TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

**KHOA CÔNG NGHỆ THÔNG TIN**



***BÁO CÁO CHUYÊN ĐỀ MÔN HỌC MÁY NÂNG CAO***

**ĐỀ TÀI: TÌM HIỂU KIẾN TRÚC MẠNG U-NET VÀ ỨNG DỤNG**

|  |  |
| --- | --- |
|  |  |
| |  |  | | --- | --- | | **Sinh viên thực hiện** | **: NGUYỄN ĐỨC THỊNH**  **NGUYỄN BÁ KIÊN**  **ĐỖ TUẤN KIỆT** | | **Giảng viên hướng dẫn** | **: PHẠM THỊ KIM DUNG** | | | **Ngành** | **: CÔNG NGHỆ THÔNG TIN** | | | **Chuyên ngành** | **: CÔNG NGHỆ PHẦN MỀM** | | | **Lớp** | **: D12CNPM2** | | | **Khóa** | **: 2017-2022** | |     ***Hà Nội, tháng 1 năm 2021*** |  |

**PHIẾU CHẤM ĐIỂM**

Sinh viên thực hiện:

|  |  |  |  |
| --- | --- | --- | --- |
| **Họ và tên** | **Nôi dung thực hiện** | **Chữ ký** | **Ghi chú** |
| Nguyễn Đức Thịnh  (1781310171) |  |  |  |
| Nguyễn Bá Kiên  (178310140) |  |  |  |
| Đỗ Tuấn Kiệt  (1781310141) |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Họ và tên giảng viên** | **Chữ ký** | **Ghi chú** |
| Giảng viên chấm 1: |  |  |
| Giảng viên chấm 2: |  |  |

MỤC LỤC

[LỜI MỞ ĐẦU 2](#_Toc60957913)

[CHƯƠNG 1: TỔNG QUAN KIẾN TRÚC MẠNG U-NET 3](#_Toc60957914)

[1.1. Giới thiệu chung 3](#_Toc60957915)

[1.2. Kiến trúc mạng U-net 4](#_Toc60957916)

[1.3. Ứng dụng của mạng U-net 5](#_Toc60957917)

[1.4. Bài toán phân đoạn hình ảnh polyp vòm họng với U-Net 7](#_Toc60957918)

[CHƯƠNG 2: BÀI TOÁN HỌC MÁY 13](#_Toc60957919)

[1. Sử dụng kỹ thuật học máy phân lớp Logistic Regressgion để dự đoán khả năng người dùng Internet nhấp vào quảng cáo 13](#_Toc60957920)

[1.1. Mô tả bài toán 13](#_Toc60957921)

[1.2 Thu thập và mô tả dữ liệu 13](#_Toc60957922)

[1.3. Xử lí dữ liệu 15](#_Toc60957923)

[1.4. Kỹ thuật học máy sử dụng 18](#_Toc60957924)

[1.5. Phương pháp đánh giá độ chính xác 20](#_Toc60957925)

[1.6. Kết luận 21](#_Toc60957926)

[2. Sử dụng kỹ thuật học máy K-Means để giải bài toán phân cụm hạt giống lúa mì 22](#_Toc60957927)

[2.1 Mô tả bộ dữ liệu 22](#_Toc60957928)

[2.2 Mô tả bài toán 23](#_Toc60957929)

[2.3 Mô tả chi tiết về cách xử lý dữ liệu 23](#_Toc60957930)

[2.4 Mô tả về K-Means 24](#_Toc60957931)

# LỜI MỞ ĐẦU

Ngày nay, với cuộc cách mạng 4.0 cùng sự bùng nổ và phát triển của mạng 5G, AI (Artificial Intelligence) hay còn gọi là Trí tuệ nhân tạo đã trở thành một ngành khoa học máy tính được nhiều công ty công nghệ lớn trên thế giới rất quan tâm phát triển. Trí tuệ nhân tạo là một bộ phận của khoa học máy tính liên quan đến việc tự động hóa và thông minh, do đó nó được xây dựng dựa trên những nguyên lý lý thuyết vứng chắc, có khả năng ứng dụng cao len lỏi vào mọi lĩnh vực trong đời sống chúng ta.

Hầu hết mọi ngành công nghệ đang làm việc với lượng dữ liệu lớn ngày nay đều nhận ra tầm quan trọng của học máy, bởi học máy là một phương pháp để huấn luyện dữ liệu, giúp máy tính trở nên thông minh hơn thông qua các phương pháp học và cụ thể hơn là các thuật toán. Vậy nên, các ứng dụng của học máy đã dần trở nên quen thuộc với mọi người: xe tự hành của Google và Tesla, hệ thống tự tag khuôn mặt trên Facebook, hệ thống gợi ý sản phẩm của Amazon, …. Sau một thời gian tìm hiểu về học máy với mong muốn áp dụng được những kiến thức đã học vào thực tiễn, chúng em đã quyết định chọn đề tài “TÌM HIỂU KIẾN TRÚC MẠNG U-NET VÀ ỨNG DỤNG”

Chúng em xin chân thành cảm ơn!

# CHƯƠNG 1: TỔNG QUAN KIẾN TRÚC MẠNG U-NET

## 1.1. Giới thiệu chung

Thị giác là một trong những giác quan quan trọng nhất mà con người sở hữu. Nhưng bạn đã bao giờ tự hỏi về sự phức tạp của nó? Khả năng thu các tia sáng phản xạ và thu được ý nghĩa từ nó là một nhiệm vụ rất phức tạp và chúng ta thực hiện nó rất dễ dàng. Chúng ta đã phát triển nó trong hàng triệu năm tiến hóa. Vậy làm thế nào chúng ta có thể cung cấp cho máy móc khả năng tương tự trong một khoảng thời gian rất ngắn? Đối với máy tính, những hình ảnh này không là gì ngoài ma trận và hiểu được các sắc thái đằng sau các ma trận này đã là nỗi ám ảnh đối với nhiều nhà toán học trong nhiều năm. Nhưng sau khi xuất hiện trí tuệ nhân tạo và đặc biệt là kiến trúc CNN, nghiên cứu đã đạt được tiến bộ hơn bao giờ hết. Nhiều vấn đề trước đây được coi là không thể giải quyết hiện đang cho thấy kết quả đáng kinh ngạc.

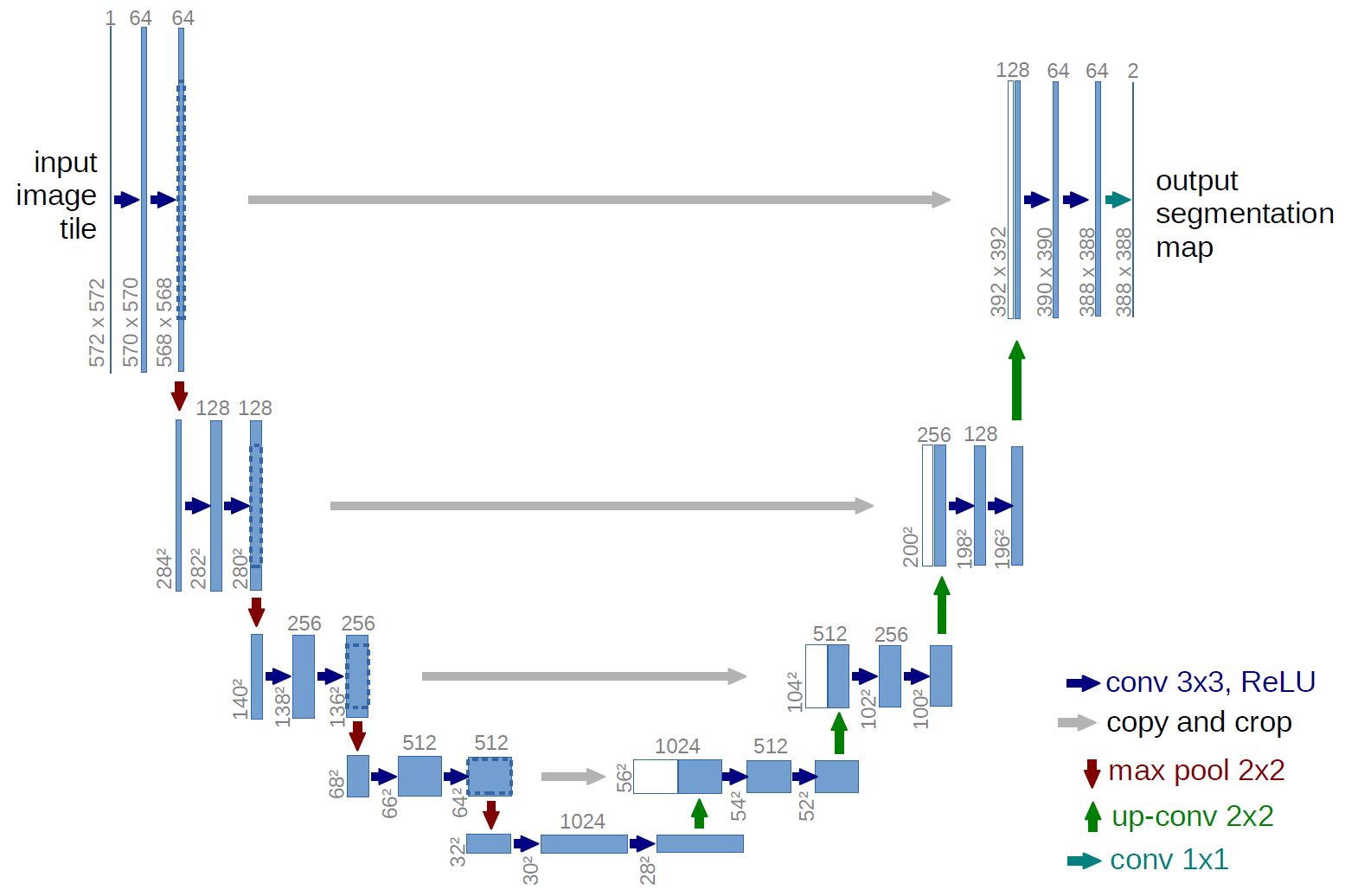
Một vấn đề như vậy là Image Segmentation (phân vùng ảnh). Trong phân vùng ảnh, máy phải phân vùng hình ảnh thành các vùng, mỗi vùng biểu diễn một thực thể khác nhau.



Như bạn có thể thấy ở hình phía trên, ảnh được chia thành hai vùng, một vùng là con mèo và một vùng là background. Phân vùng ảnh hữu ích trong nhiều lĩnh vực từ xe tự hành cho tới xử lý ảnh vệ tinh. Quan trọng nhất có lẽ là hình xử lý ảnh y tế. Các ảnh y tế khá phức tạp và đôi khi thậm chí là thách thức đối với các bác sĩ đã được đào tạo. Một cỗ máy có khả năng hiểu và xác định các khu vực cần thiết có thể tạo ra một tác động sâu sắc trong chăm sóc y tế.

Convolutional Neural Networks đã cho kết quả tốt trong các tác vụ phân vùng hình ảnh dễ nhưng chưa đạt hiểu quả tốt với các tác vụ phức tạp. Đó là nơi mà U-net xuất hiện. U-net ban đầu được thiết kế cho phân vùng ảnh y tế. Nó cho kết quả tốt và được sử dụng trong nhiều lĩnh vực khác sau đó.

## 1.2. Kiến trúc mạng U-net



Hình 1: Kiến trúc U-net

**1.2.1. Cách U-net hoạt động**

Kiến trúc của U-net giống như hình chữ ‘U’, điều này giải thích cho tên gọi của nó. Kiến trúc này gồm 3 phần: contraction (encoder), bottleneck và expansion (decoder). Phần contraction được tạo bởi nhiều khối. Mỗi khối gồm hai lớp CNN có kernel size là 3×3, theo sau bởi một lớp 2×2 max pooling. Số lượng kernel hay feature map tăng gấp đôi sau mỗi khối, do đó kiến trúc này có thể học những đặc trưng phức tạp một cách hiệu quả. Lớp ở vị trí thấp nhất làm trung gian giữa các lớp ở phần contraction và phần expansion. Nó dùng 2 lớp 3×3 CNN, theo sau bởi 1 với 2×2 up convolution.

Nhưng trái tim của kiến trúc này nằm ở phần expansion. Tương tự như contraction, nó cũng bao gồm nhiều khối. Mỗi khối gồm hai lớp 3×3 CNN theo sau bởi một lớp upsampling. Sau mỗi khối số lượng feature map được giảm đi một nửa để đảm bảo tính đối xứng. Tuy nhiên đầu vào của mỗi khối expansion sẽ được nối thêm feature map của khối contraction tương ứng. Hành động này sẽ đảm bảo các được trưng được học trong quá trình contracting ảnh sẽ được dùng để tái tạo nó. Số lượng khối expansion bằng với số lượng khối contraction. Sau cùng, kết quả sẽ được đưa qua một lớp 3×3 CNN khác với số feature map bằng với số vùng mong muốn.

**1.2.2. Tính toán loss trong U-net**

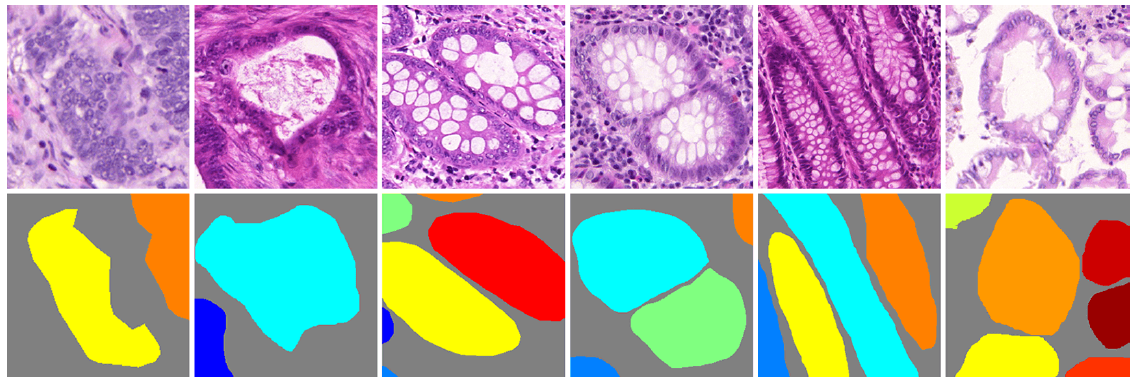
U-net sử dụng một cơ chế tính trọng số loss cho từng pixel sao cho trọng số cao hơn ở viền của các vùng chứa đối tượng. Cơ chế loss có trọng số này giúp cho mô hình U-net phân vùng các tế bào trong ảnh y sinh theo kiểu không liên tục sao cho các tế bào riêng lẻ có thể được xác định trong ảnh phân vùng nhị phân.

Trước hết, pixel-wise softmax được áp dụng trên hình ảnh kết quả, tiếp theo là hàm cross-entropy. Vì vậy, chúng ta đang phân loại từng pixel thành một trong số các lớp. Ý tưởng là khi phân vùng, mọi pixel đều phải nằm trong một số lớp. Vì vậy, chúng ta chỉ chuyển một bài toán phân vùng thành một vài toán phân loại đa lớp và nó cho kết quả rất tốt so với các hàm loss truyền thống.

## 1.3. Ứng dụng của mạng U-net

Mạng U-net được sử dụng trong việc Image Segmentation nên có rất nhiều các ứng dụng trong y học, xe tự hành, xử lý ảnh vệ tinh.

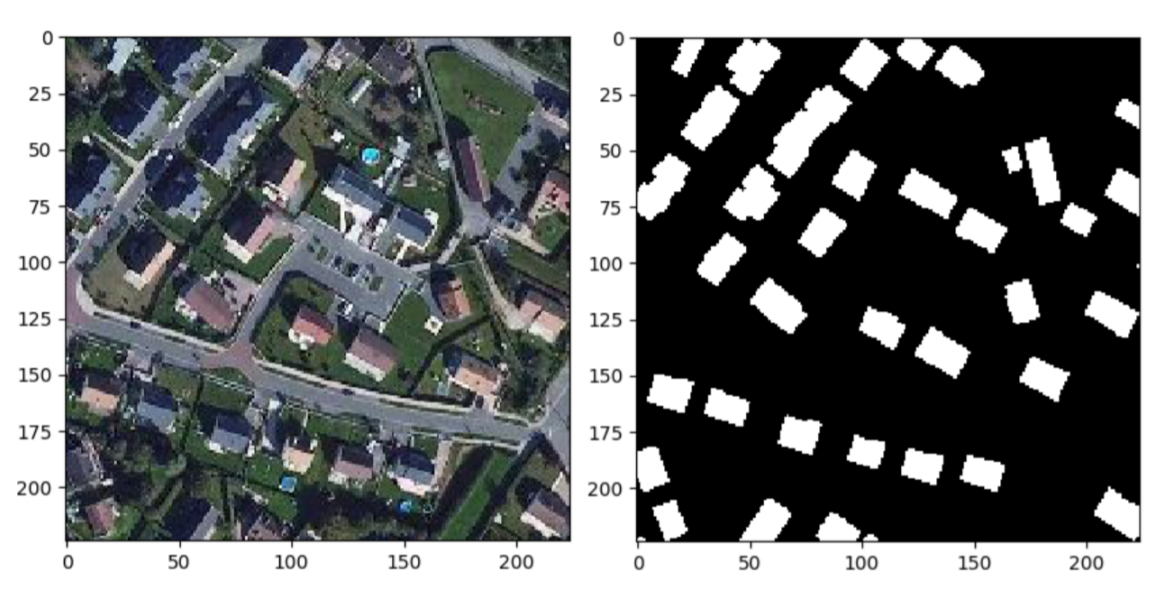
* **Y học:** Trong y học, thuật toán Image Segmentation có thể hỗ trợ bác sĩ chuẩn đoán khối u từ ảnh x-quang. Ưu điểm của Image Segmentation đó là không chỉ cho chúng ta biết vị trí của các khối u trong ảnh mà còn cho chúng ta biết được hình dạng của chúng



* **Xe tự hành:** Xe tự hành đòi hỏi phải liên tục nhận thức, xử lý và lên kế hoạch trong một môi trường phát triển liên tục. Vì yêu cầu an toàn tuyệt đối và độ chính xác cao trong mọi quyết định nên một hệ thống xe tự hành cần phải xác định chính xác các vật thể xuất hiện khi tham gia giao thông như người, đèn tín hiệu, biển báo, vạch kẻ đường, xe cộ.



* **Xử lý ảnh vệ tinh:** Các vệ tinh quay quanh trái đất sẽ liên tục thu thập hình ảnh bề mặt trái đất ở những vùng khác nhau. Từ các bức ảnh chụp vệ tinh, mô hình Image Segmentation sẽ phân đoạn hình ảnh thành tuyến đường, khu phố, biển cả, cây cối,...



* **Ứng dụng trong nông nghiệp:** Chúng ta có thể tiết kiệm được một lượng lớn thuốc trừ sâu trong nông nghiệp nhờ sử dụng hệ thống phun thuốc trừ sâu tự động có khả năng phân biệt được diện tích cỏ và cây trồng dựa trên thuật toán Image Segmentation. Khi diện tích cỏ lấn át so với cây trồng thì hệ thống sẽ tự động kích hoạt.
* **Cảnh báo cháy rừng:** Những hệ thống kiểm soát cháy rừng có thể segment được chính xác vị trí phát sinh các đám cháy từ ảnh chụp vệ tinh. Từ đó đưa ra cảnh báo về qui mô và mức độ lây lan của các đám cháy trên diện rộng.

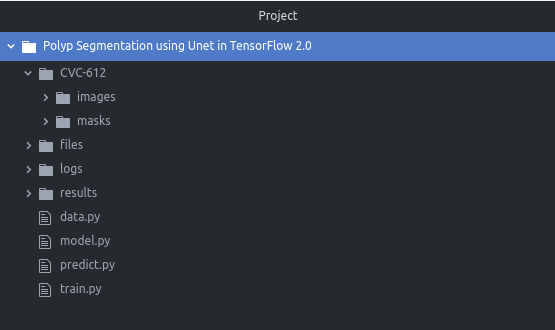
**1.4. Bài toán phân đoạn hình ảnh polyp vòm họng với U-Net**

Bài toán phân đoạn hình ảnh polyp (một dạng như khối u) vòng họng bằng cách sử dụng học sâu, kiến ​​trúc UNet, OpenCV và các thư viện khác. Chúng em sử dụng tập dữ liệu phân đoạn polyp để hiểu cách **phân đoạn ngữ nghĩa** được áp dụng cho trong bộ dữ liệu này.

**Phân đoạn ngữ nghĩa** là một trong những thách thức trong nghiên cứu thị giác máy tính. Nó có thể được coi là một vấn đề phân loại, nhưng ở cấp độ pixel vì bạn cần dự đoán nhãn hoặc lớp của mọi pixel trong một hình ảnh.

Phân đoạn ngữ nghĩa là một nhiệm vụ tiến tới việc hiểu toàn cảnh bằng cách học cách dự đoán những gì có trong hình ảnh. Tầm quan trọng của khả năng hiểu cảnh nằm ở các ứng dụng của nó đối với lĩnh vực thị giác máy tính, chẳng hạn như xe tự lái, tương tác giữa con người và máy tính, robot tự hành, thực tế tăng cường, hệ thống nhận dạng khuôn mặt, v.v.

Trong bài toán này, cấu trúc thư mục được chia ra và dưới đây là định nghĩa về các thư mục của dự án này.



Dự án có bốn thư mục:

1. **CVC-612 /:** Nó bao gồm tập dữ liệu mà chúng tôi sẽ sử dụng cho dự án này. Nó chứa hai thư mục con: hình ảnh và mặt nạ. Như tên gọi của chúng cho thấy, các thư mục con này chứa các hình ảnh và mặt nạ.
2. **files /:** Thư mục này được sử dụng để lưu trữ tệp CSV chứa tất cả thông tin trong khi mô hình đang đào tạo. Nó cũng lưu trữ tệp trọng lượng mô hình.
3. **logs /:** nó chứa các tệp nhật ký TensorBoard.
4. **results /:** Nó được sử dụng để lưu trữ kết quả sau khi đưa ra dự đoán trên tập dữ liệu thử nghiệm.

Dự án cũng có bốn tập lệnh python:

1. **data.py:** Tệp này chứa code để tải tập dữ liệu, đọc hình ảnh và mặt nạ. Nó cũng được sử dụng để tạo đường dẫn tf.data để đào tạo, xác thực và thử nghiệm tập dữ liệu.
2. **model.py:** Tệp này có code cho kiến trúc UNet dùng để phân đoạn các hình ảnh polyp.
3. **train.py:** Tệp này giúp mô hình huấn luyện trên tập dữ liệu huấn luyện. Nó cũng được sử dụng để lưu mô hình mà sau này được sử dụng để đưa ra dự đoán trên tập dữ liệu thử nghiệm.
4. **dự đoán.py:** Sau khi kết thúc đào tạo, tệp này được sử dụng để đưa ra các dự đoán trên tập dữ liệu kiểm tra.

Tập dữ liệu chứa hình ảnh được trích xuất từ ​​các video nội soi. Hình ảnh này chứa các loại polyp khác nhau.

Hướng dẫn cài đặt: Để thực hiện việc phân đoạn polyp, chúng em sử dụng IDE PyCharm, cài đặt TensorFlow 2.0 với Python 3.7

Code: Vì dự án có khá nhiều các tệp py và mất nhiều thời gian để giải thích, chúng em xin minh họa phần code Predict.py để phân đoạn polyp sau khi đã huấn luyện mô hình.

import os

import numpy as np

import cv2

import tensorflow as tf

from tensorflow.keras.utils import CustomObjectScope

from tqdm import tqdm

from data import load\_data, tf\_dataset

from train import iou

def read\_image(path):

x = cv2.imread(path, cv2.IMREAD\_COLOR)

x = cv2.resize(x, (256, 256))

x = x/255.0

return x

def read\_mask(path):

x = cv2.imread(path, cv2.IMREAD\_GRAYSCALE)

x = cv2.resize(x, (256, 256))

x = np.expand\_dims(x, axis=-1)

return x

def mask\_parse(mask):

mask = np.squeeze(mask)

mask = [mask, mask, mask]

mask = np.transpose(mask, (1, 2, 0))

return mask

if \_\_name\_\_ == "\_\_main\_\_":

## Dataset

path = "CVC-612/"

batch\_size = 8

(train\_x, train\_y), (valid\_x, valid\_y), (test\_x, test\_y) = load\_data(path)

test\_dataset = tf\_dataset(test\_x, test\_y, batch=batch\_size)

test\_steps = (len(test\_x)//batch\_size)

if len(test\_x) % batch\_size != 0:

test\_steps += 1

with CustomObjectScope({'iou': iou}):

model = tf.keras.models.load\_model("files/model.h5")

model.evaluate(test\_dataset, steps=test\_steps)

for i, (x, y) in tqdm(enumerate(zip(test\_x, test\_y)), total=len(test\_x)):

x = read\_image(x)

y = read\_mask(y)

y\_pred = model.predict(np.expand\_dims(x, axis=0))[0] > 0.5

h, w, \_ = x.shape

white\_line = np.ones((h, 10, 3)) \* 255.0

all\_images = [

x \* 255.0, white\_line,

mask\_parse(y), white\_line,

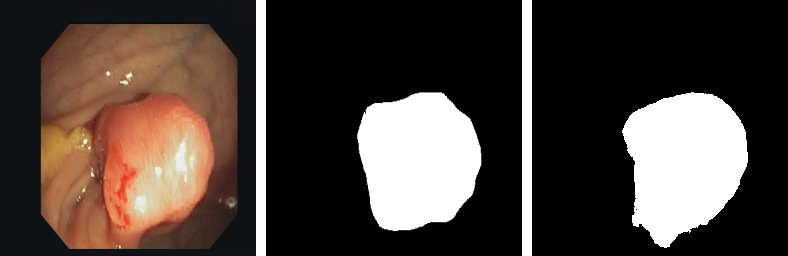
mask\_parse(y\_pred) \* 255.0

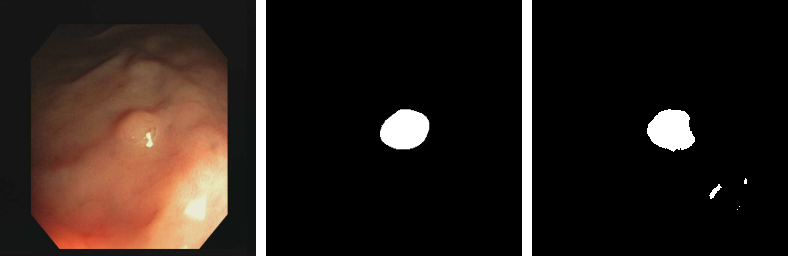
]

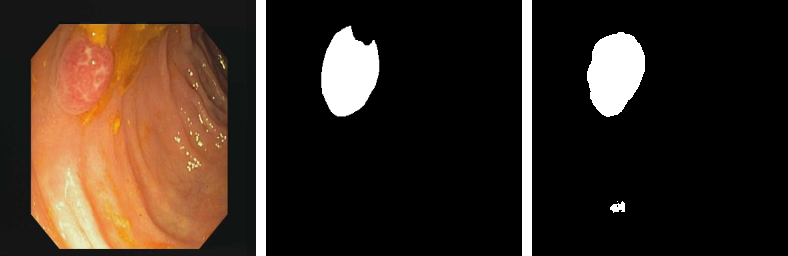
image = np.concatenate(all\_images, axis=1)

cv2.imwrite(f"results/{i}.png", image

Kết quả sau khi dự đoán:







# CHƯƠNG 2: BÀI TOÁN HỌC MÁY

## 1. Sử dụng kỹ thuật học máy phân lớp Logistic Regressgion để dự đoán khả năng người dùng Internet nhấp vào quảng cáo

### 1.1. Mô tả bài toán

Nhấp chuột là một số liệu tiếp thị tính số lượng khách hàng đã tương tác với quảng cáo của bạn. Nhấp chuột thể hiện phần trăm nhấp chuột trên tổng số lần xem quảng cáo. Nó xác định tỉ lệ thành công của chiến dịch tiếp thị của bạn. Số lượng nhấp chuột cao hơn là tín hiệu cho thấy đối tượng của bạn đang nhận được thương mại / quảng cáo thích hợp và lợi tức đầu tư (ROI) cao hơn cho công ty của bạn. Vì vậy, mục đích của phân tích này là để dự đoán ai và tại sao người dùng Internet sẽ nhấp vào quảng cáo của bạn.

Trong báo cáo này, em sử dụng kỹ thuật học máy Logistic Regression để để giải quyết bài toán phân lớp nhị phân, dự đoán người dùng có khả năng click vào quảng cáo hay không dựa trên các đặc điểm của họ. Phương pháp đánh giá mô hình học máy được sử dụng là đo độ chính xác Accuracy và Confusion Matrix.

### 1.2 Thu thập và mô tả dữ liệu

Trong bài toán dự đoán quảng cáo này, em đã tìm hiểu và thu thập đươc tập dữ liệu các đặc điểm của người dùng Internet. Tập dữ liệu nói trên được em thu thập từ **https://www.kaggle.com/fayomi/advertising.** Tập dữ liệu bao gồm 1001 dòng mẫu với 10 cột đặc điểm của người dùng.

**1.2.1. Mô tả thuộc tính**

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuộc tính** | **Mô tả** |
| 1 | Daily Time Spent on Site | Thời gian người dùng sử dụng trang web tính bằng phút |
| 2 | Age | Tuổi của người dùng |
| 3 | Area Income | Thu thập trung bình theo khu vực địa lý của người dùng |
| 4 | Daily Internet Usage | Trung bình thời gian người dùng truy cập Internet mỗi ngày tính bằng phút |
| 5 | Ad Topic Line | Tiêu đề của bài quảng cáo |
| 6 | City | Thành phố của người dùng |
| 7 | Male | Người dùng có phải nam hay không? 1 là đúng, 0 là sai |
| 8 | Country | Đất nước của người dùng |
| 9 | Timestamp | Thời gian người tiêu dùng nhấp vào Quảng cáo hoặc đóng cửa sổ |
| 10 | Clicked on Ad | 1 là click vào quảng cáo, 0 là không click vào quảng cáo |

Với 10 đặc điểm của người dùng Internet như trên, để phù hợp với yêu cầu là dự đoán người dùng nào sẽ click vào quảng cáo, chúng em chọn thuộc tính **Y là Clicked on Ad** ở đây làm biến mục tiêu cần tìm, các thuộc tính còn lại làm biến giải thích.

**1.2.2. Số lượng mẫu**

Tập dữ liệu của em bao gồm 1001 dòng tương ứng với số lượng mẫu. Dưới đây là một số mẫu có trong tập dữ liệu:

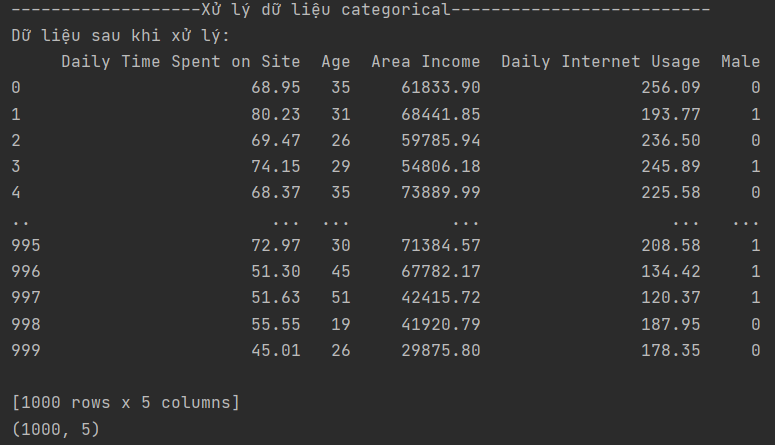
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Daily Time Spent on Site** | **Age** | **Area Income** | **Daily Internet Usage** | **Ad Topic Line** | **City** | **Male** | **Country** | **Timestamp** | **Clicked on Ad** |
| 68.95 | 35 | 61833.9 | 256.09 | Cloned 5thgeneration orchestration | Wrightburgh | 0 | Tunisia | 27/03/2016 0:53 | 0 |
| 80.23 | 31 | 68441.85 | 193.77 | Monitored national standardization | West Jodi | 1 | Nauru | 04/04/2016 1:39 | 0 |
| 69.47 | 26 | 59785.94 | 236.5 | Organic bottom-line service-desk | Davidton | 0 | San Marino | 13/03/2016 20:35 | 0 |
| 74.15 | 29 | 54806.18 | 245.89 | Triple-buffered reciprocal time-frame | West Terrifurt | 1 | Italy | 10/01/2016 2:31 | 0 |
| 68.37 | 35 | 73889.99 | 225.58 | Robust logistical utilization | South Manuel | 0 | Iceland | 03/06/2016 3:36 | 0 |
| 59.99 | 23 | 59761.56 | 226.74 | Sharable client-driven software | Jamieberg | 1 | Norway | 19/05/2016 14:30 | 0 |
| 88.91 | 33 | 53852.85 | 208.36 | Enhanced dedicated support | Brandonstad | 0 | Myanmar | 28/01/2016 20:59 | 0 |

### 1.3. Xử lí dữ liệu

**Bước 1:** đọc file dữ liệu ở dạng csv, xử lí các cột dữ liệu ở dạng object: “Ad Topic Line”, “City”, “Country”, “Timestamp”. Chọn biến mục tiêu y = “Clicked on Ads”

**Code và kết quả:**

from sklearn import preprocessing  
df = pd.read\_csv("advertising.csv")  
print('-------------------Xử lý dữ liệu categorical--------------------------')  
  
X= df.drop(["Clicked on Ad","Timestamp","City","Country","Ad Topic Line"], axis = 1)y = df["Clicked on Ad"]  
print('Dữ liệu sau khi xử lý:')  
print(X)  
print (X.shape)

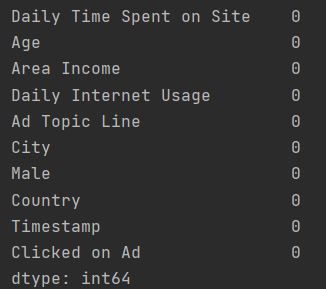
****

**Bước 2:** Kiểm tra dữ liệu thiếu

**Code và kết quả:**

print ('---------------------Kiểm tra missing value---------------------------')

print(df.isnull().sum()) #Kiểm tra missing value

****

**Bước 3:** Chia tập dữ liệu thành 2 tập train và test với train = 0.7, test = 0.3

**Code và kết quả:**

print ('-------------------------Chia dữ liệu train-test----------------------')

from sklearn.model\_selection import train\_test\_split #chia tập dữ liêu train-test

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=35)

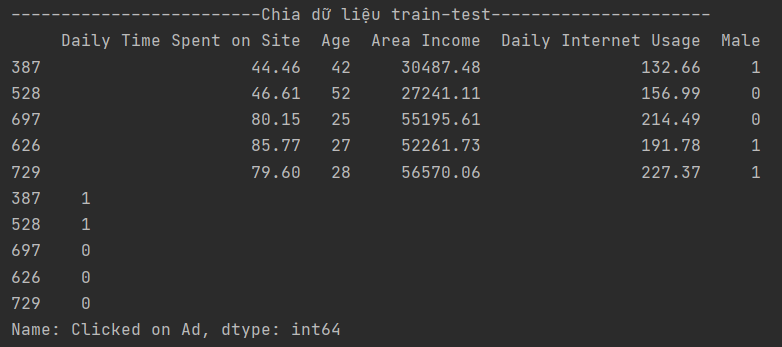
print (X\_train.head(5))

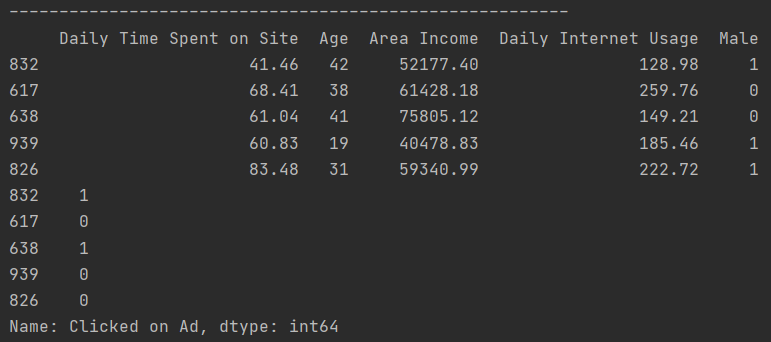
print (y\_train.head(5))

print ('--------------------------------------------------------')

print (X\_test.head(5))

print (y\_test.head(5))

****

****

**Bước 4:** Kiểm tra dữ liệu có bị mất cân bằng hay không?

**Code và kết quả:**

print ('---------------------Cân bằng SMOTE---------------------')

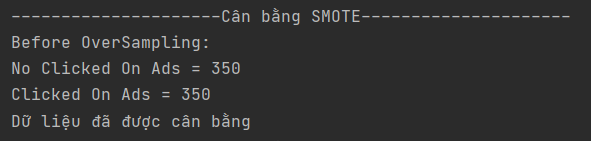
#Cân bằng smote

print ('Before OverSampling:')

print ('no =', sum (y\_train == 0))

print ('yes =', sum (y\_train == 1))

print('Dữ liệu đã được cân bằng')



### 1.4. Kỹ thuật học máy sử dụng

Với bài toán trên, em sử dụng kỹ thuật học máy Logistic Regression. Logistic Regression là 1 thuật toán phân loại được dùng để gán các đối tượng cho 1 tập hợp giá trị rời rạc (như 0, 1, ...). Một ví dụ điển hình là phân loại Email, gồm có email công việc, email gia đình, email spam,... Giao dịch trực tuyến có là an toàn hay không an toàn, khối u lành tính hay ác tình. Thuật toán trên dùng hàm sigmoid logistic để đưa ra đánh giá theo xác suất.

Kỹ thuật Logistic Regression sử dụng hàm Sigmoid làm hàm dự đoán. Hàm này có dạng:

https://images.viblo.asia/43965a36-f47e-4f23-8d05-695a315f7ce5.png

**Code và kết quả:**

print ('--------------------Chạy kỹ thuật học máy----------------')

#Kỹ thuật học máy Logistic Regression

from sklearn.linear\_model import LogisticRegression

#Bước 2. Huấn luyện (với tập dữ liệu train X\_train, y\_train)

model = LogisticRegression(max\_iter = 10000).fit(X\_train,y\_train)

#Bước 3. Dự đoán (với tập dữ liệu train X\_test, y\_test)

y\_pred = model.predict(X\_test) #y\_prediction là y dự đoán được

print ("Hệ số w:", model.coef\_)

print (model.coef\_.shape)

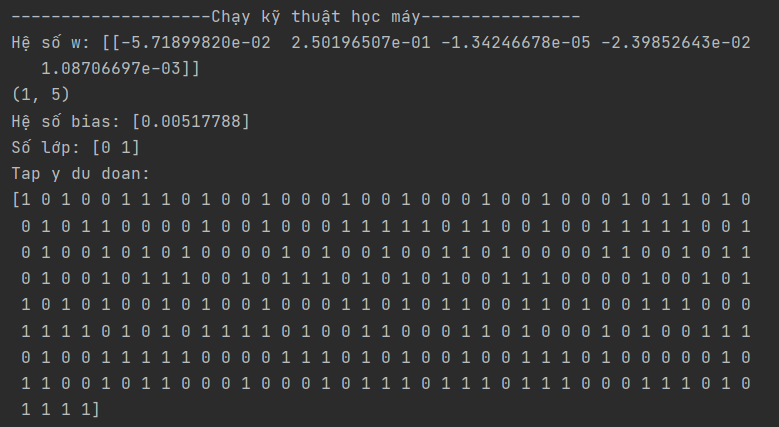
print ("Hệ số bias:", model.intercept\_)

print ("Số lớp:", model.classes\_)

#print ("Số vòng lặp", log\_model.n\_iter\_)

print ("Tap y du doan:")

print (y\_pred)

****

### 1.5. Phương pháp đánh giá độ chính xác

Phương pháp đánh giá độ chính xác được sử dụng trong bài toán này là Accuracy và Confusion Matrix.

**Accuracy:** Cách đánh giá này tính tỉ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử. Giả sử, ta đếm được 7 điểm dữ liệu được dự đoán đúng trên tổng số 10 điểm. Như vậy, Accuracy Score của bài toán sẽ là 0.7 (hay 70%).

**Confusion Matrix:** Cách tính sử dụng Accuracy như ở trên chỉ cho chúng ta biết được bao nhiêu phần trăm lượng dữ liệu được phân loại đúng mà không chỉ ra được cụ thể mỗi loại được phân loại như thế nào, lớp nào được phân loại đúng nhiều nhất, và dữ liệu thuộc lớp nào thường bị phân loại nhầm vào lớp khác. Để có thể đánh giá được các giá trị này, chúng ta sử dụng một ma trận được gọi là confusion matrix. Về cơ bản, confusion matrix thể hiện có bao nhiêu điểm dữ liệu thực sự thuộc vào một class, và được dự đoán là rơi vào một class.

**Code và kết quả:**

print ('------------------Đánh giá mô hình học máy----------------')

#Bước 4: Đánh giá mô hình học dựa trên kết quả dự đoán

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

accuracy = accuracy\_score(y\_test, y\_pred)

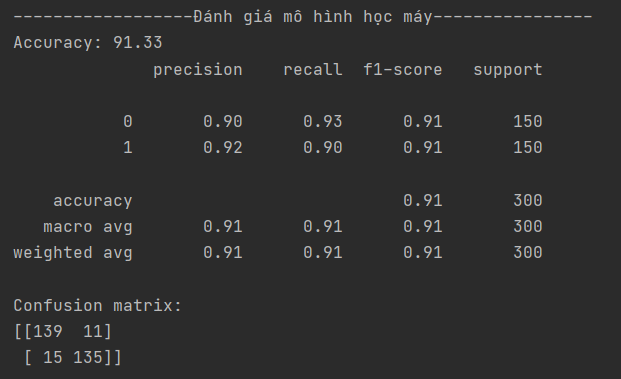
print ('Accuracy: %.2f' % (accuracy \* 100))

print(classification\_report(y\_test,y\_pred))

cnf\_matrix = confusion\_matrix(y\_test, y\_pred)

print ('Confusion matrix:')

print(cnf\_matrix)



### 1.6. Kết luận

Logistic Regression chạy khá nhanh và chuẩn xác, tuy nhiên nhược điểm của nó cũng như các thuật toán phân lớp khác như SVM, DecisionTreeClassifier, Random Forest là rất nhạy cảm với dữ liệu mất cân bằng. Kết quả khi chạy có thể rất đẹp nhưng mô hình dự đoán thường không đáng tin cậy. Chính vì thế, để đảm bảo dữ liệu được huấn luyện tốt và cho ra dự đoán chính xác, cần tiến hành xử lý mất cân bằng ngay từ khâu ban đầu trước khi chạy mô hình dự đoán kết quả.

**2. Sử dụng kỹ thuật học máy K-Means để giải bài toán phân cụm hạt giống lúa mì**

**2.1 Mô tả bộ dữ liệu**

**Nguồn gốc:** Piotr Kulczycki, Piotr A. Kowalski, Szymon Lukasik, Khoa Công nghệ Thông tin và Điều khiển Tự động Slawomir Zak, Đại học Công nghệ Cracow, Warszawska 24, PL 31-155 Cracow, Ba Lan và Viện Nghiên cứu Hệ thống, Học viện Khoa học Ba Lan, Newelska 6, PL 01 -447 Warsaw, Ba Lan  
**E-mail:** {kulczycki, pakowal, slukasik, slzak}@ ibspan.waw.pl  
Hồ sơ về nấm được rút ra từ Hướng dẫn thực địa về nấm ở Bắc Mỹ của Hiệp hội Audubon (1981). GH Lincoff (Pres.), New York: Alfred A. Knopf  
**Nhà tài trợ:** Jeff Schlimmer ( Jeffrey.Schlimmer **'@'** a.gp.cs.cmu.edu )

**Thông tin tập dữ liệu:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Đặc điểm của Tập dữ liệu:** | Đa biến | **Số phiên bản:** | 210 | **Khu vực:** | Đời sống |
| **Đặc điểm thuộc tính:** | Thực tế | **Số thuộc tính:** | 7 | **Ngày được tặng** | 2012-09-29 |
| **Nhiệm vụ liên quan:** | Phân loại, phân cụm | **Giá trị bị mất?** | N/A | **Số lượt truy cập web:** | 332411 |

Nhóm được kiểm tra bao gồm các hạt nhân thuộc ba giống lúa mì khác nhau: Kama, Rosa và Canadian, mỗi loại 70 nguyên tố, được chọn ngẫu nhiên cho  
thí nghiệm. Hình ảnh chất lượng cao của cấu trúc nhân bên trong đã được phát hiện bằng kỹ thuật tia X mềm. Nó không phá hủy và rẻ hơn đáng kể so với các kỹ thuật hình ảnh phức tạp khác như kính hiển vi quét hoặc công nghệ laser. Hình ảnh được ghi lại trên tấm KODAK X-quang 13x18 cm. Các nghiên cứu được thực hiện bằng cách sử dụng kết hợp hạt lúa mì thu hoạch có nguồn gốc từ các cánh đồng thí nghiệm, được khám phá tại Viện Vật lý Nông học của Viện Hàn lâm Khoa học Ba Lan.   
 Tập dữ liệu có thể được sử dụng cho các nhiệm vụ phân loại và phân tích cụm.

**Thông tin thuộc tính:**

Để xây dựng dữ liệu, người ta đo bảy thông số hình học của hạt lúa mì:  
1. diện tích A,  
2. chu vi P,  
3. độ nén C = 4 \* pi \* A / P ^ 2,  
4. chiều dài của nhân,  
5. chiều rộng của nhân ,  
6. hệ số không đối xứng  
7. chiều dài của rãnh nhân.  
Tất cả các tham số này đều có giá trị thực liên tục.

**2.2 Mô tả bài toán**

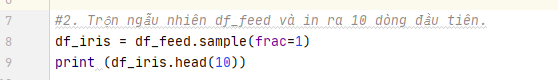
Bài toán sử dụng bộ dữ liệu các thuộc tính vật lý của hạt giống lúa mì. Áp dụng ký thuật K-Means Claustering

### 2.3 Mô tả chi tiết về cách xử lý dữ liệu

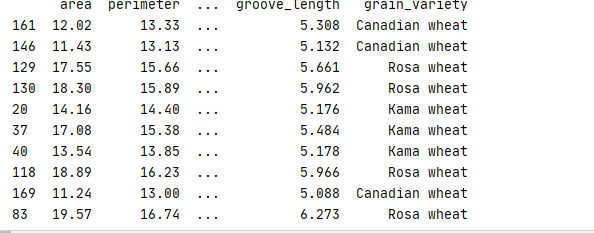
Đọc file dữ liệu seeds.csv

https://lh6.googleusercontent.com/YIQ58gCJr293nuMKQP6PAjGLfGXGPUJrWQzBgzw3UOYRGR4itQZn5rlkTpRIC7HkOKluPCF4EU2xToHgZr09i7hVENpN6EJ9Ej2iD8tjKeVAc7KZoL0U7YyzSrB535_6OTG4CUQ

Trộn bộ df\_feed và in ra 10 dòng đầu tiên:



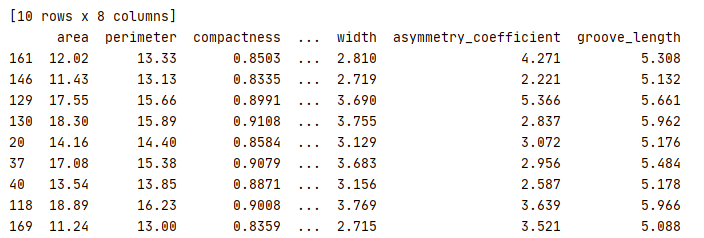
Kết quả:



Gán X là 7 cột (area , perimeter, compactness, length, width, asymmetry\_coefficent, groove\_length) và hiển thị 10 hàng đầu:

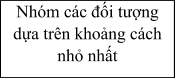
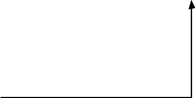
https://lh5.googleusercontent.com/rWbmrRkHxuha70qsX_nD2m7y3Vo7VFJ53UrzCnKMDIIShYE29xY0H1-qWf_veWcdjb8M11_a0DWgoCZx99pljIQZSQa7CjJ_p5H_RDMfeMWJ2CrsLSxcU5bJswLFWt558TW9jtU

Kết quả:



### 2.4 Mô tả về K-Means

***Thuật toán K-Means được mô tả như sau:***

https://docs.google.com/drawings/u/0/d/s2x1p72hzSgdFehmtjcF7fQ/image?w=107&h=42&rev=1&ac=1&parent=1D3lQCGJRBwLkDdc8zBt--oTIc9WTnOpV  
https://docs.google.com/drawings/u/0/d/sket4IOzPJR9nGkAfeXuoSg/image?w=3&h=23&rev=1&ac=1&parent=1D3lQCGJRBwLkDdc8zBt--oTIc9WTnOpV  
https://docs.google.com/drawings/u/0/d/s2wdSDRCpP7DzBaQLtnZxUw/image?w=129&h=52&rev=1&ac=1&parent=1D3lQCGJRBwLkDdc8zBt--oTIc9WTnOpV  
  
https://docs.google.com/drawings/u/0/d/s9e_UmQXo_NaWBdIU7rFfng/image?w=3&h=47&rev=1&ac=1&parent=1D3lQCGJRBwLkDdc8zBt--oTIc9WTnOpVhttps://docs.google.com/drawings/u/0/d/sS42p7RPbXHWq0r1fmpg_vw/image?w=204&h=41&rev=1&ac=1&parent=1D3lQCGJRBwLkDdc8zBt--oTIc9WTnOpV  
https://docs.google.com/drawings/u/0/d/st_EtGAWjcEmtUB7OjArjHw/image?w=175&h=50&rev=1&ac=1&parent=1D3lQCGJRBwLkDdc8zBt--oTIc9WTnOpVhttps://docs.google.com/drawings/u/0/d/sFjRgEfQPVq3c09kG0XH5LA/image?w=39&h=29&rev=1&ac=1&parent=1D3lQCGJRBwLkDdc8zBt--oTIc9WTnOpV  
  
https://docs.google.com/drawings/u/0/d/s7Zx4C2gRCb8aEupW-qFK1g/image?w=3&h=41&rev=1&ac=1&parent=1D3lQCGJRBwLkDdc8zBt--oTIc9WTnOpV  
https://docs.google.com/drawings/u/0/d/sHZGAv1W-tb7vAwcl9M3jVQ/image?w=175&h=55&rev=1&ac=1&parent=1D3lQCGJRBwLkDdc8zBt--oTIc9WTnOpVhttps://docs.google.com/drawings/u/0/d/sZVbaAqscRNw0STas4i_EwA/image?w=122&h=45&rev=1&ac=1&parent=1D3lQCGJRBwLkDdc8zBt--oTIc9WTnOpVhttps://docs.google.com/drawings/u/0/d/sGpfeQh9U15_r5u2xcFzmUQ/image?w=3&h=44&rev=1&ac=1&parent=1D3lQCGJRBwLkDdc8zBt--oTIc9WTnOpVhttps://docs.google.com/drawings/u/0/d/s3iauoWqbNkWOiDHHDtK5rg/image?w=39&h=29&rev=1&ac=1&parent=1D3lQCGJRBwLkDdc8zBt--oTIc9WTnOpV  
https://docs.google.com/drawings/u/0/d/smTGDmMiA582SQTmK6YDt4w/image?w=1&h=33&rev=1&ac=1&parent=1D3lQCGJRBwLkDdc8zBt--oTIc9WTnOpVhttps://docs.google.com/drawings/u/0/d/shyrP7BWVL39yAiO5Oep81w/image?w=34&h=3&rev=1&ac=1&parent=1D3lQCGJRBwLkDdc8zBt--oTIc9WTnOpV  
  
  


 Hình 1. Thuật toán K-means

* ***Thuật toán K-Means thực hiện qua các bước chính sau:***

1.    Chọn ngẫu nhiên K tâm (centroid) cho K cụm (cluster). Mỗi cụm được đại diện bằng các tâm của cụm.

2.    Tính khoảng cách giữa các đối tượng (objects) đến K tâm (thường dùng khoảng cách Euclidean)

3.    Nhóm các đối tượng vào nhóm gần nhất

4.    Xác định lại tâm mới cho các nhóm

5.    Thực hiện lại bước 2 cho đến khi không có sự thay đổi nhóm nào của các đối tượng

Thuật toán k-means có thể được chia thành các bước như sau:

**Bước 1:** Tạo các trung tâm ngẫu nhiên

https://upload.wikimedia.org/wikipedia/commons/thumb/c/c6/Random_centers.png/220px-Random_centers.png

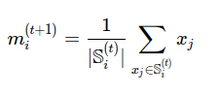
**Bước 2:** Gán các điểm dữ liệu vào các cụm

Với mỗi điểm dữ liệu, ta sẽ tính khoảng cách của nó tới các trung tâm (bằng [Khoảng cách Euclid](https://vi.wikipedia.org/wiki/Kho%E1%BA%A3ng_c%C3%A1ch_Euclid)). Ta sẽ gán chúng vào trung tâm gần nhất. Tập hợp các điểm được gán vào cùng 1 trung tâm sẽ tạo thành cụm.

https://upload.wikimedia.org/wikipedia/commons/thumb/a/a3/Clustering_point.png/414px-Clustering_point.png

**Bước 3:** Cập nhật trung tâm

Với mỗi cụm đã tìm được ở bước 2, trung tâm mới sẽ là [trung bình cộng](https://vi.wikipedia.org/wiki/Trung_b%C3%ACnh_c%E1%BB%99ng_%C4%91%C6%A1n_gi%E1%BA%A3n) của các điểm dữ liệu trong cụm đó.

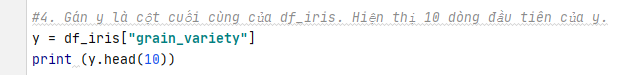


Thuật toán sẽ lặp lại các bước trên cho tới khi đạt được kết quả chấp nhận được.

Ngắn gọn phương pháp k-means là 1 trong số những thuật toán phân cụm (clustering). Đầu vào tập dữ liệu cần phân cụm và số cụm (cluster), đầu ra chúng ta sẽ được kết quả dữ liệu đã được phân về các cluster.

Mục đích là phân dữ liệu thành các cụm (cluster) khác nhau sao cho dữ liệu trong cùng một cụm có tính chất giống nhau.

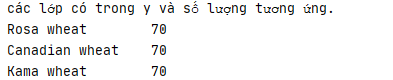
Gán y là cột cuối cùng của df\_iris.



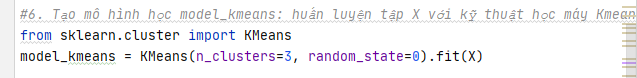
Hiện thị các lớp có trong y và số lượng tương ứng

https://lh4.googleusercontent.com/TsFIWQQet_bWGggySlmt9h-qo_glXp58jDnUzbDn8WXBSA-cPYi13rdwkj3ERX4ZgY-OY8aJTY7WQyIAecUEJYCiF1mnW81KS2g263EMJnq_lrw9cgop7OSLaCTA26VbDTkFC6A

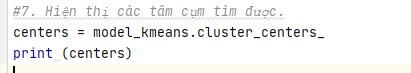
Kết quả:



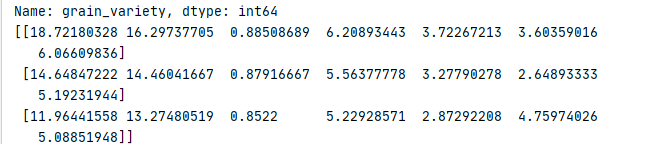
Tạo mô hình học model\_kmeans: huấn luyện tập X với kỹ thuật học máy Kmeans (số cụm = 3).



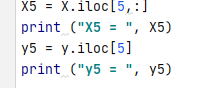
Hiện thị các tâm cụm tìm được.



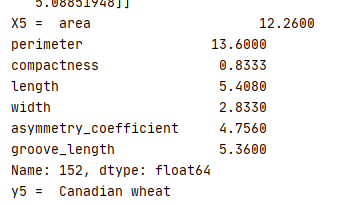
Kết quả:



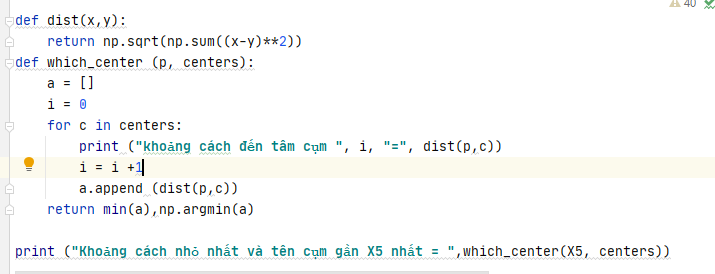
Hiện thị điểm dữ liệu X5 (dòng số 5 của X) và nhãn y5 tương ứng.



Kết quả:



Tìm cụm gần điểm X5 nhất.



Kết quả:

