ:

- Tác vụ 1: Nhận diện khuôn mặt ➔ Phân lớp.
- Tác vụ 2: Nhận diện giới tính ➔ Phân lớp.
- Tác vụ 3: Nhận diện tuổi ➔ Hồi quy.

Các dạng bài toán phân lớp

Pick one	Pick one	Pick all applicable
Label 1 ✓	Label 1	Label 1
Label 2	Label 2	Label 2 ✓
	Label 3	Label 3
	Label 4 ✓	Label 4 ✓
	...	...
	...	...
	Label L	Label L ✓

Binary	Multi-class	Multi-label
Phân lớp nhị phân	Phân lớp đa lớp	Phân lớp đa nhãn

Một số ví dụ về bài toán phân lớp

– Phân lớp nhị phân

Một số ví dụ về bài toán phân lớp

– Phân lớp đa lớp

Multi-Class Classification

Cat
Dog
Fox
Tiger
Lion

0 5
1 6
2 7
3 8
4 9

Person A
Person B
Person C
.
.
.

Một số ví dụ về bài toán phân lớp

– Phân lớp đa nhãn

Multi-Label Classification

Java
C++
Python
C#
Go

Action
Crime
Thriller
Comedy
Drama

Cat
Dog
Tiger
Bird
Fish

BÀI TOÁN PHÂN LỚP NHỊ PHÂN

Dẫn nhập

– Bài toán đặt ra: cần xây dựng một hệ thống trợ giúp bác sĩ đọc ảnh X-quang phổi của bệnh nhân, từ đó trợ giúp bác sĩ chẩn đoán xem bệnh nhân có bệnh phổi hay không?

– Mô hình hoá bài toán:

+ Input: Bức ảnh X-quang của bệnh nhân.

+ Output: nhãn của bức ảnh, gồm 1 trong 2 nhãn sau:

- ***NORMAL: phổi bình thường.***
- ***PNEUNOMIA: phổi bị bệnh.***

➔ Bài toán phân lớp nhị phân (dự đoán 2 lớp).

Định nghĩa tác vụ máy học

- Tác vụ (T): Dự đoán xem một người có bị bệnh về phổi hay không thông qua ảnh X – quang phổi (Chest X – ray).
- Kinh nghiệm (E): dữ liệu về **các bức ảnh X – Quang về phổi đã được gán nhãn trước** để biết đâu là phổi bị bệnh và đâu là phổi bình thường.
- Độ đo đánh giá (P): Accuracy, Precision, Recall, F1-score,…

Bộ dữ liệu

Bộ dữ liệu: **Chest X-Ray Images**

– Gồm 5,863 ảnh X-Quang phổi, được gán nhãn sẵn bởi con người,

– Mỗi ảnh được gán 1 trong 2 nhãn là NORMAL hoặc PNEUNOMIA.

NORMAL

PNEUMONIA

https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

Cấu trúc bộ dữ liệu

— train:

+ NORMAL: *các file .jpeg*

+ PNEUMONIA: *các file .jpeg*

— dev:

+ NORMAL: *các file .jpeg*

+ PNEUMONIA: *các file .jpeg*

— test:

+ NORMAL: *các file .jpeg*

+ PNEUMONIA: *các file .jpeg*

NORMAL

PNEUMONIA

NORMAL

PNEUMONIA

NORMAL

PNEUMONIA

Ý nghĩa các tập dữ liệu train, dev và test

- Tập train (gọi là tập huấn luyện): Dùng để huấn luyện mô hình máy học.
- Tập dev (gọi là tập phát triển): Dùng để tinh chỉnh tham số cho mô hình.
- Tập test (gọi là tập kiểm tra): Dùng để kiểm tra độ chính xác cho mô hình.

Lưu ý: Tập test chỉ dùng để kiểm tra độ chính xác cuối cùng, không được dùng để điều chỉnh mô hình.

Các bước thực hiện

Load dữ liệu → Mã hoá dữ liệu → Huấn luyện mô hình → Dự đoán cho dữ liệu mới → Đánh giá mô hình

Đọc dữ liệu ảnh

– Đọc dữ liệu: Sử dụng thư viện OpenCV

```
1. import cv2
2. IMG_SIZE = 227
3. img = cv2.imread(”
      drive/MyDrive/ML/chest_xray/train/PNEUMONIA/person1_bacteria_1.jpeg
   ")
4. img = cv2.resize(img, (IMG_SIZE, IMG_SIZE))
```

– Kết quả:

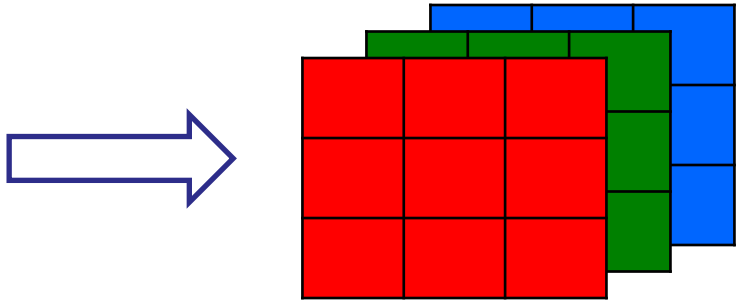
Nhãn là: NORMAL

Đưa ảnh về dạng ma trận

— Thư viện sử dụng: numpy

```
1. import numpy as np
2. img = np.asarray(img)
```

227x227



227x227x3

3 kênh màu ứng với giá trị của 3 màu cơ bản: RGB

Tập dữ liệu có N ảnh:

[

ảnh thứ 1,
ảnh thứ 2,
ảnh thứ 3,
....
ảnh thứ n

]

➔ Số chiều của tập dữ liệu là:

(N x 227 x 227 x 3)

Số ảnh — Kích thước mỗi ảnh — Số kênh màu mỗi ảnh

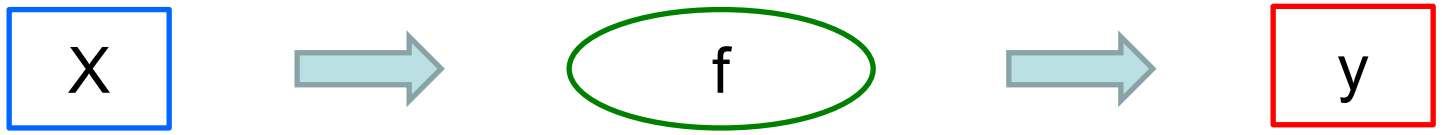
Load bộ dữ liệu

- Đọc từng thư mục train, dev và test.
- Đọc từng file ảnh trong từng thư mục ứng với tên lớp (NORMAL hoặc PNEUMONIA): Sử dụng thư viện glob trong python.
- Đọc từng file ảnh, sau đó chuyển ảnh sang dạng ma trận: Sử dụng thư viện PIL image và numpy.

Hàm đọc dữ liệu đầy đủ

```
import glob
import numpy as np
import cv2
IMG_SIZE = 227
def load_dataset(path):
    X = np.array([])
    y = np.array([])
    classes = ['NORMAL', 'PNEUMONIA’]
    for c in classes:
        files = glob.glob(path + c + "/*.jpeg")
        for f in files:
            img = cv2.imread(f)
            img = cv2.resize(img, (IMG_SIZE, IMG_SIZE))
            if X.size == 0: X = np.array([img])
            else: X = np.vstack([X,[img]])
            y = np.append(y, c)
    return (X,y)
```

Huấn luyện mô hình



- Vấn đề đặt ra: đi tìm f dựa vào X và y
  + X và y ở đây chính là dữ liệu huấn luyện.
- Để tìm được f: dựa vào dữ liệu huấn luyện ➔ huấn luyện mô hình.
- Mục tiêu: dự đoán dữ liệu X' mới chưa biết nhãn trong tương lai.
- Vấn đề: làm sao biết được f tốt ➔ đánh giá mô hình.
- Cơ sở đánh giá: các độ đo đánh giá (evaluation metric).

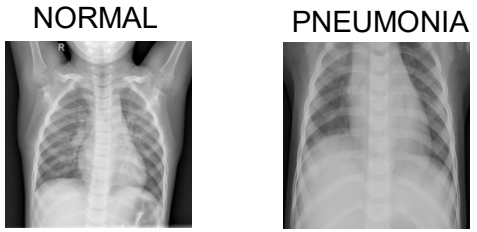
Mã hoá dữ liệu

```
1. from sklearn.preprocessing import LabelEncoder
2. le = LabelEncoder()
3. le.fit(y_train)
4. X1 = X_train.reshape(X_train.shape[0],
   IMG_SIZE*IMG_SIZE*3)
5. X2 = X_dev.reshape(X_dev.shape[0],
   IMG_SIZE*IMG_SIZE*3)
6. X3 = X_test.reshape(X_test.shape[0],
   IMG_SIZE*IMG_SIZE*3)

7. y1 = le.transform(y_train)
8. y2 = le.transform(y_dev)
9. y3 = le.transform(y_test)
```

y_train

X_train



....

Tập dữ liệu huấn luyện

Label encoder là một thư viện trong sklearn sẽ giúp mã hoá nhãn (label) về dạng số (numeric)
VD:

- "*normal*" sẽ đưa về 0.
- "*pneumonia*" sẽ đưa về 1.

reshape là một kỹ thuật trong thư viện numpy sẽ giúp biến đổi chiều của ma trận.
VD: một ảnh ban đầu: (1, 227, 227, 3) là ma trận **4 chiều**, sau khi biến đổi sẽ thành ma trận **2 chiều** có kích thước là (1, 227*227*3) = (1, 154587)

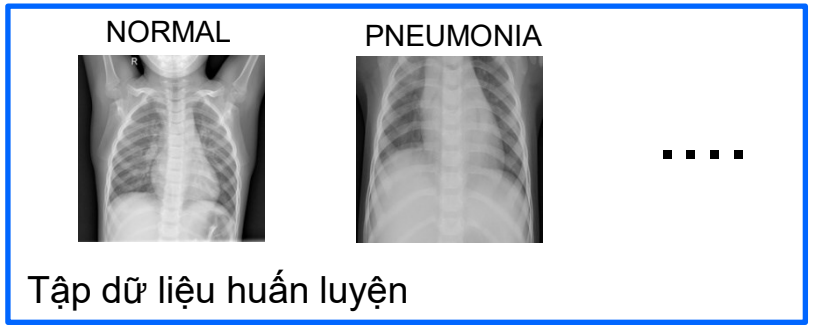
Mã hoá dữ liệu

```
1.  from sklearn.preprocessing import LabelEncoder
2.  le = LabelEncoder()
3.  le.fit(y_train)
4.  X1 = X_train.reshape(X_train.shape[0],
    IMG_SIZE*IMG_SIZE*3)
5.  X2 = X_dev.reshape(X_dev.shape[0],
    IMG_SIZE*IMG_SIZE*3)
6.  X3 = X_test.reshape(X_test.shape[0],
    IMG_SIZE*IMG_SIZE*3)

7.  y1 = le.transform(y_train)
8.  y2 = le.transform(y_dev)
9.  y3 = le.transform(y_test)
```

y_train

X_train



Mã hoá dữ liệu

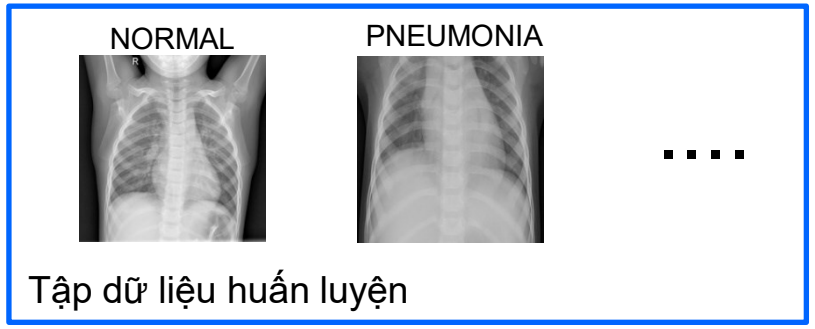
y1

X1

0	1
Array:	Array:
[[1 2 3]	[[1 2 3]
[4 5 6]	[4 5 6]
[7 8 9]]	[7 8 9]]

X2, y2 – mã hoá cho X_dev và y_dev
X3, y3 – mã hoá cho X_test và y_test

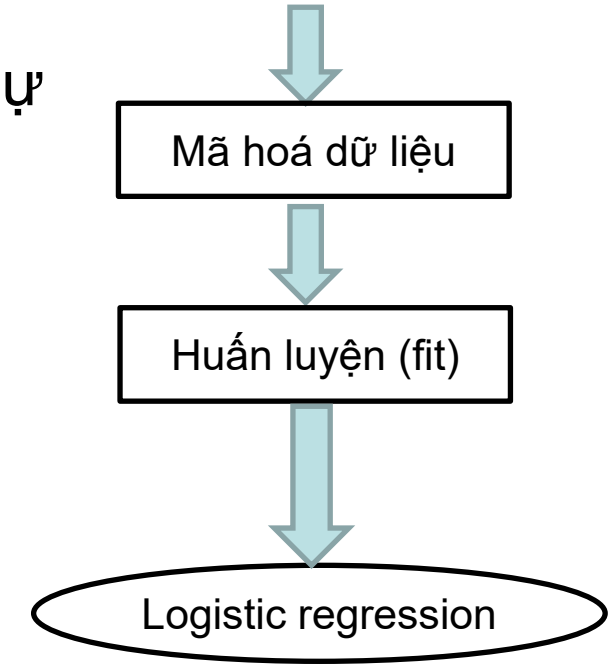
Huấn luyện mô hình



– Áp dụng mô hình Logistic Regression để phân loại và dự đoán kết quả. Giá trị dự đoán sẽ là nhãn của ảnh.

```
1. from sklearn.linear_model import LogisticRegression

2. model = LogisticRegression(random_state=0)
3. model.fit(X1, y1)
```



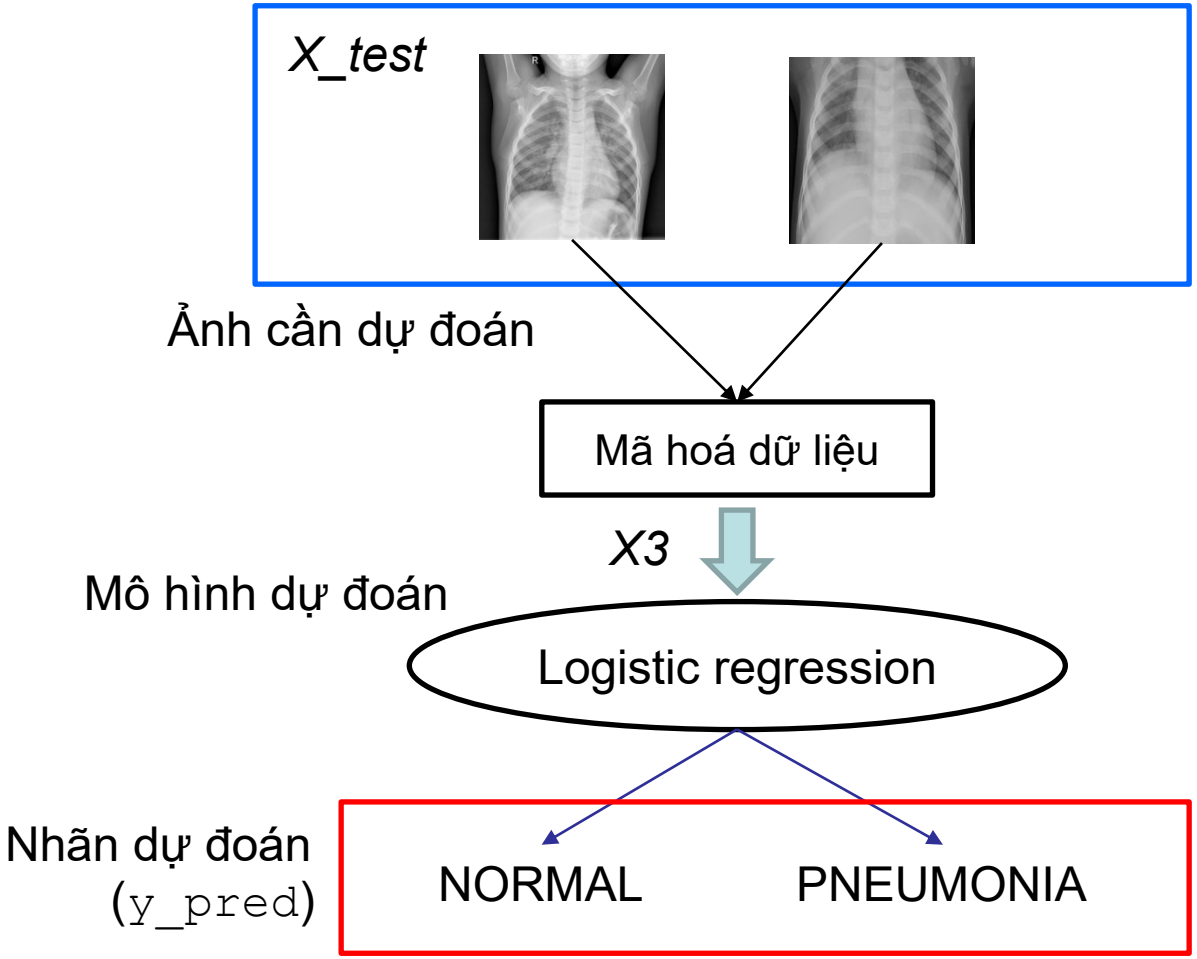
Dự đoán dữ liệu mới

– Áp dụng mô hình Logistic Regression để phân loại và dự đoán kết quả. Giá trị dự đoán sẽ là nhãn của ảnh.

```
// dự đoán kết quả
y_pred = model.predict(X3)
```

– So sánh nhãn dự đoán (`y_pred`) với nhãn thật (`y3`) ➔ đánh giá độ hiệu quả của mô hình phân lớp.

➔ Để đánh giá cần dựa vào độ đo (metric) cụ thể.



Đánh giá độ chính xác của mô hình

Sử dụng các độ đo đánh giá được hỗ trợ sẵn trong thư viện sklearn: Accuracy, Precision, Recall, F1-score,…

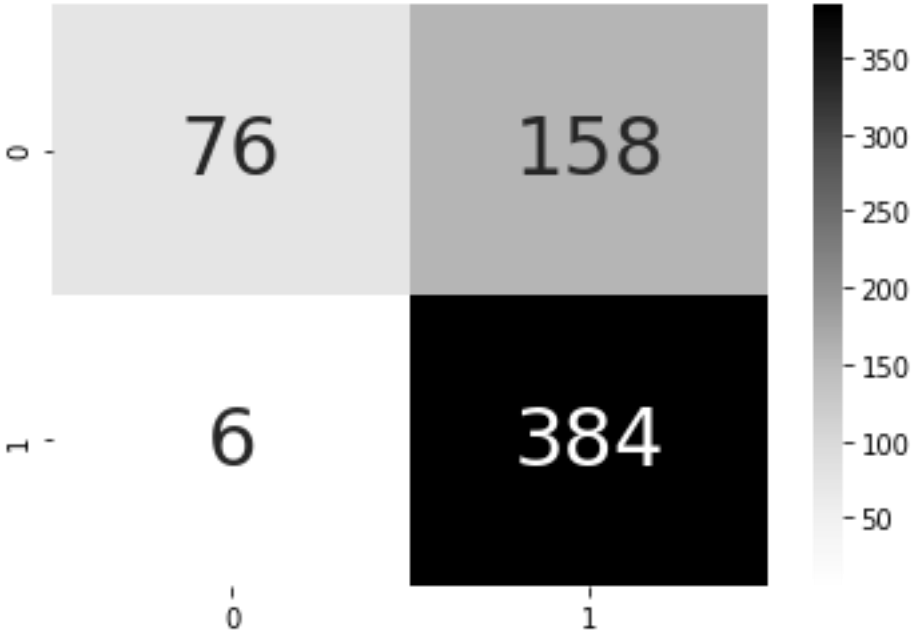
```
1. from sklearn.metrics import accuracy_score, precision_score,
   recall_score, f1_score, confusion_matrix
2. print("Accuracy: " + str(accuracy_score(y3, y_pred)))
3. print("Precision: " + str(precision_score(y3, y_pred)))
4. print("Recall: " + str(recall_score(y3, y_pred)))
5. print("F1-micro: " + str(f1_score(y3, y_pred)))
6. print("F1-macro: " + str(f1_score(y3, y_pred, average='macro')))
7. cf = confusion_matrix(y3, y_pred)
```

Kết quả của mô hình

```
Accuracy: 0.7371794871794872
Precision: 0.7084870848708487
Recall: 0.9846153846153847
F1-micro: 0.7371794871794872
F1-macro: 0.6525234964958984
```

Ý nghĩa các độ đo trên là gì ??

ma trận nhầm lẫn (confusion matrix)



Ý nghĩa của hình này là gì??

CÁC ĐỘ ĐO ĐÁNH GIÁ DÙNG CHO BÀI TOÁN PHÂN LỚP

Ma trận lỗi

– Ma trận lỗi (confusion matrix) là công cụ giúp cho việc đánh giá được khả năng của một mô hình trong tác vụ phân lớp.

Nhãn dự đoán (predicted label)

Tập nhãn: C = {0,1}

Nhãn thật sự (True label)

	0	1
0	a	b
1	d	c

- Giá trị **a**: số lượng mẫu dữ liệu **nhãn "0" được dự đoán đúng** (được dự đoán là nhãn "0").
- Giá trị **b**: số lượng mẫu dữ liệu **nhãn "0" bị dự đoán nhầm thành nhãn "1"**.
- Giá trị **c**: số lượng mẫu dữ liệu **nhãn "1" được dự đoán đúng** (được dự đoán là nhãn "1").
- Giá trị **d**: số lượng mẫu dữ liệu **nhãn "1" bị dự đoán nhầm thành nhãn "0"**.

a + b + c + d = N (số mẫu)

STT	Mẫu dữ liệu	Nhãn thật sự	Nhãn dự đoán
1	“Tổng thống Kenedy là người đàn ông đẹp trai.”		
2	“Mặt trời mọc ở hướng Đông.”		
3	“UIT là trường thành viên của ĐHQG.”		
4	“Tất cả các ngành học đại học đều chỉ kéo dài 4 năm.”		
5	“Năm 2022, VN có 63 tỉnh/thành.”		
6	“Con người là một loài động vật có vú.”		
7	“Xe máy là xe honda.”		
8	“Trên thế giới có 4 đại dương.”		
9	“Nước sôi ở 100ºC.”		
10	“Thảo là 1 loại cây cỏ”.		

Ví dụ 1

Tập nhãn: {1 = “Đúng”, 0 = “Sai”}

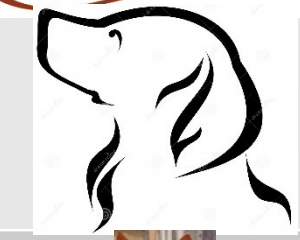
		Nhãn dự đoán	
		0	**1**
Nhãn thật sự	**0**	1	4
	1	2	3

Số mẫu: N = 10

Trong đó:

- a = 1 → 1 mẫu nhãn “0” được dự đoán đúng.
- b = 4 → 4 mẫu nhãn “0” bị dự đoán nhầm thành nhãn “1”.
- c = 3 → 3 mẫu nhãn “1” được dự đoán đúng.
- d = 2 → 2 mẫu nhãn “1” bị dự đoán nhầm thành nhãn “0”.

a + b + c + d = 1 + 4 + 3 + 2 = 10 (số mẫu)

STT	Mẫu dữ liệu	Nhãn thật sự	Nhãn dự đoán
1			
2			
3			
4			
5			
6			

Tập nhãn: {“Chó”, “Mèo”}

Ví dụ 2

		Nhãn dự đoán	
		“Chó”	“Mèo”
Nhãn thật sự	“Chó”	2	1
	“Mèo”	1	2

Số mẫu: N = 6

Trong đó:

- a = 2 → 2 mẫu nhãn “Chó” được dự đoán đúng.
- b = 1 → 1 mẫu nhãn “Chó” bị dự đoán nhầm thành nhãn “Mèo”.
- c = 2 → 2 mẫu nhãn “Mèo” được dự đoán đúng.
- d = 1 → 1 mẫu nhãn “1” bị dự đoán nhầm thành nhãn “0”.

a + b + c + d = 1 + 4 + 3 + 2 = 10 (số mẫu)

CÁC ĐỘ ĐO CHO BÀI TOÁN PHÂN LỚP

- Giả sử: 0 là negative, 1 là positive
- Dự đoán đúng là true, dự đoán sai là false

Nhãn dự đoán (predicted label)

Nhãn thực (True label)

	0	1
0	TN	FP
1	FN	TP

$$Accuracy = \frac{\mathrm{TN} + \mathrm{TP}}{\mathrm{TN} + \mathrm{FP} + \mathrm{FN} + \mathrm{TP}}$$

$$precision_{positive} = \frac{TP}{TP + FP}$$

$$precision_{negative} = \frac{TN}{TN + FN}$$

Độ chính xác trên từng nhãn

$$recall_{positive} = \frac{TP}{TP + FN}$$

$$recall_{negative} = \frac{TN}{TN + FP}$$

Độ phủ trên từng nhãn

Ý nghĩa của Precision và Recall

- Xét nhãn positive:
  + Precision cao đồng nghĩa với việc độ chính xác của các điểm *positive* tìm được là cao.
  + Recall cao đồng nghĩa với việc tỉ lệ bỏ sót các điểm thực sự là *positive* thấp. Recall còn được gọi là độ phủ.
- Đối với nhãn negative: tương tự.
- F1 score: độ đo trung bình điều hoà. Tận dụng cả 2 lợi thế của độ đo *precision* và *recall*.

$$\mathrm{F1} = 2 * \frac{\mathrm{precision} * \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

Các dạng F1 score

F1 micro score

– Tính tổng precision và recall của các nhãn.

$$\text{precision} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{recall} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{F1}_{\text{micro}} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

F1 macro score

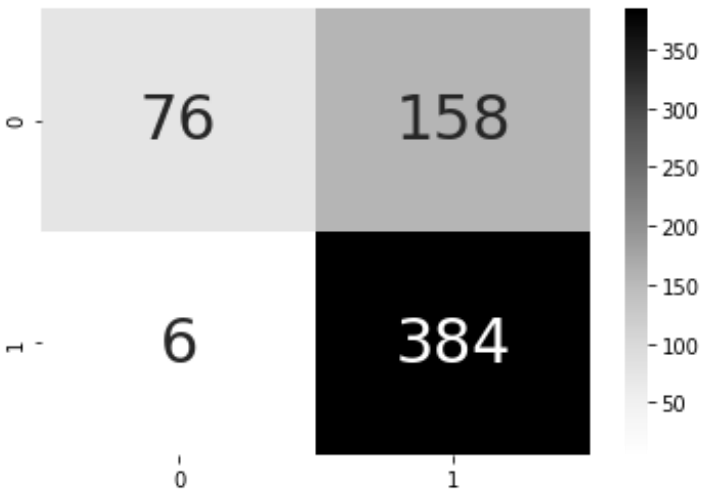
– Tính trung bình cộng của precision và recall của các nhãn

$$\text{precision} = \frac{precision_{positive} + precision_{\text{negative}}}{2}$$

$$\text{recall} = \frac{\text{recall}_{positive} + \text{recall}_{\text{negative}}}{2}$$

$$\text{F1}_{\text{macro}} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Ví dụ



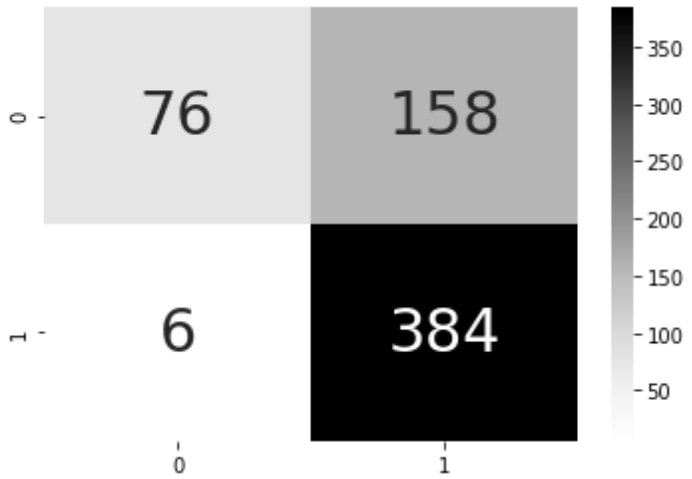
Tính hiệu năng phân lớp của mô hình Logistic Regression dựa vào các độ đo sau:

a) Accuracy

b) F1 micro

c) F1 macro

Ví dụ



$$precision_pos = \frac{384}{384 + 158} = 0.7084$$

$$precision_neg = \frac{76}{76 + 6} = 0.9268$$

$$recall_neg = \frac{76}{158 + 76} = 0.3247$$

$$recall_pos = \frac{384}{384 + 6} = 0.9846$$

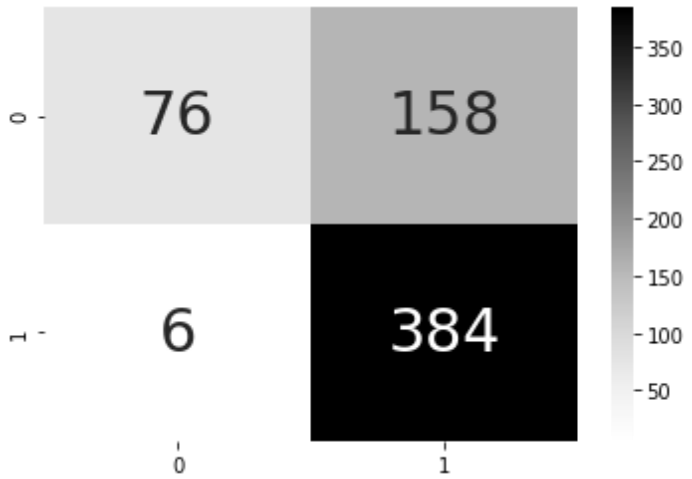
$$precision_{\text{micro}} = \frac{384 + 76}{384 + 158 + 76 + 6} = 0.7371$$

$$recall_{\text{micro}} = \frac{384 + 76}{384 + 158 + 76 + 6} = 0.7371$$

$$Accuracy = \frac{76 + 384}{76 + 158 + 6 + 384} = 0.7371$$

$$F1_{micro} = 2 * \frac{0.7371 * 0.7371}{0.7371 + 0.7371} = 0.7371$$

Ví dụ



$$precision_pos = \frac{384}{384 + 158} = 0.7084$$

$$precision_neg = \frac{76}{76 + 6} = 0.9268$$

$$recall_neg = \frac{76}{158 + 76} = 0.3247$$

$$recall_pos = \frac{384}{384 + 6} = 0.9846$$

$$precision_{\text{macro}} = \frac{0.7084 + 0.9268}{2} = 0.8176$$

$$recall_{\text{macro}} = \frac{0.3247 + 0.9846}{2} = 0.6546$$

$$Accuracy = \frac{76 + 384}{76 + 158 + 6 + 384} = 0.7371$$

$$F1_{\text{macro}} = 2 * \frac{0.8176 * 0.6546}{0.8176 + 0.6546} = 0.7270$$

*F1 macro trong **sklearn** sẽ tính bằng cách lấy trung bình F1 trên từng nhãn*

Một vài lưu ý

- Đối với các bài toán phân lớp, nên dùng độ đo F1 hơn là độ đo accuracy vì độ đo F1 thể hiện được hiệu năng dự đoán trên từng nhãn của bộ dữ liệu.
- Việc lựa chọn độ đo dựa vào:
  + Đặc thù của tác vụ / bài toán đang giải quyết.
  + Công trình đi trước.
- Để đánh giá và so sánh được tính hiệu quả của mô hình trên bộ dữ liệu thì độ đo phải thống nhất.

Độ đo cho bài toán phân lớp nhiều nhãn

Measures for multi-class classification based on a generalization of the measures of Table 1 for many classes C_i: tp_i are true positive for C_i, and fp_i – false positive, fn_i – false negative, and tn_i – true negative counts respectively. μ and M indices represent micro- and macro-averaging.

Measure	Formula	Evaluation focus
Average Accuracy	$\frac{\sum_{i=1}^{l} \frac{tp_i+tn_i}{tp_i+fn_i+fp_i+tn_i}}{l}$	The average per-class effectiveness of a classifier
Error Rate	$\frac{\sum_{i=1}^{l} \frac{fp_i+fn_i}{tp_i+fn_i+fp_i+tn_i}}{l}$	The average per-class classification error
$Precision_\mu$	$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l} (tp_i+fp_i)}$	Agreement of the data class labels with those of a classifiers if calculated from sums of per-text decisions
$Recall_\mu$	$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l} (tp_i+fn_i)}$	Effectiveness of a classifier to identify class labels if calculated from sums of per-text decisions
$Fscore_\mu$	$\frac{(\beta^2+1)Precision_\mu Recall_\mu}{\beta^2 Precision_\mu + Recall_\mu}$	Relations between data's positive labels and those given by a classifier based on sums of per-text decisions
$Precision_M$	$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i+fp_i}}{l}$	An average per-class agreement of the data class labels with those of a classifiers
$Recall_M$	$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i+fn_i}}{l}$	An average per-class effectiveness of a classifier to identify class labels
$Fscore_M$	$\frac{(\beta^2+1)Precision_M Recall_M}{\beta^2 Precision_M + Recall_M}$	Relations between data's positive labels and those given by a classifier based on a per-class average

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427–437.

Một số độ đo đánh giá khác

SV tự nghiên cứu thêm:

		Predicted Class: Positive	Predicted Class: Negative	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

PHÂN TÍCH LỖI (ERROR ANALYSIS)

Các bước thực hiện để phân tích lỗi

- Lấy ra các dữ liệu bị dự đoán nhầm bởi mô hình.
- Liệt kê nhãn dự đoán (predicted label) - nhãn thực sự (true label) kèm theo.
- Tìm hiểu xem vì sao dữ liệu bị dự đoán sai. Các đặc điểm nào của dữ liệu khiến cho mô hình dự đoán nhầm lẫn.

Dữ liệu ban đầu

NORMAL

PNEUMONIA

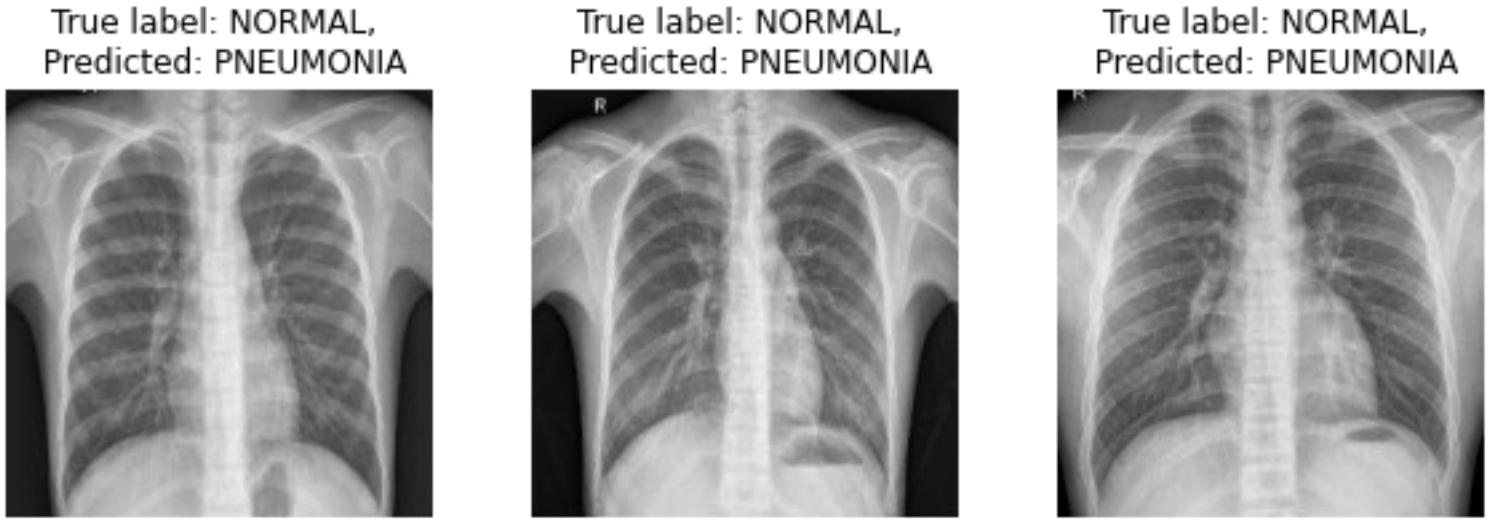
NORMAL

PNEUMONIA

NORMAL

PNEUMONIA

PHÂN TÍCH LỖI



- Tại sao 3 bức ảnh trên bị dự đoán sai từ nhãn NORMAL thành nhãn PNEUMONIA?
- Đặc trưng nào giúp phân biệt được giữa NORMAL và nhãn PNEUMONIA?
- Để lấy được đặc trưng đó cần sử dụng các kỹ thuật / công cụ gì?
- Có công trình / bài báo nào đã áp dụng kỹ thuật / công cụ đó để giải quyết cho cùng bài toán, hoặc bài toán tương tự hay chưa?

VAI TRÒ PHÂN TÍCH LỖI

- Tìm ra các đặc điểm hạn chế của mô hình.
- Hiểu thêm về đặc điểm của dữ liệu cho bài toán đang xét.
- Từ các đặc điểm hạn chế trên ➔ tiến hành cải tiến mô hình, hoặc đề xuất giải pháp cải tiến trong tương lai (future works).

Đối với một dự án máy học, phân tích lỗi là nội dung rất cần thiết, bên cạnh các kết quả về độ chính xác của mô hình. Phân tích lỗi là cơ sở để mở đường cho các nghiên cứu tiếp theo.

CÁC DẠNG PHÂN LỚP KHÁC

MULTI-CLASS CLASSIFICATION

– Input: Dữ liệu cần dự đoán.

– Output: nhãn c của dữ liệu, thuộc tập nhãn C. |C| > 2.

VD: **NHẬN DIỆN CUNG BẬC CẢM XÚC CỦA BÌNH LUẬN TRÊN MẠNG XÃ HỘI (VSMEC)**

– Input: Một câu bình luận dưới dạng văn bản (text).

– Output: 1 trong 7 cung bậc cảm xúc khác nhau: *Enjoyment, Sadness, Fear, Anger, Disgust, Surprise, Other*.

Ho V.A. et al. (2020) Emotion Recognition for Vietnamese Social Media Text (PACLING 2019)
https://arxiv.org/abs/1911.09339

Xây dựng bộ phân lớp đa lớp từ bộ phân lớp nhị phân

- Một với tất cả nhãn còn lại (OvR): sử dụng *n* bộ phân lớp nhị phân.

VD: đối với 7 cung bậc cảm xúc từ bộ VSMEC, có 7 bộ nhận diện cảm xúc khác nhau, sau đó chọn bộ nhận diện nào có điểm số cao nhất.

- Một với một (OvO): sử dụng $n(n-1)/2$ bộ phân lớp nhị phân.

VD: đối với 7 cung bậc cảm xúc, có 21 bộ nhận diện khác nhau giữa các cảm xúc: enjoyment – fear, enjoyment – sadness, ...

MULTI-LABEL CLASSIFICATION

- Input: Dữ liệu cần dự đoán.
- Output: Tập nhãn dự đoán $C = \{c_1, c_2, ..., c_n\}$.

VD: **NHẬN DIỆN CẢM XÚC THEO KHÍA CẠNH CHO DỮ LIỆU NHÀ HÀNG KHÁCH SẠN**

Input: một câu bình luận dưới dạng văn bản.

Output: tập nhãn bao gồm: {cảm xúc 1#khía cạnh 1, ..., cảm xúc n#khía cạnh n}

Dang Van Thin, Ngan Luu-Thuy Nguyen, Tri Minh Truong, Lac Si Le, and Duy Tin Vo. 2021. Two New Large Corpora for Vietnamese Aspect-based Sentiment Analysis at Sentence Level. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 20, 4, Article 62

TỔNG KẾT

1. Khái niệm về tác vụ phân lớp (đầu vào, đầu ra là gì?)
2. Tác vụ phân lớp và hồi quy khác nhau như thế nào?
3. Các dạng bài toán phân lớp
4. Đánh giá một mô hình phân lớp ra sao?
5. Vai trò của phân tích lỗi.

BÀI TẬP ÁP DỤNG

- Các bạn thực hiện đề tài phân loại các loài hoa dựa vào ảnh chụp của hoa. Sử dụng bộ dữ liệu Iris.
- *Mô hình phân loại*: các bạn có thể tự chọn mô hình.
- Bộ dữ liệu Iris: bộ này được hỗ trợ sẵn, các bạn có thể tham khảo tại đây:

https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html

Yêu cầu

- Thực hiện theo các bước của tác vụ máy học như đã hướng dẫn.
- Hướng dẫn: bộ này không có chia ra train / test sẵn, các bạn nên dùng kỹ thuật cross validation để huấn luyện và kiểm tra:

https://scikit-learn.org/stable/modules/cross_validation.html

- Phân tích lỗi:
  + Cho biết các trường hợp nào mô hình dự đoán sai, vì sao?
  + *Đề ra giải pháp nâng cao độ chính xác của mô hình.

TÀI LIỆU THAM KHẢO

1. Chương 3 của sách: *Hands-on Machine Learning with ScikitLearn, Keras & TensorFlow.*
2. Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427–437. doi:10.1016/j.ipm.2009.03.002