**MỤC LỤC**

[**CHƯƠNG 1. GIỚI THIỆU KỸ THUẬT PHÂN LỚP BẰNG CÂY QUYẾT ĐỊNH 3**](#_heading=h.30j0zll)

[**1.1 Cây quyết định 3**](#_heading=h.1fob9te)

[**1.1.1 Khái niệm 3**](#_heading=h.3znysh7)

[**1.1.2 Cảm ứng của cây quyết định 3**](#_heading=h.2et92p0)

[**1.1.3 Phân loại 4**](#_heading=h.tyjcwt)

[**1.1.4 Ưu, nhược điểm của thuật toán cây quyết định 11**](#_heading=h.3dy6vkm)

[**1.2 Hướng dẫn cài đặt môi trường cho python 13**](#_heading=h.1t3h5sf)

[**1.2.1 Giới thiệu về Python 13**](#_heading=h.4d34og8)

[**1.2.2 Các bước cài đặt 13**](#_heading=h.2s8eyo1)

[**CHƯƠNG 2. KỸ THUẬT PHÂN LỚP BẰNG CÂY QUYẾT ĐỊNH VÀ ỨNG DỤNG 16**](#_heading=h.17dp8vu)

[**1.1 Mô tả bộ dữ liệu 16**](#_heading=h.3rdcrjn)

[**1.2  Xác định bài toán học máy 17**](#_heading=h.26in1rg)

[**1.3 Phương pháp đánh giấ mô hình học máy dựa trên kết quả dự đoán 18**](#_heading=h.lnxbz9)

**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.

Decision trees là một trong những phương pháp Data mining, cụ thể Classìfication được sử dụng nhiều nhất trong các dự án nghiên cứu dữ liệu, là phương pháp Supervised learning – học có giám sát hiệu quả nhất vì các đặc điểm của nó và được ứng dụng trong mọi khía cạnh của các lĩnh vực khác nhau từ kinh tế đến xã hội, không chỉ riêng khoa học dữ liệu, và là mảng kiến thức quan trọng mà bất kỳ chuyên gia phân tích nào phải có.

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 phân lớp bằng cây quyết định và ứng dụng”.**

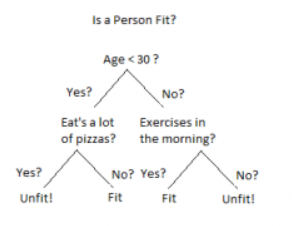
# CHƯƠNG 1. GIỚI THIỆU KỸ THUẬT PHÂN LỚP BẰNG CÂY QUYẾT ĐỊNH

Chương này tóm tắt cơ bàn về khái niệm, phân loại công dụng của cây quyết định và các vấn đề liên quan.

## 1.1 Cây quyết định

### 1.1.1 Khái niệm

Cây quyết định là một loại Học máy được giám sát (nghĩa là bạn giải thích đầu vào là gì và đầu ra tương ứng là gì trong dữ liệu đào tạo) trong đó dữ liệu liên tục được phân chia theo một tham số nhất định.



Ví dụ về cây quyết định

### 1.1.2 Cảm ứng của cây quyết định

Quy nạp cây quyết định là phương pháp học các cây quyết định từ tập huấn luyện. Tập huấn luyện bao gồm các thuộc tính và nhãn lớp. Các ứng dụng của quy nạp cây quyết định bao gồm thiên văn học, phân tích tài chính, chẩn đoán y tế, chế tạo và sản xuất.

Cây quyết định là một cấu trúc giống như cây lưu đồ được tạo từ các bộ giá trị tập huấn luyện. Tập dữ liệu được chia thành các tập con nhỏ hơn và hiện diện dưới dạng các nút của cây. Cấu trúc cây có một nút gốc, các nút bên trong hoặc các nút quyết định, nút lá và các nhánh.

Nút gốc là nút trên cùng. Nó đại diện cho thuộc tính tốt nhất được chọn để phân loại. Các nút bên trong của các nút quyết định đại diện cho việc kiểm tra một thuộc tính của nút lá bộ dữ liệu hoặc nút đầu cuối đại diện cho nhãn phân loại hoặc quyết định. Các nhánh hiển thị kết quả của thử nghiệm được thực hiện.

Một số cây quyết định chỉ có *các nút nhị phân* , có nghĩa là chính xác hai nhánh của một nút, trong khi một số cây quyết định là không nhị phân.

### **1.1.3 Phân loại**

Cây quyết định được chia thành 2 loại: cây phân loại và cây hồi quy

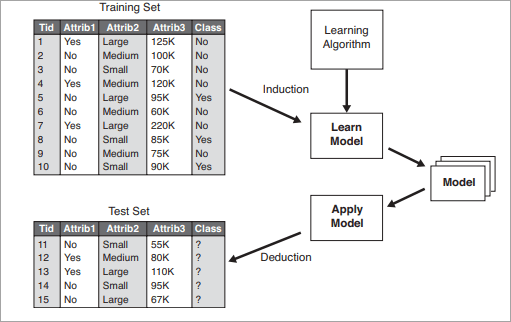
#### 1.1.3.1 phân tích phân loại phân loại

Phân loại dữ liệu là một dạng phân tích xây dựng mô hình mô tả các biến lớp quan trọng. Ví dụ , một mô hình được xây dựng để phân loại các đơn xin vay ngân hàng là an toàn hay rủi ro. Các phương pháp phân loại được sử dụng trong học máy và nhận dạng mẫu.

Ứng dụng của phân loại bao gồm phát hiện gian lận, chẩn đoán y tế, tiếp thị mục tiêu, v.v. Đầu ra của bài toán phân loại được coi là "Chế độ" của tất cả các giá trị quan sát của nút đầu cuối.

**Một quy trình hai bước được tuân theo, để xây dựng một mô hình phân loại.**

* Trong bước đầu tiên tức là học tập: Một mô hình phân loại dựa trên dữ liệu đào tạo được xây dựng.
* Trong bước đầu tiên tức là học tập: Một mô hình phân loại dựa trên dữ liệu đào tạo được xây dựng.

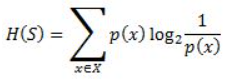


Cách tiếp cận chung để xây dựng mô hình phân loại

#### **1.1.3.2 Phân tích hồi quy**

Phân tích hồi quy được sử dụng để dự đoán các thuộc tính số.

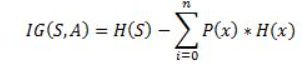
Có rất nhiều thuật toán xây dựng Cây quyết định, nhưng một trong những **thuật toán** tốt nhất được gọi là **Thuật toán ID3** . ID3 là viết tắt của **Lặp đi lặp lại Dichotomiser 3** . Trước khi thảo luận về thuật toán ID3, chúng ta sẽ đi qua một số định nghĩa. **Entropy** Entropy, còn được gọi là Shannon Entropy được ký hiệu là H (S) cho một tập hữu hạn S, là thước đo lượng không chắc chắn hoặc ngẫu nhiên trong dữ liệu.



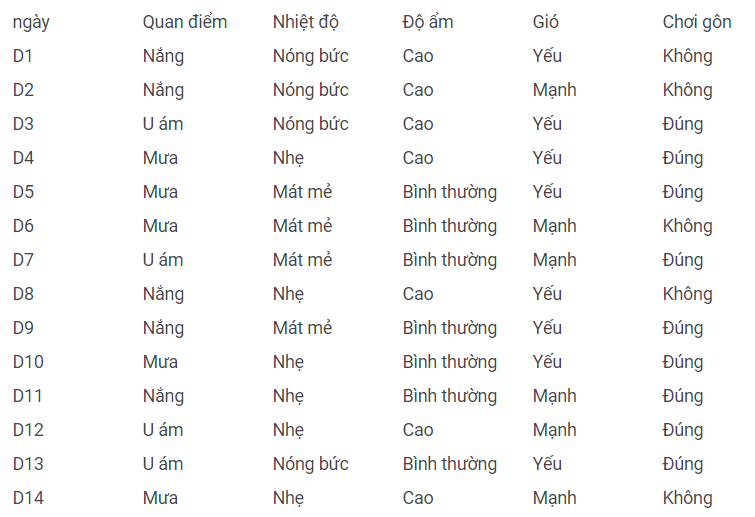
Trực quan, nó cho chúng ta biết về khả năng dự đoán của một sự kiện nhất định. Ví dụ, xem xét một lần tung đồng xu có xác suất đầu là 0,5 và xác suất sấp là 0,5. Ở đây entropy là cao nhất có thể, vì không có cách nào để xác định kết quả có thể là gì. Ngoài ra, hãy xem xét một đồng xu có đầu ở cả hai mặt, entropy của một sự kiện như vậy có thể được dự đoán một cách hoàn hảo vì chúng ta biết trước rằng nó sẽ luôn có đầu. Nói cách khác, sự kiện này **không có tính ngẫu nhiên** do đó entropy của nó bằng không. Đặc biệt, giá trị thấp hơn ngụ ý ít độ không đảm bảo hơn trong khi giá trị cao hơn ngụ ý độ không đảm bảo cao. **Thông tin thu được**Độ lợi thông tin còn được gọi là phân kỳ Kullback-Leibler được ký hiệu là IG (S, A) đối với một tập S là sự thay đổi hiệu quả của entropy sau khi quyết định một thuộc tính cụ thể A. Nó đo lường sự thay đổi tương đối trong entropy đối với các biến độc lập.



Ngoài ra,



trong đó IG (S, A) là thông tin thu được bằng cách áp dụng tính năng A. H (S) là Entropy của toàn bộ tập hợp, trong khi số hạng thứ hai tính Entropy sau khi áp dụng tính năng A, trong đó P (x) là xác suất của sự kiện x. Hãy hiểu điều này với sự trợ giúp của một ví dụ Hãy xem xét một phần dữ liệu được thu thập trong suốt 14 ngày, trong đó các tính năng là Outlook, Nhiệt độ, Độ ẩm, Gió và biến kết quả là liệu Golf có được chơi trong ngày hay không. Bây giờ, công việc của chúng ta là xây dựng một mô hình dự đoán trong đó có 4 thông số trên và dự đoán liệu Golf có được chơi trong ngày hay không. Chúng tôi sẽ xây dựng một cây quyết định để thực trong đó IG (S, A) là thông tin thu được bằng cách áp dụng tính năng A. H (S) là Entropy của toàn bộ tập hợp, trong khi số hạng thứ hai tính Entropy sau khi áp dụng tính năng A, trong đó P (x) là xác suất của sự kiện x. Hãy hiểu điều này với sự trợ giúp của một ví dụ Hãy xem xét một phần dữ liệu được thu thập trong suốt 14 ngày, trong đó các tính năng là Outlook, Nhiệt độ, Độ ẩm, Gió và biến kết quả là liệu Golf có được chơi trong ngày hay không. Bây giờ, công việc của chúng ta là xây dựng một mô hình dự đoán trong đó có 4 thông số trên và dự đoán liệu Golf có được chơi trong ngày hay không. Chúng tôi sẽ xây dựng một cây quyết định để thực hiện điều đó bằng **thuật toán ID3.**  

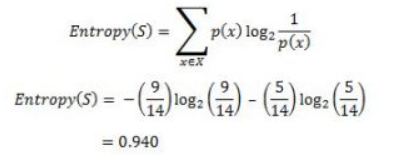


**Thuật toán ID3 sẽ thực hiện các nhiệm vụ sau một cách đệ quy**

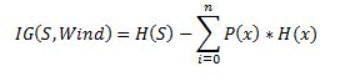
1. Tạo nút gốc cho cây
2. Nếu tất cả các ví dụ đều tích cực, hãy trả về nút lá 'dương'
3. Nếu tất cả các ví dụ đều là tiêu cực, hãy trả về nút lá 'âm'
4. Tính entropi của trạng thái hiện tại H (S)
5. Đối với mỗi thuộc tính, tính toán entropy đối với thuộc tính 'x' được ký hiệu là H (S, x)
6. Chọn thuộc tính có giá trị lớn nhất là IG (S, x)
7. Xóa thuộc tính cung cấp IG cao nhất khỏi tập hợp các thuộc tính
8. Lặp lại cho đến khi chúng tôi sử dụng hết tất cả các thuộc tính hoặc cây quyết định có tất cả các nút lá.

Bây giờ chúng ta sẽ t iếp tục và phát triển cây quyết định. Bước đầu tiên là tính H (S), Entropy của trạng thái hiện tại. Trong ví dụ trên, chúng ta có thể thấy tổng cộng có 5 Không và 9 Có.

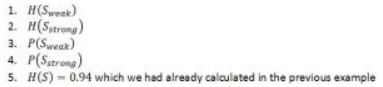
|  |  |  |
| --- | --- | --- |
| Đúng | Không | Toàn bộ |
| 9 | 5 | 14 |



Hãy nhớ rằng Entropy bằng 0 nếu tất cả các thành viên thuộc cùng một lớp, và 1 khi một nửa trong số họ thuộc một lớp và nửa còn lại thuộc lớp khác là ngẫu nhiên hoàn hảo. Đây là 0,94, có nghĩa là phân phối khá ngẫu nhiên. **Bây giờ bước tiếp theo là chọn thuộc tính mang lại cho chúng ta Mức tăng thông tin cao nhất có thể** mà chúng ta sẽ chọn làm nút gốc. Hãy bắt đầu với 'Wind'

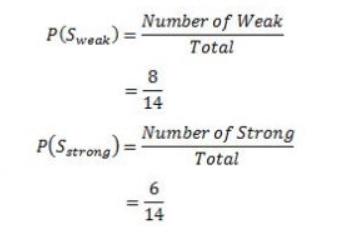


trong đó 'x' là các giá trị có thể có cho một thuộc tính. Ở đây, thuộc tính 'Gió' nhận hai giá trị có thể có trong dữ liệu mẫu, do đó x = {Yếu, Mạnh} Chúng ta sẽ phải tính:

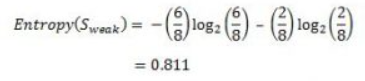


Trong số tất cả 14 ví dụ, chúng ta có **8 nơi gió yếu và 6 nơi gió lớn Mạnh mẽ** .

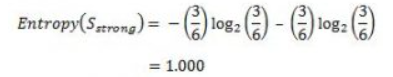
|  |  |  |
| --- | --- | --- |
| Gió = Yếu | Gió = Mạnh | Toàn bộ |
| số 8 | 6 | 14 |



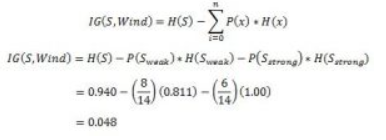
Bây giờ trong số 8 ví dụ Yếu, 6 trong số đó là "Có" cho Chơi Golf và 2 trong số đó là "Không" cho "Chơi Golf". Vì vậy, chúng tôi,



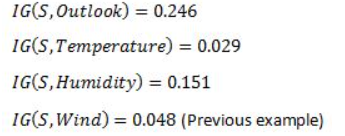
Tương tự, trong số 6 ví dụ Mạnh, chúng tôi có **3 ví dụ trong đó kết quả là "Có" cho Chơi Golf và 3 trong đó chúng tôi có "Không" cho Chơi Golf** .



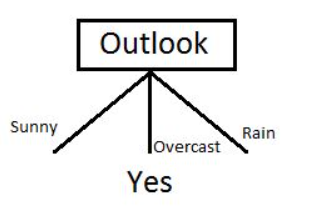
Tương tự, trong số 6 ví dụ Mạnh, chúng tôi có **3 ví dụ trong đó kết quả là "Có" cho Chơi Golf và 3 trong đó chúng tôi có "Không" cho Chơi Golf** .



Điều này cho chúng tôi biết Mức tăng thông tin bằng cách coi 'Gió' là tính năng và cung cấp cho chúng tôi mức tăng thông tin là **0,048** . Bây giờ chúng ta phải tính toán tương tự Mức tăng thông tin cho tất cả các tính năng.



Chúng ta có thể thấy rõ rằng IG (S, Outlook) có mức tăng thông tin cao nhất là 0,246, **do đóchúng tôi đã chọn thuộc tính Outlook làm nút gốc** . Tại thời điểm này, cây quyết định trông như thế nào.



Ở đây chúng tôi quan sát thấy rằng bất cứ khi nào triển vọng là U ám, Chơi gôn luôn là "Có", không phải ngẫu nhiên mà có, cây đơn giản cho ra kết quả **là do thuộc tính Outlook đưa ra thông tin cao nhất** . Bây giờ chúng ta tiến hành như thế nào từ thời điểm này? Chúng ta có thể đơn giản áp dụng **đệ quy** , bạn có thể muốn xem các bước thuật toán được mô tả trước đó. Bây giờ chúng tôi đã sử dụng Outlook, chúng tôi còn lại ba trong số đó là Độ ẩm, Nhiệt độ và Gió. Và, chúng tôi có ba giá trị có thể có của Outlook: Nắng, U ám, Mưa. Trong đó nút Overcast đã kết thúc với nút lá 'Có', vì vậy chúng ta còn lại hai cây con để tính toán: Nắng và Mưa.

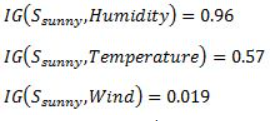


 Bảng trong đó giá trị của Outlook là Sunny trông như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Nhiệt độ | Độ ẩm | Gió | Chơi gôn |
| Nóng bức | Cao | Yếu | Không |
| Nóng bức | Cao | Mạnh | Không |
| Nhẹ | Cao | Yếu | Không |
| Mát mẻ | Bình thường | Yếu | Đúng |
| Nhẹ | Bình thường | Mạnh | Đúng |



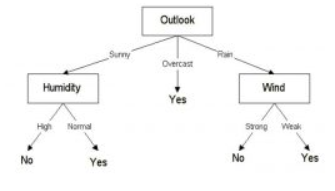
Theo cách tương tự, chúng tôi tính toán các giá trị sau:



Như chúng ta có thể thấy **Mức tăng thông tin cao nhất được đưa ra bởi Độ ẩm** . Tiến hành theo cách tương tự



sẽ cho chúng ta Wind là người có thu thập thông tin cao nhất. Cây Quyết định cuối cùng trông giống như thế này. Cây Quyết định cuối cùng trông giống như thế này.



### 1.1.4 Ưu, nhược điểm của thuật toán cây quyết định

**1.1.4.1 Ưu điểm**

* Thuật toán Decision trees đơn giản, trực quan, không quá phức tạp để hiểu ngay lần đầu tiên, khác với các thuật toán ví dụ như Artificial Neural network không thể hiện rõ quy luật phân loại. Đồng thời bộ dữ liệu training không nhất thiết phải quá lớn để xây dựng mô hình phân tích.
* Một số thuật toán cây quyết định có khả năng xử lý dữ liệu bị missing và dữ liệu bị lỗi mà không cần áp dụng phương pháp như “input missing values” hay loại bỏ. Bên cạnh đó Decision trees ít bị ảnh hưởng bởi các dữ liệu ngoại lệ (outliers).
* Thuật toán cây quyết định là phương pháp không sử dụng tham số, “nonparametric”, nên không cần phải có các giả định ban đầu về các quy luật phân phối như trong thống kê, và nhờ đó kết quả phân tích có được là khách quan , “tự nhiên” nhất.
* Thuật toán cây quyết định có thể giúp chúng ta phân loại đối tượng dữ liệu theo biến mục tiêu có nhiều lớp, nhiều nhóm khác nhau (multi-class classification) đặc biệt nếu biến mục tiêu là dạng biến định lượng phức tạp.
* Thuật toán cây quyết định có thể áp dụng linh hoạt cho các biến target, biến mục tiêu là biến định tính (classification task)
* Thuật toán cây quyết định mang lại kết quả dự báo có độ chính xác cao, dễ dàng thực hiện, nhanh chóng trong việc huấn luyện, không cần phải chuyển đổi các biến vì kết quả sẽ như nhau với bất kể loại biến dữ liệu biến đổi ra sao.
* Dựa trên quy luật ra quyết định (Decision rule) để xây dựng nên thuật toán cây quyết định rất dễ diễn giải hay giải thích đến người xem, người nghe.
* Thuật toán cây quyết định vẫn nói lên được mối quan hệ giữa các biến, các thuộc tính dữ liệu một cách trực quan nhất mặc dù không thể hiện được rõ mối quan hệ tuyến tính, hay mức độ liên hệ giữa chúng như phương pháp phân tích hồi quy (regression analysis) có được.
* Ngoài kinh tế, tài chính, thuật toán cây quyết định có thể được ứng dụng trong lĩnh vực y tế, nông nghiệp, sinh học.

**1.1.4.2 Khuyết điểm**

* Thuật toán cây quyết định khi được áp dụng với bộ dữ liệu phức tạp, nhiều biến và thuộc tính khác nhau có thể dẫn đến mô hình bị overfitting, quá khớp với dữ liệu training dẫn đến vấn đề không đưa ra kết quả phân loại chính xác khi áp dụng cho dữ liệu test, và dữ liệu mới.
* Đối với thuật toán cây quyết định khi có sự thay đổi nhỏ trong bộ dữ liệu có thể gây ảnh hưởng đến cấu trúc của mô hình.
* Thuật toán cây quyết định chỉ áp dụng cho biếnt định tính nếu phân loại sai có thể dẫn đến sai lầm nghiêm trọng. Còn đối với thuật toán cây quyết định áp dụng cho biến định lượng, thì chỉ phân loại đối tượng, hay dự báo theo phạm vi giá trị (range) được tạo ra trước đó, vì vậy đây cũng là hạn chế khi khả năng có nhiều phạm vi giá trị khác mà thuật toán chưa xét đến.
* Thuật toán cây quyết định có khả năng “bias” hay thiên vị nếu bộ dữ liệ không được cân bằng. Tuy nhiên khi mô hình quá phức tạp, mọi biến dữ liệu đều có khả năng phân nhánh và làm cơ sở phân loại các đối tượng dữ liệu, thì lúc này “bias” ở mức thấp nhưng nguy cơ không thể áp dụng được dữ liệu mới.
* Thuật toán cây quyết định yêu cầu bộ dữ liệu training và test phải được chuẩn bị hoàn hảo, chất lượng tốt phải được cân đối theo các lớp, các nhóm trong biến mục tiêu. Ngoài ra biến mục tiêu phải có các giá trị “rời rạc” dễ nhận biết, không được quá đa dạng, và phải cụ thể để quá trình phân loại diễn ra dễ dàng hơn cho thuật toán.
* Thuật toán cây quyết định được hình thành trên các phân nhánh tại mỗi thời điểm bất kỳ, ở một node hay biến dữ liệu bất kỳ và chỉ quan tâm duy nhất vào việc phân nhanh sao cho tối ưu tại thời điểm ấy, chứ không xét đến toàn bộ mô hình phải được thiết lập hiệu quả ra sao.
* Thuật toán cây quyết định không “hỗ trợ “ kỹ thuật hay khả năng “truy vấn ngược” mà chỉ phân nhánh liên tục dựa trên các công thức phân nhánh cho đến khi thấy được kết quả sau cùng nên chúng ta khó phát hiện được các lỗi ở đâu nếu có sai sót.

## 1.2 Hướng dẫn cài đặt môi trường cho python

### 1.2.1 Giới thiệu về Python

**Python** là ngôn ngữ lập trình hướng đối tượng được tạo bởi Guido Rossum vào năm 1989.

**Python** được sử dụng rộng rãi trong các lĩnh vực:

* Trí tuệ nhân tạo (Artificial Intelligence)
* Tạo ngôn ngữ tự nhiên (Natural Language Generation)
* Mạng thần kinh (Neural Networks)
* và các lĩnh vực Khoa học máy tính tiên tiến khác.

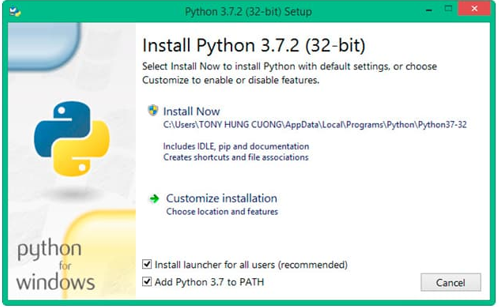
### 1.2.2 Các bước cài đặt

Bước 1: Download và cài đặt Python

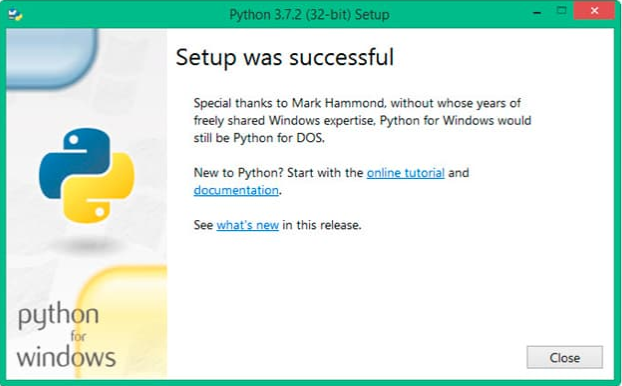
* Vào website [www.python.org](http://www.python.org/)
* Chọn phiên bản Python phù hợp với hệ điều hành (OS) của máy tính mình.
* Download Python về máy tính của mình.



Sau khi download về máy cá nhân, ta cần cài Python lên máy. Ta tích chọn các tùy chọn trên màn hình cài đặt, rồi chọn **Install Now** để tiến hành việc cài đặt. Sau đó ta chờ một lúc để quá trình cài đặt hoàn tất.



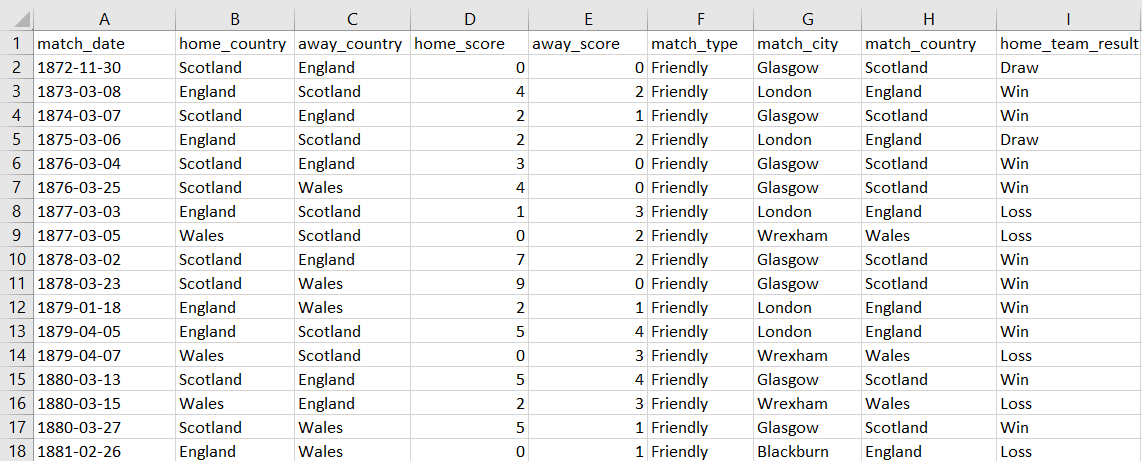
Sau khi cài đặt xong, trình cài đặt sẽ hiển thị màn hình thông báo thành công. Ta nhấn **Close** để kết thúc việc cài đặt Python.



# CHƯƠNG 2. KỸ THUẬT PHÂN LỚP BẰNG CÂY QUYẾT ĐỊNH VÀ ỨNG DỤNG

## 1.1 Mô tả bộ dữ liệu

-        Bộ dữ liệu train.csv gồm 12 thuộc tính (12 cột) và 891 dòng dữ liệu:



-        Link tải: https://www.kaggle.com/hesh97/titanicdataset-traincsv

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Mô tả** |
| 1 | PassengerId | Mã hành khách |
| 2 | Survived | Sốn sót |
| 3 | Pclass | Lớp |
| 4 | Name | Tên hành khách |
| 5 | Sex | Giới tính |
| 6 | Age | Tuổi |
| 7 | SibSp |  |
| 8 | Parch | Tên nước để đối chiếu với kết quả thi đấu |
| 9 | Ticket | Vé |
| 10 | Fare | Gía vé |
| 11 | Cabin | Ca bin |
| 12 | Embarked | Tham gia |

**1.** -        Với bộ dữ liệu này, ta sử dụng bài toán phân lớp nhiều lớp (3 lớp)

* Đầu vào X là 6 thuộc tính (cột) đầu tiên: Survived, Sex, Age, SibSp, Fare, Embarked .
* Đầu ra y là thuộc tính (cột): Survived
* Áp dụng bài toán phân lớp sẽ y chia thành 2 lớp: 0 ,1

- Đọc dữ liệu:

-        Xử lý dữ liệu: Vì dữ liệu chủ yếu là dạng text, ta sử dụng các kỹ thuật tiền xử lý dữ liệu:

* LabelEncoder để mã hóa về số để tính toán

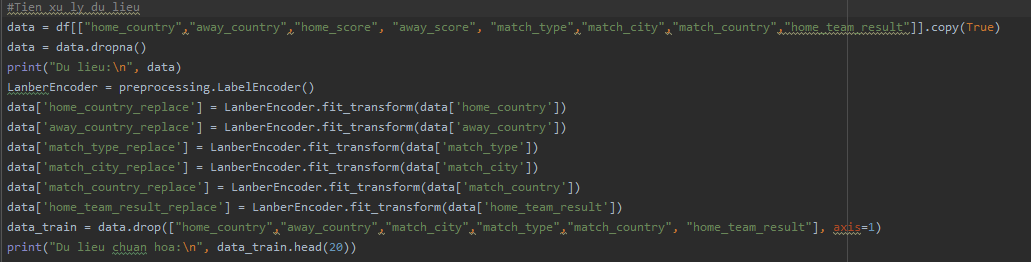
Áp dụng bài toán phân lớp sẽ y chia thành 2 lớp: 0(Sống) 1(Chết)

- Đọc dữ liệu:

df = pd.read\_csv("train.csv")

-        Xử lý dữ liệu: Vì dữ liệu chủ yếu là dạng text, ta sử dụng các kỹ thuật tiền xử lý dữ liệu để chuyển hóa dữ liệu từ dạng chữ về dạng số:

* LabelEncoder để mã hóa về số để tính toán



* Chia tập train và test

from sklearn.model\_selection import train\_test\_split  
X = np.asarray(data\_train[['Sex\_replace', 'Age', 'SibSp', 'Fare', 'Embarked\_replace']])  
Y = np.asarray(data\_train['Survived'])  
X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.3, random\_state=0)  
print('Data:\n', data\_train.shape)

* Dự đoán với tạo dữ liệu train X\_test, Y\_test

# Du doan voi tao du lieu train X\_test , y\_test  
decsionTreeEntropy = DecisionTreeClassifier(criterion="entropy", random\_state=100, max\_depth=5, min\_samples\_leaf=5)  
decsionTreeEntropy.fit(X\_train, Y\_train)  
y\_preEntropy = decsionTreeEntropy.predict(X\_test)  
print("Du doan bang Entrophy", y\_preEntropy)

* Kỹ thuật học máy được sử dụng: Decision Tree Classifier

## 1.3 Phương pháp đánh giấ mô hình học máy dựa trên kết quả dự đoán

**-** Kỹ thuật học máy Decision Tree Classifier sử dụng với độ đo đơn giản Accuracy, Precision, Recall

* Báo cáo phân lớp: classification\_report

#classfier report  
from sklearn.metrics import classification\_report  
print("Bao cao:\n", classification\_report(Y\_test, y\_preEntropy))

- Ma trận hỗn loạn trước khi cân bằng : confusion\_matrix

#Confusion matrix  
from sklearn.metrics import confusion\_matrix  
print('Ma tran hon loan truoc can bang:\n', confusion\_matrix(Y\_test, y\_preEntropy))

-Phương pháp đánh giá: accuracy\_score trước cân bằng

#Bước 3: Đánh giá mô hình học dựa trên kết quả dự đoán (với độ đo đơn giản Accuracy, Precision, Recall)  
print("Accuracy Score truoc can bang:", accuracy\_score(Y\_test, y\_preEntropy)\*100)

* + Kỹ thuật lấy mẫu quá mức

print('Before OverSampling:')  
  
print('No =', sum(Y\_train == 0))  
  
print('Yes =', sum(Y\_train == 1))  
  
sm = SMOTE(random\_state=2)  
  
X\_train\_res, y\_train\_res = sm.fit\_sample(X\_train, Y\_train)  
  
print("After OverSampling, counts of label 0: {}".format(sum(y\_train\_res == 0)))  
  
print("After OverSampling, counts of label 1: {}".format(sum(y\_train\_res == 1)))

* Dự đoán sau khi cân bằng

decsionTreeEntropy = DecisionTreeClassifier(criterion="entropy", random\_state=100, max\_depth=5, min\_samples\_leaf=5)  
decsionTreeEntropy.fit(X\_train\_res, y\_train\_res)  
y\_preEntropHy = decsionTreeEntropy.predict(X\_test)  
print("Du doan bang Entrophy", y\_preEntropHy)

* Báo cáo sau khi cân bằng

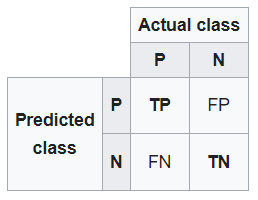
print("Bao cao sau can bang:\n", classification\_report(Y\_test, y\_preEntropHy))

* Ma trận hỗn loạn sau cân bằng
* print('Ma tran hon loan sau can bang:\n', confusion\_matrix(Y\_test, y\_preEntropHy))
* Phương pháp đánh giá: accuracy\_score trước cân bằng

print("Accuracy Score sau cân bằng:", accuracy\_score(Y\_test, y\_preEntropHy)\*100)

-

* Phương pháp đáng giá confusion\_matrix và một số công thức tính:



Trong đó:

P = Positive;

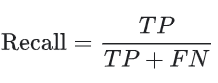
N = Negative;

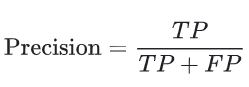
TP = True Positive;

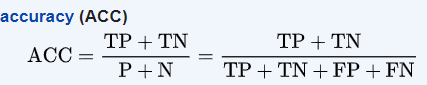
FP = False Positive;

TN = True Negative;

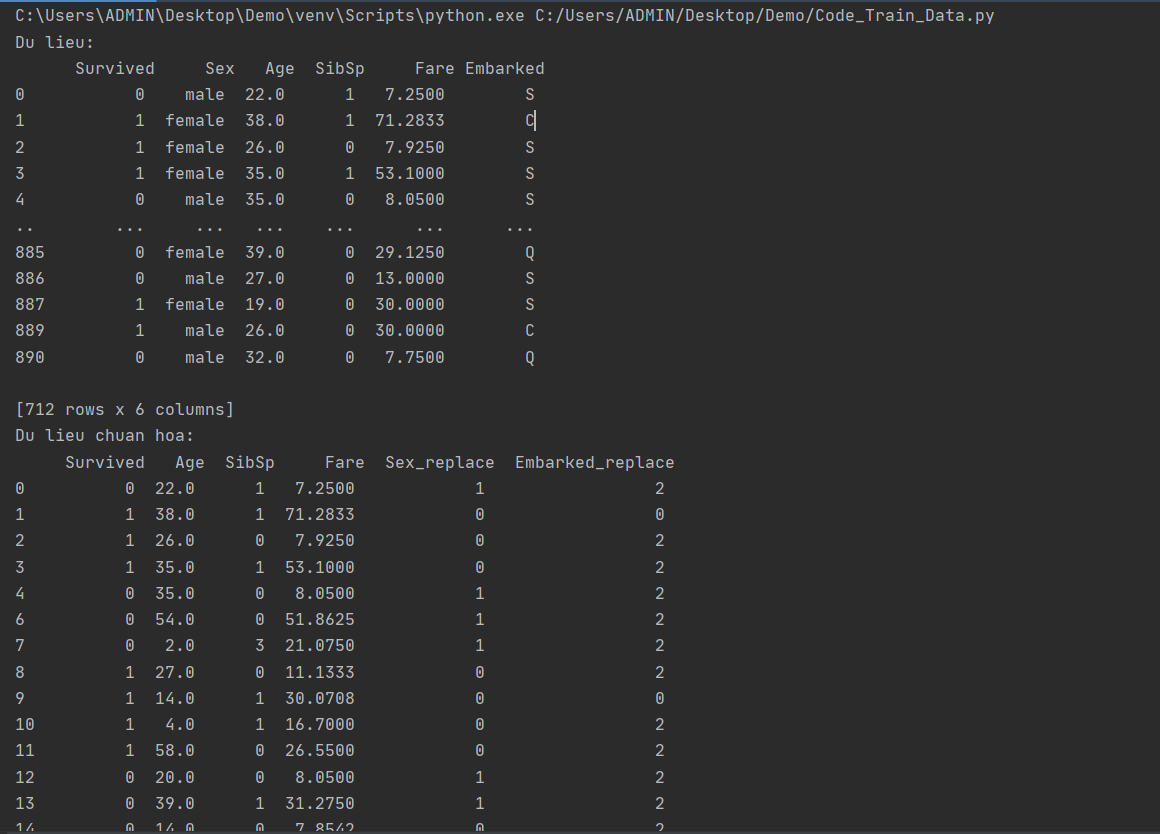
FN = False Negative.

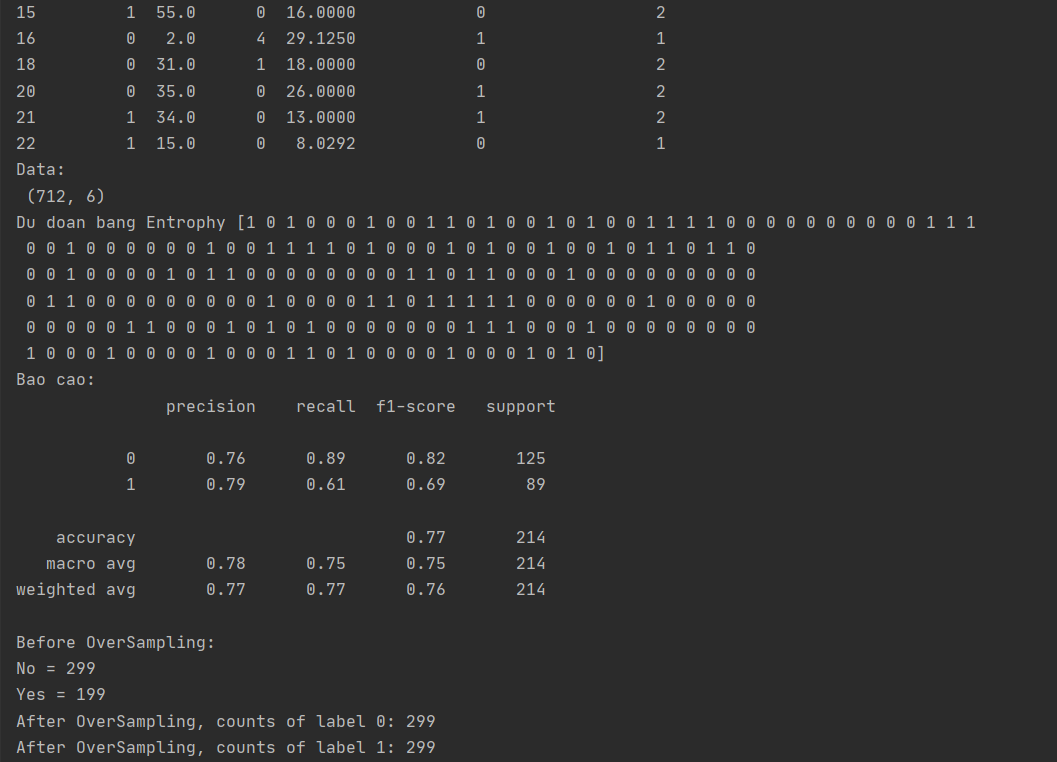






**Kết quả chạy:**

****

****

