# LỜI MỞ ĐẦU

Học máy (Machine Learning) là một lĩnh vực của trí tuệ nhân tạo (Artificial Intelligence - AI). Các thuật toán học máy cho phép máy tính đào tạo đầu vào dữ liệu và sử dụng phân tích thống kê để đưa ra các giá trị nằm trong một phạm vi cụ thể.

Ngày nay, những người sử dụng công nghệ đều được hưởng lợi từ việc học máy. Công nghệ nhận diện khuôn mặt giúp người dùng gắn thẻ và chia sẻ ảnh của bạn bè. Công nghệ nhận dạng ký tự quang học (OCR) chuyển đổi hình ảnh văn bản sang dạng di chuyển. Khi mà khả năng tính toán của máy tính được nâng lên một tầm cao mới cùng với lượng dữ liệu khổng lồ được thu thập, Machine Learning đã tiến thêm một bước dài và Deep Learning (DL) một lĩnh vực mới được ra đời.

Deep Learning được xây dựng từ mạng nơ ron sinh học và bao gồm nhiều lớp trong mạng nơ ron nhân tạo được tạo thành từ phần cứng và GPU. Deep Learning sử dụng một tầng các lớp đơn vị xử lý phi tuyến để trích xuất hoặc chuyển đổi các tính năng (hoặc biểu diễn) của dữ liệu. Đầu ra của một lớp phục vụ như là đầu vào của lớp kế tiếp. Deep learning tập trung giải quyết các vấn đề liên quan đến mạng thần kinh nhân tạo nhằm nâng cấp các công nghệ như nhận diện giọng nói, dịch tự động (machine translation), xử lý ngôn ngữ tự nhiên…

Trong số các thuật toán học máy hiện đang được sử dụng và phát triển, học sâu thu hút được nhiều nghiên cứu nhất và có thể đánh bại con người trong một số nhiệm vụ nhận thức. Do những đặc tính nổi bật và kết quả tối ưu, học tập sâu đã trở thành phương pháp tiếp cận được nghiên cứu và ứng dụng trong giải quyết nhiều bài toán thuộc lĩnh vực trí tuệ nhân tạo.

Chính vì vậy bọn em chọn đề tài:” Kỹ thuật LSTM, cài đặt của tensorflow và ứng dụng”….

# CHƯƠNG 1: KỸ THUẬT LSTM, CÀI ĐẶT CỦA TENSORFLOW

**1.Kỹ thuật LSTM**

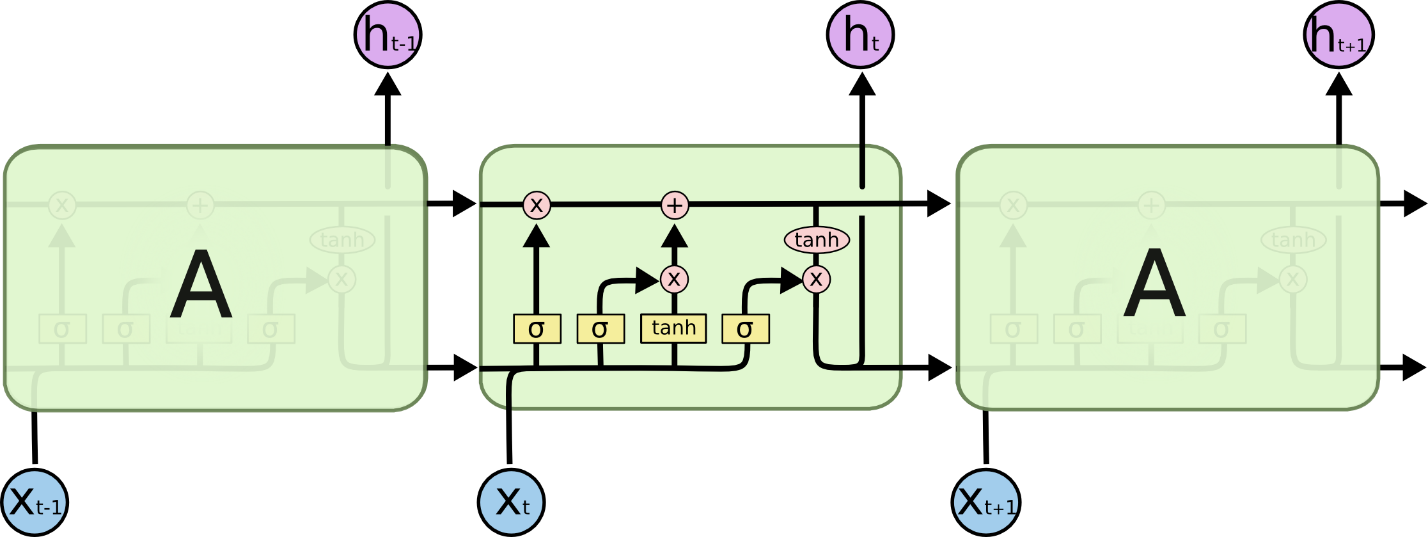
**1.1.Mạng LSTM**

Mạng bộ nhớ dài-ngắn (Long Short Term Memory networks), thường được gọi là LSTM - là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi [Hochreiter & Schmidhuber (1997)](http://deeplearning.cs.cmu.edu/pdfs/Hochreiter97_lstm.pdf), và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay.

LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-dun này có cấu trúc rất đơn giản, thường là một tầng *tanh*.

LSTM cũng có kiến trúc dạng chuỗi như vậy, nhưng các mô-đun trong nó có cấu trúc khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng nơ-ron, chúng có tới 4 tầng tương tác với nhau một cách rất đặc biệt.



## 1.2. Ý tưởng cốt lõi của LSTM

## Chìa khóa của LSTM là trạng thái tế bào (cell state) - chính đường chạy thông ngang phía trên của sơ đồ hình vẽ.

## Trạng thái tế bào là một dạng giống như băng truyền. Nó chạy xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi.

## https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM3-C-line.png

LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate).

Các cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân.

## https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM3-gate.png

Tầng sigmoid sẽ cho đầu ra là một số trong khoản [0, 1][0,1], mô tả có bao nhiêu thông tin có thể được thông qua. Khi đầu ra là 00 thì có nghĩa là không cho thông tin nào qua cả, còn khi là 11 thì có nghĩa là cho tất cả các thông tin đi qua nó.

Một LSTM gồm có 3 cổng như vậy để duy trì và điều hành trạng thái của tế bào.

**2.Cài đặt của Tensorflow và ứng dụng**

**2.1. Tensorflow là gì?**

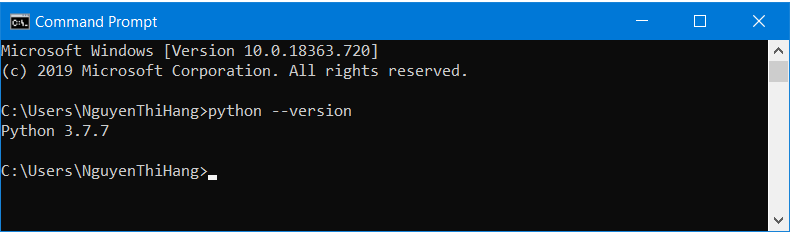
Tensorflow là một thư viện mã nguồn mở mạnh mẽ cho machine learning được phát triển bởi các nhà cứu của Google. Thư viên này có rất nhiều các hàm được xây dựng sẵn cho từng bài toán cho phép xây dựng nhiều mạng neural network khác nhau. Tensorflow cũng cho phép tính toán song song trên nhiều máy tính khác nhau, tất nhiên là cũng có thể trên nhiều CPU, GPU trong cùng một máy. Tensorflow cung cấp các api làm việc với Python, C++. Tại tutorial này, tôi sẽ sử dụng Python.

Tutorial này sẽ trình bày lý thuyết và cách sử dụng thư viện Tensorflow phục vụ bài toán deep learning. Các khái niệm cơ bản trong tensorflow, cách tự xây dựng một mô hình học máy sử dụng tensorflow từ mô hình đơn giản như linear regressioin đến các mô hình CNN, RNN. Giải quyết các bài toán như word embedding, dịch máy, nhận dạng chữ viết, học tăng cường.

**2.2. Cài đặt**

Ở mục này, mình sẽ cài đặt **Tensorflow** với ngôn ngữ lập trình là **Python**. Trước khi cài đặt thì mình kiểm cần phải kiểm tra version của **Python**hiện tại của PC của mình bằng câu lệnh đơn giản bên dưới nhé.





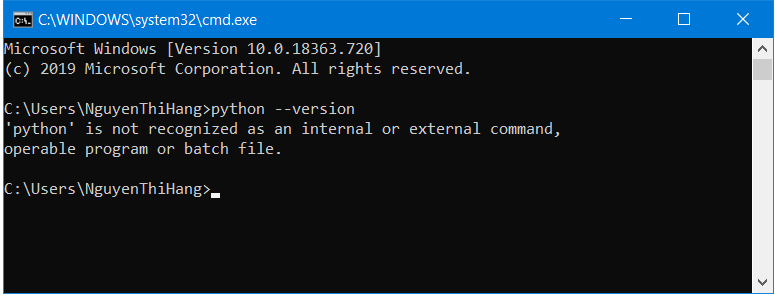
Theo như kết quả hiển thị trên màn hình thì PC của mình đang sử dụng **Python 3** có version là **3.7.7**.

Ghi chú: Ở thời điểm viết blog thì **Tensorflow 2.1.0** chỉ hỗ trợ cho **Python3** từ 3.5 -> 3.7. Nếu trong PC của bạn đã có **Python** nhưng khác phiên bản thì sẽ không cài đặt được **Tensorflow 2.1.0** và sẽ nhận được thông báo lỗi như sau:

ERROR: Could not find a version that satisfies the requirement tensorflow (from versions: none) ERROR: No matching distribution found for tensorflow

Bạn có thể fix lỗi đó bằng cách cài lại **Python 3.5.x** hoặc**3.6.x** hoặc **3.7.x.**!

Nếu khi gõ lệnh python –version mà nhận được lỗi như hình bên dưới



Theo như lỗi thông báo ở trên thì PC có thể chưa có cài đặt Python hoặc biến môi trường của Python thiết lập chưa đúng.

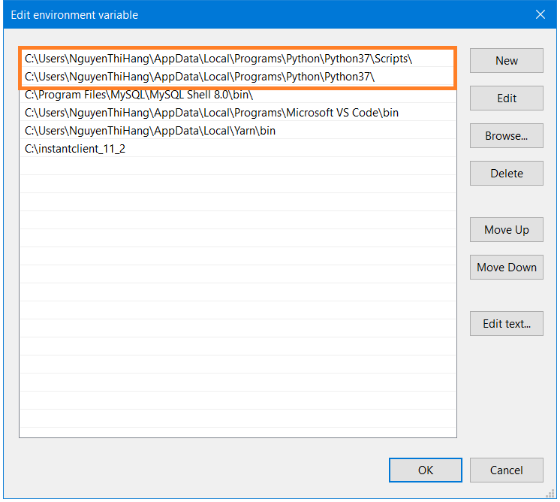
### **Trường hợp chưa cài đặt Python**

Bạn vào trang chủ của [**Python**](https://www.python.org/downloads/)và chọn phiên bản[**Python**](https://www.python.org/downloads/)[**2x**](https://www.python.org/downloads/)hoặc [**3x**](https://www.python.org/downloads/)để tải về và cài đặt. Theo quan điểm cá nhân của mình thì nên xài [**Python**3x](https://www.python.org/downloads/) do nó tương thích nhiều với những **framework**mới. Còn nếu bạn vẫn muốn làm việc với những **framework**cũ thì [**Python 2x**](https://www.python.org/downloads/) sẽ là sự lựa chọn.



Sau khi cài đặt xong rồi thì bạn mở mới **command line** và gõ lệnh **python –version**lại nhé. Nếu vẫn bị lỗi như hình 2 thì bạn thử restart PC và kiểm tra lại. Nếu vẫn còn lỗi thì xem bước tiếp theo ở bên dưới nhé.

* **Trường hợp đã cài đặt nhưng vẫn bị lỗi**
* Lúc này bạn nhấp phải **My Computer** chọn **Properties**(hoặc vào **Control Panel / System and Security / System**).
* Nhìn về phía bên trái chọn **Advances System Settings**
* Một Popup mới hiện ra. Click vào button **Environment Variables ….**
* Trong group **User variables for {x}**. Bạn chú ý đến **Path**. Hãy Click chọn **Path**và Click vào button **Edit**.
* Bạn Click **New**và thêm giá trị cho nó chính là đường dẫn thư mục cài đặt của Python. Theo hình hình bến dưới thì mình đã cài Python 3x vào thư mục *C:UsersAppDataLocalProgramsPythonPython37*



Xong rồi, click vào button **Ok** để thiết lập của bạn mới có giá trị. Sau khi thiết lập xong thì bạn mở mới **command line** khác và kiểm tra lại việc thiết lập bằng lệnh python –version. Nếu vẫn còn lỗi thì hãy restart PC lại để refresh biến môi trường vừa mới thiết lập nhé.

Sau khi đã cài đặt được **Python** đúng phiên bản thì mình đã hoàn thành được **80%** việc cài đặt Tensorflow rồi đó. Công việc còn lại là cài đặt thêm một số thư viện liên quan và Tensorflow thôi. Chúng ta tiếp tục nào !!

### **2.2.1. Upgrade pip và virtualenv**

Tiếp đến mình cần upgrade thư viện Pip và virtualenv trong python lên phiên bản mới nhất. Tensorflow cần **pip >=19.0** và **virtualenv**



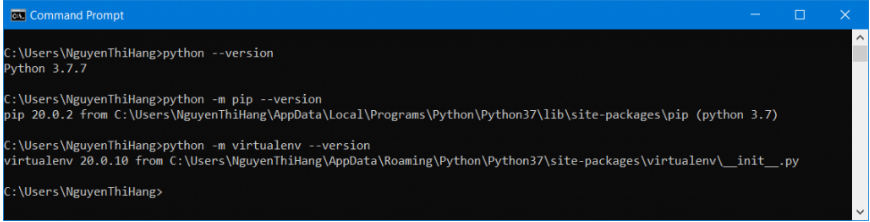
Ghi chú: Nếu bạn nhận được thông báo lỗi :

ERROR: Could not install packages due to an EnvironmentError: [WinError 5] Access is denied: ‘C:xxxxpip.exe’  
Consider using the --user option or check the permissions.

Thì bạn thêm **–user** vào câu lệnh bên trên hoặc vào thư mục bị **permissions**– rồi thay đổi quyền cho nó nhé.



Sau khi upgrade pip và virtualenv thành công, mình kiếm tra phiên bản của nó.



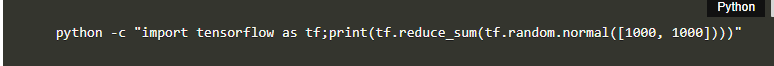
Cuối cùng thì mình sẽ cài đặt Tensorflow. Tensorflow có 3 phiên bản

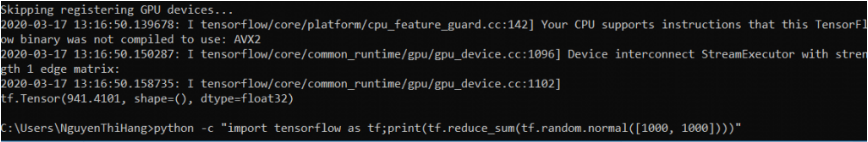
* tensorflow —Phiên bản mới nhất – hiện là 2x
* tf-nightly —Phiên bản đang phát triển.
* tensorflow==1.15 —Chỉ định phiên bản mà mình muốn cài đặt. Ở đây là phiên bản TensorFlow 1.15.

Ok. Ở đây mình cài đặt phiên bản mới nhất.



Sau khi cài đặt Tensorflow thành công. Mình cần kiểm tra lại xem nó có cài đặt đúng hay chưa?. Từ command line. Mình chay đoạn script sau:

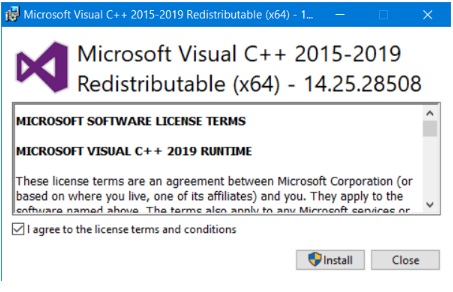




Nếu không đạt được kết quả như trên mà xuất hiện lỗi:

….ImportError: Could not find the DLL(s) ‘msvcp140\_1.dll’. TensorFlow requires that these DLLs be installed in a directory that is named in your %PATH% environment variable. You may install these DLLs by downloading “Microsoft C++ Redistributable for Visual Studio 2015, 2017 and 2019” for your platform from this URL: https://support.microsoft.com/help/2977003/the-latest-supported-visual-c-downloads..

Thì bạn cần cài đặt thêm thư viện [*Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017, and 2019*](https://support.microsoft.com/en-us/help/2977003/the-latest-supported-visual-c-downloads) nữa là ok nhé.



Chúc mừng bạn đã cài đặt thành công **Tensorflow 2.1.0** trên **Window 10** với ngôn ngữ lập trình phổ biến là **Python**với phiên bản **3.7.7**.

# CHƯƠNG 2: THUẬT TOÁN SVM,KNN TRONG PHÂN LOẠI RƯỢU

## 2.1 .Bộ dữ liệu về rượu

## 2.1.1.Giới thiệu

Tập dữ liệu rượu là tập dữ liệu đa biến được giới thiệu bởi nhà thống kê và nhà khoa học người Anh Ronald Fisher trong bài báo năm 1936 Việc sử dụng nhiều phép đo trong các vấn đề phân loại như một ví dụ về phân tích phân biệt tuyến tính. Đôi khi nó được gọi là tập dữ liệu rượu của Anderson vì Edgar Anderson đã thu thập dữ liệu để định lượng sự thay đổi trạng thái của ba loại rượu liên quan.

Bộ dữ liệu bao gồm 60 mẫu từ mỗi ba loại rượu (Rượu whisky, Rượu vodka và Rượu Cocktail). Bốn đặc điểm được đo từ mỗi mẫu: Nồng đọ axit tataric, nồng độ Ph, nồng độ Clo, nồng độ Alc,... Dựa trên sự kết hợp của bốn tính năng này, Fisher đã phát triển một mô hình phân biệt tuyến tính để phân biệt các loại với nhau.

### **2.1.2. Tập dữ liệu**

Bộ dữ liệu chứa một bộ 178 bản ghi bao gồm các thuộc tính – nồng độ axit tataric, Ph, Clo, Alc.

*Bảng 1: Thông tin loại rượu Whisky*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thứ tự | Nồng độ axit tataric | Nồng độ Ph | Nồng độ Clo | Nồng độ Alc |
| 1 | 5.1 | 3.5 | 1.4 | 0.2 |
| 2 | 4.9 | 3.0 | 1.4 | 0.2 |
| 3 | 4.7 | 3.2 | 1.3 | 0.2 |
| 4 | 4.6 | 3.1 | 1.5 | 0.2 |
| 5 | 5.0 | 3.6 | 1.4 | 0.3 |
| 6 | 5.4 | 3.9 | 1.7 | 0.4 |
| 7 | 4.6 | 3.4 | 1.4 | 0.3 |
| 8 | 5.0 | 3.4 | 1.5 | 0.2 |
| 9 | 4.4 | 2.9 | 1.4 | 0.2 |
| 10 | 4.9 | 3.1 | 1.5 | 0.1 |
| 11 | 5.4 | 3.7 | 1.5 | 0.2 |
| 12 | 4.8 | 3.4 | 1.6 | 0.2 |
| 13 | 4.8 | 3.0 | 1.4 | 0.1 |
| 14 | 4.3 | 3.0 | 1.1 | 0.1 |
| 15 | 5.8 | 4.0 | 1.2 | 0.2 |
| 16 | 5.7 | 4.4 | 1.5 | 0.4 |
| 17 | 5.4 | 3.9 | 1.3 | 0.4 |
| 18 | 5.1 | 3.5 | 1.4 | 0.3 |
| 19 | 5.7 | 3.8 | 1.7 | 0.3 |
| 20 | 5.1 | 3.8 | 1.5 | 0.3 |
| 21 | 5.4 | 3.4 | 1.7 | 0.2 |
| 22 | 5.1 | 3.7 | 1.5 | 0.4 |
| 23 | 4.6 | 3.6 | 1.0 | 0.2 |
| 24 | 5.1 | 3.3 | 1.7 | 0.5 |
| 25 | 4.8 | 3.4 | 1.9 | 0.2 |
| 26 | 5.0 | 3.0 | 1.6 | 0.2 |
| 27 | 5.0 | 3.4 | 1.6 | 0.4 |
| 28 | 5.2 | 3.5 | 1.5 | 0.2 |
| 29 | 5.2 | 3.4 | 1.4 | 0.2 |
| 30 | 4.7 | 3.2 | 1.6 | 0.2 |
| 31 | 4.8 | 3.1 | 1.6 | 0.2 |
| 32 | 5.4 | 3.4 | 1.5 | 0.4 |
| 33 | 5.2 | 4.1 | 1.5 | 0.1 |
| 34 | 5.5 | 4.2 | 1.4 | 0.2 |
| 35 | 4.9 | 3.1 | 1.5 | 0.2 |
| 36 | 5.0 | 3.2 | 1.2 | 0.2 |
| 37 | 5.5 | 3.5 | 1.3 | 0.2 |
| 38 | 4.9 | 3.6 | 1.4 | 0.1 |
| 39 | 4.4 | 3.0 | 1.3 | 0.2 |
| 40 | 5.1 | 3.4 | 1.5 | 0.2 |
| 41 | 5.0 | 3.5 | 1.3 | 0.3 |
| 42 | 4.5 | 2.3 | 1.3 | 0.3 |
| 43 | 4.4 | 3.2 | 1.3 | 0.2 |
| 44 | 5.0 | 3.5 | 1.6 | 0.6 |
| 45 | 5.1 | 3.8 | 1.9 | 0.4 |
| 46 | 4.8 | 3.0 | 1.4 | 0.3 |
| 47 | 5.1 | 3.8 | 1.6 | 0.2 |
| 48 | 4.6 | 3.2 | 1.4 | 0.2 |
| 49 | 5.3 | 3.7 | 1.5 | 0.2 |
| 50 | 5.0 | 3.3 | 1.4 | 0.2 |

*Bảng 2: Thông tin rượu Vodka*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thứ tự | Nồng độ axit tataric | Nồng độ Ph | Nồng độ Clo | Nồng độ Alc |
| 1 | 7.0 | 3.2 | 4.7 | 1.4 |
| 2 | 6.4 | 3.2 | 4.5 | 1.5 |
| 3 | 6.9 | 3.1 | 4.9 | 1.5 |
| 4 | 5.5 | 2.3 | 4.0 | 1.3 |
| 5 | 6.5 | 2.8 | 4.6 | 1.5 |
| 6 | 5.7 | 2.8 | 4.5 | 1.3 |
| 7 | 6.3 | 3.3 | 4.7 | 1.6 |
| 8 | 4.9 | 2.4 | 3.3 | 1.0 |
| 9 | 6.6 | 2.9 | 4.6 | 1.3 |
| 10 | 5.2 | 2.7 | 3.9 | 1.4 |
| 11 | 5.0 | 2.0 | 3.5 | 1.0 |
| 12 | 5.9 | 3.0 | 4.2 | 1.5 |
| 13 | 6.0 | 2.2 | 4.0 | 1.0 |
| 14 | 6.1 | 2.9 | 4.7 | 1.4 |
| 15 | 5.6 | 2.9 | 3.6 | 1.3 |
| 16 | 6.7 | 3.1 | 4.4 | 1.4 |
| 17 | 5.6 | 3.0 | 4.5 | 1.5 |
| 18 | 5.8 | 2.7 | 4.1 | 1.0 |
| 19 | 6.2 | 2.2 | 4.5 | 1.5 |
| 20 | 5.6 | 2.5 | 3.9 | 1.1 |
| 21 | 5.9 | 3.2 | 4.8 | 1.8 |
| 22 | 6.1 | 2.8 | 4.0 | 1.3 |
| 23 | 6.3 | 2.5 | 4.9 | 1.5 |
| 24 | 6.1 | 2.8 | 4.7 | 1.2 |
| 25 | 6.4 | 2.9 | 4.3 | 1.3 |
| 26 | 6.6 | 3.0 | 4.4 | 1.4 |
| 27 | 6.8 | 2.8 | 4.8 | 1.4 |
| 28 | 6.7 | 3.0 | 5.0 | 1.7 |
| 29 | 6.0 | 2.9 | 4.5 | 1.5 |
| 30 | 5.7 | 2.6 | 3.5 | 1.0 |
| 31 | 5.5 | 2.4 | 3.8 | 1.1 |
| 32 | 5.5 | 2.4 | 3.7 | 1.0 |
| 33 | 5.8 | 2.7 | 3.9 | 1.2 |
| 34 | 6.0 | 2.7 | 5.1 | 1.6 |
| 35 | 5.4 | 3.0 | 4.5 | 1.5 |
| 36 | 6.0 | 3.4 | 4.5 | 1.6 |
| 37 | 6.7 | 3.1 | 4.7 | 1.5 |
| 38 | 6.3 | 2.3 | 4.4 | 1.3 |
| 39 | 5.6 | 3.0 | 4.1 | 1.3 |
| 40 | 5.5 | 2.5 | 4.0 | 1.3 |
| 41 | 5.5 | 2.6 | 4.4 | 1.2 |
| 42 | 6.1 | 3.0 | 4.6 | 1.4 |
| 43 | 5.8 | 2.6 | 4.0 | 1.2 |
| 44 | 5.0 | 2.3 | 3.3 | 1.0 |
| 45 | 5.6 | 2.7 | 4.2 | 1.3 |
| 46 | 5.7 | 3.0 | 4.2 | 1.2 |
| 47 | 5.7 | 2.9 | 4.2 | 1.3 |
| 48 | 6.2 | 2.9 | 4.3 | 1.3 |
| 49 | 5.1 | 2.5 | 3.0 | 1.1 |
| 50 | 5.7 | 2.8 | 4.1 | 1.3 |

*Bảng 3: Thông tin rượu Cocktail*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thứ tự | Nồng độ axit tataric | Nồng độ Ph | Nồng độ Clo | Nồng độ Alc |
| 1 | 6.3 | 3.3 | 6.0 | 2.5 |
| 2 | 5.8 | 2.7 | 5.1 | 1.9 |
| 3 | 7.1 | 3.0 | 5.9 | 2.1 |
| 4 | 6.3 | 2.9 | 5.6 | 1.8 |
| 5 | 6.5 | 3.0 | 5.8 | 2.2 |
| 6 | 7.6 | 3.0 | 6.6 | 2.1 |
| 7 | 4.9 | 2.5 | 4.5 | 1.7 |
| 8 | 7.3 | 2.9 | 6.3 | 1.8 |
| 9 | 6.7 | 2.5 | 5.8 | 1.8 |
| 10 | 7.2 | 3.6 | 6.1 | 2.5 |
| 11 | 6.5 | 3.2 | 5.1 | 2.0 |
| 12 | 6.4 | 2.7 | 5.3 | 1.9 |
| 13 | 6.8 | 3.0 | 5.5 | 2.1 |
| 14 | 5.7 | 2.5 | 5.0 | 2.0 |
| 15 | 5.8 | 2.8 | 5.1 | 2.4 |
| 16 | 6.4 | 3.2 | 5.3 | 2.3 |
| 17 | 6.5 | 3.0 | 5.5 | 1.8 |
| 18 | 7.7 | 3.8 | 6.7 | 2.2 |
| 19 | 7.7 | 2.6 | 6.9 | 2.3 |
| 20 | 6.0 | 2.2 | 5.0 | 1.5 |
| 21 | 6.9 | 3.2 | 5.7 | 2.3 |
| 22 | 5.6 | 2.8 | 4.9 | 2.0 |
| 23 | 7.7 | 2.8 | 6.7 | 2.0 |
| 24 | 6.3 | 2.7 | 4.9 | 1.8 |
| 25 | 6.7 | 3.3 | 5.7 | 2.1 |
| 26 | 7.2 | 3.2 | 6.0 | 1.8 |
| 27 | 6.2 | 2.8 | 4.8 | 1.8 |
| 28 | 6.1 | 3.0 | 4.9 | 1.8 |
| 29 | 6.4 | 2.8 | 5.6 | 2.1 |
| 30 | 7.2 | 3.0 | 5.8 | 1.6 |
| 31 | 7.4 | 2.8 | 6.1 | 1.9 |
| 32 | 7.9 | 3.8 | 6.4 | 2.0 |
| 33 | 6.4 | 2.8 | 5.6 | 2.2 |
| 34 | 6.3 | 2.8 | 5.1 | 1.5 |
| 35 | 6.1 | 2.6 | 5.6 | 1.4 |
| 36 | 7.7 | 3.0 | 6.1 | 2.3 |
| 37 | 6.3 | 3.4 | 5.6 | 2.4 |
| 38 | 6.4 | 3.1 | 5.5 | 1.8 |
| 39 | 6.0 | 3.0 | 4.8 | 1.8 |
| 40 | 6.9 | 3.1 | 5.4 | 2.1 |
| 41 | 6.7 | 3.1 | 5.6 | 2.4 |
| 42 | 6.9 | 3.1 | 5.1 | 2.3 |
| 43 | 5.8 | 2.7 | 5.1 | 1.9 |
| 44 | 6.8 | 3.2 | 5.9 | 2.3 |
| 45 | 6.7 | 3.3 | 5.7 | 2.5 |
| 46 | 6.7 | 3.0 | 5.2 | 2.3 |
| 47 | 6.3 | 2.5 | 5.0 | 1.9 |
| 48 | 6.5 | 3.0 | 5.2 | 2.0 |
| 49 | 6.2 | 3.4 | 5.4 | 2.3 |
| 50 | 5.9 | 3.0 | 5.1 | 1.8 |

## -Mô tả tập dữ liệu wine\_data:

## +Được lấy từ kanggle.com

## +Bộ dữ liệu gồm 178 bản ghi của 3 loại rượu khác nhau (Rượu vodka, Rượu Whisky, Rượu Cocktail)

## + Bao gồm 13 thuộc tính (nồng độ axit tataric, Ph, Clo, Alc,…) .

## 2.2. Áp dụng bài toán học máy(SVM ,KNN) cho bài toán phân loại rượu

Trong phần này, chúng ta sẽ tách 178 dữ liệu dataset ra thành hai phần, gọi là training set và test set. Thuật toán sẽ dựa vào thông tin ở training set để dự đoán xem mỗi dữ liệu trong test set tương ứng với loại rượu nào. Dữ liệu được chuẩn đoán này sẽ được đối chiếu với loại rượu thật của mỗi dữ liệu trong test set để đánh giá hiệu quả của thuật toán SVM so với KNN.

**2.2.1.Thuật toán KNN trong bài toán phân loại rượu**

1. **Định nghĩa**

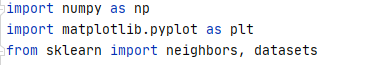
K-nearest neighbor (KNN) là một trong những thuật toán học có giám sát trong Machine Learning. Ý tưởng của KNN là tìm ra output của dữ kiệu dựa trên thông tin của những dữ liệu training gần nó nhất.

1. **Quy trình làm việc của thuật toán KNN**

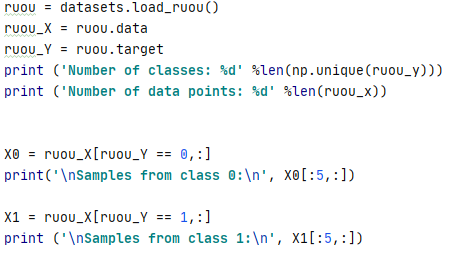
* Bước 1: xác định tham số K= số láng giềng gần nhất.
* Bước 2: tính khoảng cách đối tượng cần phân lớp với tất cả các đối tượng trong training data.
* Bước 3: sắp xếp khoảng cách theo thứ tự tăng dần và xác định K láng giềng gần nhất với đối tượng cần phân lớp
* Bước 4: lấy tất cả các lớp của K láng giềng gần nhất.
* Bước 5: dựa vào phần lớn lớp của K để xác định lớp cho đối tượng cần phân lớp.

1. **Áp dụng vào bài toán phân loại rượu**

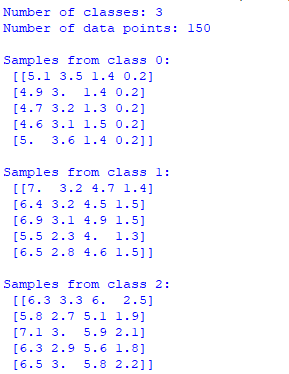
* Bước 1: Khai báo các thư viện cần thiết



* Bước 2: Load dữ liệu và hiện thị vài dữ liệu mẫu

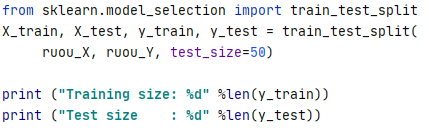


Kết quả nhận được:



* Bước 3: Tách training set và test set:

Giả sử chúng ta muốn dùng 50 điểm dữ liệu cho test set và 100 điểm còn lại cho training set. Thư viện sklearn có một hàm số cho phép chúng ta ngẫu nhiên lựa chọn:



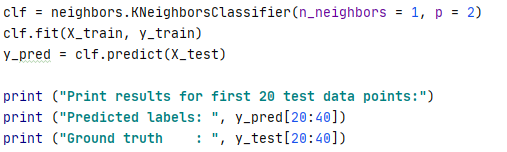
Kết quả nhận được:

Training size: 128

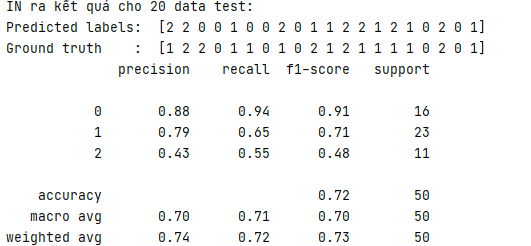
Test size : 50

* Bước 4: KNN dự đoán

Xét trường hớp K=1, tức là với mỗi điểm dữ liệu test ta chỉ xét 1 điểm dữ liệu training gần nhất và lấy nhãn cả điểm đó để dự đoán cho điểm dữ liệu test.



Kết quả nhận được:



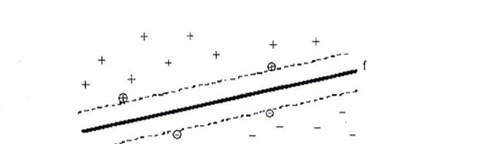
**2.2.2.Thuật toán SVM trong bài toán phân loại rượu.**

1. **Định nghĩa**

+Là phương pháp dựa trên nền tảng của lý thuyết thống kê nên có một nền tảng toán học chặt chẽ để đảm bảo rằng kết quả tìm được là chính xác

Là thuật toán học giám sát (*supervied learning*) được sử dụng cho phân lớp dữ liệu.

+Là 1 phương pháp thử nghiệm, đưa ra 1 trong những phương pháp mạnh và chính xác nhất trong số các thuật toán nổi tiếng về phân lớp dữ liệu



+SVM là một phương pháp có tính tổng quát cao nên có thể được áp dụng cho nhiều loại bài toán nhận dạng và phân loại.

1. **Quy trình làm việc của thuật toán SVM**

+Để phân nhiều lớp thì kỹ thuật SVM nguyên thủy sẽ chia không gian dữ liệu thành 2 phần và quá trình này lặp lại nhiều lần. Khi đó hàm quyết định phân dữ liệu vào lớp thứ i của tập n , 2-Iớp sẽ là:

*fi(x)* = wiix + bi

Những phần tử *x* là support vector sẽ thỏa điều kiện

+1 nếu thuộc lớp i

*fi (x)* =

                                 -1 nếu thuộc phần còn lại

+Như vậy, bài toán phân nhiều lớp sử dụng phương pháp SVM hoàn toàn có thể thực hiện giống như bài toán hai lớp. Bằng cách sử dụng chiến lược *"một- đối- một”*(*one - against - one*).

Giả sử bài toán cần phân loại có k lớp (k > 2), chiến lược *"một-đối-một”*sẽ tiến hành  k(k-l)/2 lần phân lớp nhị phân sử dụng phương pháp SVM. Mỗi lớp sẽ tiến hành phân tách với k-1 lớp còn lại để xác định k-1 hàm phân tách dựa vào bài toán phân hai lớp bằng phương pháp SVM.

**+**Các bước chính của phương pháp SVM

Phương pháp SVM yêu cầu dữ liệu được diễn tả nhớ các vector của các số thực. Như vậy nếu đầu vào chưa phải là số thì ta cần phải tìm cách chuyển chúng về dạng số của SVM

Tiền xử lý dữ liệu: Thực hiện biến đổi dữ liệu phù hợp cho quá trình tính toán, tránh các số quá lớn mô tả các thuộc tính. Thương nên co giãn (*scaling*) dữ liệu để chuyển về đoạn [-1, 1] hoặc [0, 1].

Chọn hàm hạt nhân: Lựa chọn hàm hạt nhân phù hợp tương ứng cho từng bài toán cụ thể để đạt được độ chính xác cao trong quá trình phân lớp.

Thực hiện việc kiểm tra chéo để xác định các tham số cho ứng dụng. Điều này cũng quyết định đến tính chính xác của quá trình phân lớp.

Sử dụng các tham số cho việc huấn luyện với tập mẫu. Trong quá trình huấn luyện sẽ sử dụng thuật toán tối ưu hóa khoảng cách giữa các siêu phẳng trong quá trình phân lớp, xác định hàm phân lớp trong không gian đặc trưng nhờ việc ánh xạ dữ liệu vào không gian đặc trưng bằng cách mô tả hạt nhân, giải quyết cho cả hai trường hợp dữ liệu là phân tách và không phân tách tuyến tính trong không gian đặc trưng.

1. **Áp dụng cho bài toán phân loại rượu**

import numpy as np  
import matplotlib.pyplot as plt  
from sklearn import neighbors, datasets  
import pandas as pd  
from phanloairuou import y\_pred  
  
*#Load dữ liệu*wine = datasets.load\_wine()  
wine\_X = wine.data  
wine\_y = wine.target  
print(wine\_X.shape)  
print (**'Number of classes: %d'** %len(np.unique(wine\_y)))  
print (**'Number of data points: %d'** %len(wine\_y))  
*#whisky*X0 = wine\_X[wine\_y == 0,:]  
print (**'**\n**Samples from class 0:**\n**'**, X0[:5,:])  
  
*#ruouvodka*X1 = wine\_X[wine\_y == 1,:]  
print (**'**\n**Samples from class 1:**\n**'**, X1[:5,:])  
  
*#ruoucocktail*X2 = wine\_X[wine\_y == 2,:]  
print (**'**\n**Samples from class 2:**\n**'**, X2[:5,:])  
  
from sklearn.model\_selection import train\_test\_split  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(  
 wine\_X, wine\_y, test\_size=50)  
  
print(**"Training size: %d"** %len(y\_train))  
print (**"Test size : %d"** %len(y\_test))  
  
  
*#kỹ thuật học máy SVM*from sklearn.svm import SVC  
model2 = SVC().fit(X\_train,y\_train)  
y\_pred2 = model2.predict(X\_test)  
  
def accuracy\_score(y\_test, y\_pred2):  
 pass  
  
*#print("Accuracy Score 2:", accuracy\_score(y\_test, y\_pred2), "\n")  
#print( (y\_test,y\_pred2))*print( **"IN ra kết quả cho 20 data test:"**)  
print (**"Predicted labels: "**, y\_pred[20:40])  
print (**"Ground truth : "**, y\_test[20:40])  
  
 *#Đánh giá mô hình học máy*from sklearn.metrics import confusion\_matrix, classification\_report  
print(classification\_report(y\_test,y\_pred))  
  
print (**'confusion\_matrix ='**)  
print(confusion\_matrix(y\_test,y\_pred))

## 🡪Kết quả nhận được:

## 

## Kết quả cho thấy từ 2 thuật toán SVM và KNN thì KNN cho độ dự đoán chính xác hơn (0.72 so với 0.4).

## Ngoài ra để cải thiện độ dự đoán chính xác cho thuật toán SVM ta có thể sử dụng kỹ thuật SMOTE để cân bằng dữ liệu sau đó mới huấn luyện bằng SVM rồi chạy code .

## 2.3. Kết luận

## Những kết quả đạt được:

- Sự hiểu biết về thuật toán KNN,SVM cơ bản tương đối tốt.

-Thuật toán SVM:

* Tiết kiệm bộ nhớ: Do chỉ có một tập hợp con của các điểm được sử dụng trong quá trình huấn luyện và ra quyết định thực tế cho các điểm dữ liệu mới nên chỉ có những điểm cần thiết mới được lưu trữ trong bộ nhớ khi ra quyết định.
* Tính linh hoạt - phân lớp thường là phi tuyến tính. Khả năng áp dụng Kernel mới cho phép linh động giữa các phương pháp tuyến tính và phi tuyến tính từ đó khiến cho hiệu suất phân loại lớn hơn.

-Thuật toán KNN:

* Là thuật toán phân lớp đơn giản dễ dàng sử dụng
* Độ phức tạp tính toán nhỏ.
* Xử lý tốt với tập dữ liệu nhiễu

🡪 Từ những gì đã làm được, từ đó hiểu biết thêm về AI, ứng dụng của của ML vào đời sống công nghệ hiện đại.

* **Những hạn chế:**

**-Thuật toán KNN:**

* Thuật toán KNN phụ thuộc nhiều vào hệ số K.
* Kết quả đưa ra có sự thay đổi (vì các điểm xét lấy ngẫu nhiên).
* Chưa thực sự hiểu hết về bài toán.

**-Thuật toán SVM:**

* Chưa thể hiện rõ tính xác suất: Việc phân lớp của SVM chỉ là việc cố gắng tách các đối tượng vào hai lớp được phân tách bởi siêu phẳng SVM. Điều này chưa giải thích được xác suất xuất hiện của một thành viên trong một nhóm là như thế nào.

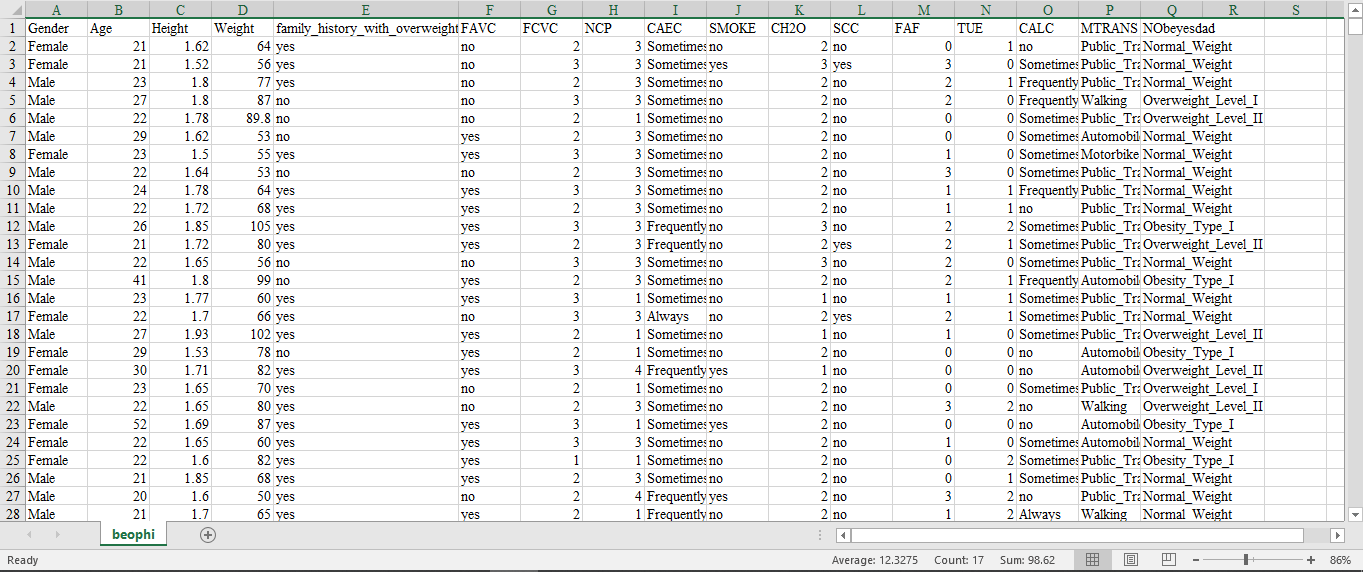
# CHƯƠNG 3:THUẬT TOÁN DESION TREE TRONG BÀI TOÁN ƯỚC TÍNH ĐỘ BÉO PHÌ

## *3.1. Mô tả chi tiết dữ liệu*

Bộ dữ liệu được lấy tại UCI – Machine Learning Repository, bộ dữ liệu muốn liệt kê các thông số dẫn đến các mức độ béo phì mà 3 quốc gia Mexico, Peru và Colombia, dự trên thói quen ăn uống và tình trạng thể chất của họ.

Nguồn lấy dữ liệu:

<https://archive.ics.uci.edu/ml/datasets/Estimation+of+obesity+levels+based+on+eating+habits+and+physical+condition>



Bộ dataset này gồm 17 thông số được đề ra gồm có:

- Bảng mô tả chi tiết các thuộc tính của bộ dữ liệu:

|  |  |
| --- | --- |
| Gender | giới tính  + Female: giới tính nữ  + Male: giới tính nam |
| Age | độ tuổi |
| Height | chiều cao |
| Weight | chiều cao |
| Family\_history\_with\_overweight | tiền sử gia đình bị thừa cân |
| FAVC | Thường xuyên tiêu thụ thực phẩm nhiều calo |
| FCVC | Tần suất tiêu thụ dau |
| NCP | Số bữa ăn chính |
| CAEC | Mức tiêu thụ thực phẩm giữa các bữa ăn |
| SMOKE | tiền sử hút thuốc lá |
| CH2O | tiêu thụ nước hằng ngày |
| SCC | Theo dõi lượng calo tiêu thụ |
| FAF | Tần suất hoạt động thể chất |
| TUE | Thời gian sử dụng thiết bị công nghệ |
| CALC | Mức độ tiêu thụ rượu |
| MTRANS | Phương tiện di chuyển được sử dụng |
| Nobeyesdad | biến lớp  + Normal\_Weight: Cân nặng bình thường  + Overweight\_Level\_I: Thừ cân loại 1  + Overweight\_Level\_II: Thừa cân loại 2  + Overweight\_Level\_III: Thừa cân loại 3  + Obesity\_Type\_I: Béo phì loại 1  + Insufficient\_Weight: Không đủ trọng lượng |

**3.2. Mô tả bài toán học máy**

**3.2.1. Mô tả**

Mục tiêu của bài toán phân lớp được sử dụng trong bộ dataset này là tìm ra các đối tượng năm trong lớp béo phì , thừa cân ở 2111 đối tượng được liệt kê trong bộ dataset.

**3.2.2. Bài toán phân lớp (classification)**

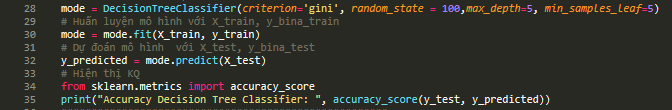
Với bộ dữ liệu này ta xử dụng bài toán phân lớp nhiều lớp (17 lớp) tượng đương với 17 thuộc tính.

Đầu vào X gồm 16 thuộc tính (16 cột ): Gender, Age, Height, Weight, FAVC, FCVC, NCP, Family\_history\_with\_overweight, CAEC, SMOKE, CH2O, SCC, FAF, TUE, CALC, MTRANS.

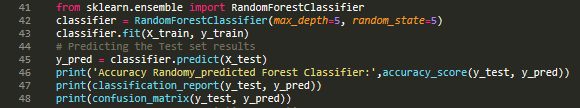
Đầu ra y gồm 1 thuộc tính (cột) cuối cùng: Nobeyesdad, y gồm có 6 lớp: Normal\_Weight, Overweight\_Level\_I, Overweight\_Level\_II, Overweight\_Level\_III, Obesity\_Type\_I, Insufficient\_Weight.

**3.2.3. Các kỹ thuật học máy**

- Kỹ thuật Decision Tree Classifier

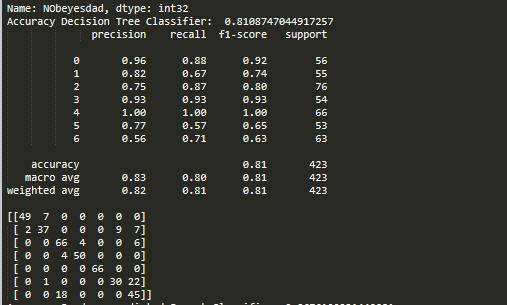


- Kỹ thuật Random Forest Classifier

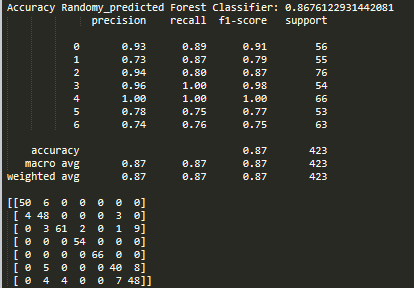


**3.2.4. Phương pháp đánh giá của các kỹ thuật**

**-** Độ chính xác của kỹ thuật Decision Tree Classifier sử dụng phương pháp đánh giá accuracy



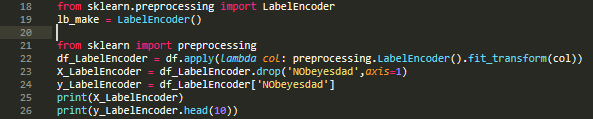
**-** Độ chính xác của kỹ thuật Random Forest Classifier sử dụng phương pháp đánh giá accuracy



**3.3. Mô tả chi tiết cách xử lý dữ liệu**

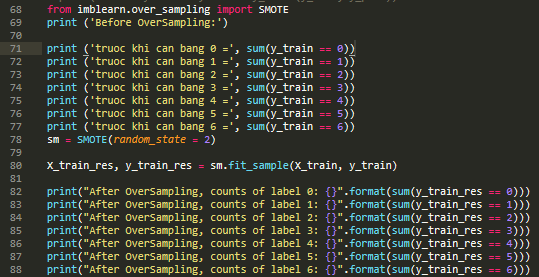
**3.3.1. Xử lý dữ liệu**

Do dữ liệu gồm có 2 dạng text,bool, number, ta sử dụng kỹ thuật LableEncoder để mã hóa về số để tính toán:

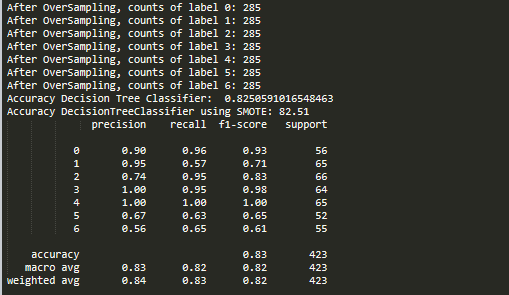


**3.3.2. Cân bằng dữ liệu**

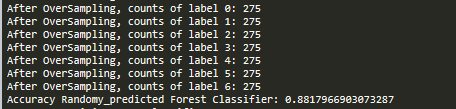
- Ta dùng SMOTE để cân bằng dữ liệu:



* Kết quả của kỹ thuật Decision Tree Classifier sau khi cân bằng dữ liệu:



* Kết quả của kỹ thuật Random Forest Classifier sau khi cân bằng dữ liệu:



**3.4. Kết luận**

**-** Kỹ thuật Decision Tree Classifier

Với kỹ thuật Decision Tree là một mỹ thuật rất mạnh mẽ nhưng khi áp dụng vào dataset này thì chưa đem lại độ chính sác cao lắm.

- Kỹ thuật Random Forest Classifier

Với kỹ thuật Random Forest nó rất nhạy cảm với dữ liệu không có sự cân bằng

- So sánh 2 kỹ thuật

Với việc được dùng trong bài ta thấy Randorm Forest vẫn cao hơn Decision Tree, nhưng 2 giá trị chỉ ở khoảng (0.8 ~ 0.882) chứng tỏ 2 kỹ thuật này chưa đem lại hiệu quả cao khi áp dụng vào việc lọc ra các đối tượng thừa cân và béo phì trong dataset này.

- Phương án giải quyết:

Cần áp dụng các thuật toán cũng như kỹ thuật học máy khác để xử lý bộ dữ liệu này làm sao tìm ra các thuật toán và kỹ thuật máy học đem lại độ chính sác cao nhất, tối ưu nhất.