TRƯỜNG ĐẠI HỌC THỦY LỢI

**KHOA CÔNG NGHỆ THÔNG TIN**

A blue and white logo

Description automatically generated with low confidence

**BÀI TẬP LỚN**

HỌC PHẦN: HỌC MÁY

**ĐỀ TÀI: Dự đoán chất lượng rượu vang Vinho Verde**

Giáo viên hướng dẫn: Nguyễn Thị Kim Ngân

Nhóm sinh viên thực hiện: Nhóm 6

1. Nguyễn Thị Mây, lớp 62TH1

2. Nguyễn Thị Tươi, lớp 62TH1

3. Nguyễn Hồng Thương, lớp 62TH1

**Hà Nội, ngày 6 tháng 10 năm 2022**

# Phần 1: Lý thuyết

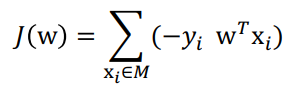
1. **Perceptron**

* Là thuật toán phân lớp đơn giản giúp tìm ra siêu phẳng cho bài toán phân lớp nhị phân (chỉ có 2 lớp dữ liệu).
* Input: Là một tập dữ liệu đã được gán nhãn
* Output: là 1 siêu phẳng
* Cách xây dựng hàm mất mát, và tìm nghiệm bài toán:
  + PT đường boundary:

*f*w(x) = w1.x1 + w2.x2 + … + wd.xd + w0 = 0

*f*w(x) = wT.x = 0

* w = (w0, w1,…, wd) : vector hệ số
* w0 : số hạng tự do
* x = (x0, x1,…, xd): vector đặc trưng
  + Hàm mất mát:



* + - M: số điểm phân lớp lỗi
  + Nghiệm của bài toán w: tìm w sao cho hàm mất mát đạt giá trị nhỏ nhất là 0 (không có điểm dữ liệu nào bị phân lớp sai)
    - Với mỗi điểm phân lớp đúng -> kết thúc thuật toán
    - Với mỗi điểm phân lớp sai nghiệm w được cập nhật lại:



* Phương pháp:

B1: tại thời điểm t=0, chọn ngẫu nhiên 1 vector hệ số w0

B2:

* + tại thời điểm t:
    - nếu không có điểm dữ liệu nào bị phân lớp lỗi thì dừng thuật toán.
    - nếu có điểm bị phân lớp lỗi thì cập nhật lại giá trị của w.

wt+1 = wt + yi.xi

B3: thay đổi t = t+1 quay lại bc 2

1. **ID3**

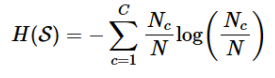
* ID3 là một thuật toán decision tree được áp dụng cho các bài toán classification mà tất cả các thuộc tính đều ở dạng categorical.
* Innput: Là một tập dữ liệu đã được gán nhãn.
* Output: là cây quyết định
* Công việc thực hiện của bài toán:

+ Sử dụng hàm “entropy” làm thước đo trong khi tạo cây quyết định (decision tree) trong thuật toán ID3 (Classification and Regression Tree)

* Các bước thực hiện:

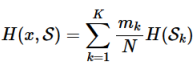
**B1:** tính entropy trên toàn bộ tập S

* + H(S) : entropy trên toàn bộ tập dữ liệu S
  + S: tập dữ liệu huấn luyện(gồm cả x và y)
  + N: tổng số mẫu dữ liệu huấn luyện
  + Nc: số các mẫu dữ liệu trong tập S đc gán nhãn c
  + C: tổng số nhãn khác nhau trong tập S
  + c: chỉ số của nhãn dữ liệu



**B2:** Tính entropy của thuộc tính x trên tập S

* + x: là thuộc tính đang xét
  + K: số giá trị của thuộc tính đó
  + mk: số mẫu có thuộc tính x và có giá trị là k
  + Sk: tập các mẫu sao cho thuộc tính x có giá trị là k
  + H(Sk): entropy trên tập dữ liệu trong S có thuộc tính là x và có giá trị k



**B3:** tính *information gain* trên thuộc tính x



* + Thay vì tìm thuộc tính có information gain cao nhất thì ta tìm thuộc tính có entropy nhỏ nhất



* **Điều kiện dừng:**
  + TH1: nếu tất cả các mẫu trong node con thuộc cùng 1 lớp C (entropy=0) thì node đó đc gán nhãn C
  + TH2: nếu node con là rỗng (không có dữ liệu) thì node đó sẽ đc gán = nhãn phổ biến nhất trong tập S
  + TH3: nếu không còn thuộc tính nào để phân chia (các thuộc tính đều đc xét hết) -> node lá đó sẽ đc gán = nhãn phổ biến nhất

1. **CART**

* Là một kỹ thuật phi tham số hữu ích có thể được sử dụng để giải thích một biến phụ thuộc liên tục hoặc phân loại dưới dạng nhiều biến độc lập. Các biến độc lập có thể liên tục hoặc phân loại. CART sử dụng phương pháp phân vùng thường được gọi là “**chia để trị**”. Thuật toán CART hoạt động để tìm biến độc lập tạo ra nhóm đồng nhất tốt nhất khi tách dữ liệu.
* Innput: Là một tập dữ liệu đã được gán nhãn
* Output: là cây quyết định
* Công việc thực hiện của bài toán:

+ Sử dụng chỉ số “gini index” làm thước đo trong khi tạo cây quyết định (decision tree) trong thuật toán CART (Classification and Regression Tree), chỉ số “gini index” càng cao càng tốt.

* Các bước thực hiện:

**B1:** Tính giá trị Gini

Gini =

* + - C: số lớp cần phân loại (số nhãn trong 1 thuộc tính)
    - N: là tổng số lượng phần tử ở node đó

**B2:** Tính chỉ số gini\_index.

Gini\_index = gini(p) - gini()

* + - gini(p): chỉ số gini ở node cha
    - K: số node con được tách ra
    - gini(ck): chỉ số gini ở node con thứ k
    - M: số phần tử ở node p
    - mi : là số phần tử ở node con thứ i
* chọn ra thuộc tính có gini\_index lớn nhất 🡪 Ginisplit nhỏ nhất

gini()

**B3:** Chia dữ liệu vào các nút con tương ứng với các giá trị của thuộc tính, thực hiện lặp tìm ra thuộc tính tốt nhất trong tập dữ liệu mới.

* **Điều kiện dừng:**
  + TH1: nếu tất cả các mẫu trong node con thuộc cùng 1 lớp C (entropy=0) thì node đó đc gán nhãn C
  + TH2: nếu node con là rỗng (không có dữ liệu) thì node đó sẽ đc gán = nhãn phổ biến nhất trong tập S
  + TH3: nếu không còn thuộc tính nào để phân chia (các thuộc tính đều đc xét hết) -> node lá đó sẽ đc gán = nhãn phổ biến nhất

# Phần 2: Ứng dụng trong thực tế

## 1. .Mô tả bài toán

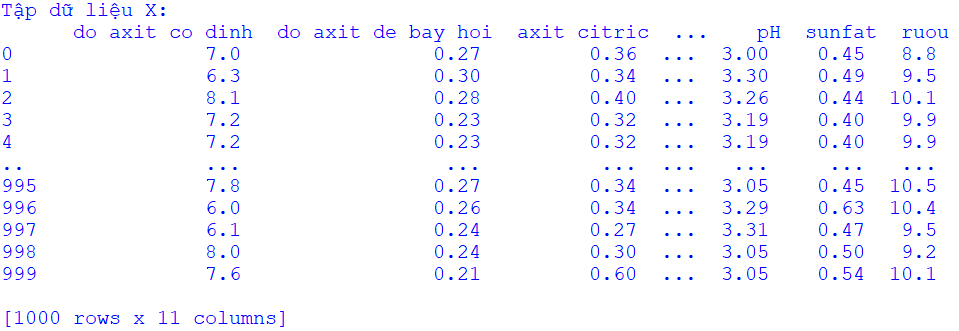
* Tên bài toán: Dự đoán chất lượng rượu vang Vinho Verde
* Mục đích của bài toán: dự đoán chất lượng rượu khi biết các thông tin liên quan.
* Input:
  + Dữ liệu có liên quan đến các biến thể màu trắng của rượu vang “Vinho Verde” của Bồ Đào Nha:

1. Độ axit cố định
2. Độ axit dễ bay hơi
3. Axit citric
4. Đường dư
5. Clorua
6. Lưu huỳnh đioxit tự do
7. Tổng lưu huỳnh đioxit
8. Tỷ trọng
9. pH
10. Sunfat
11. Rượu

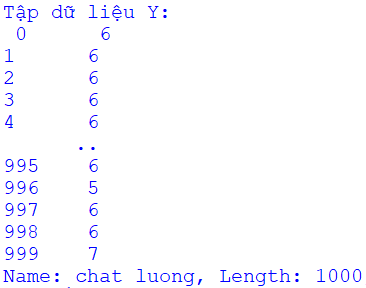
* Ouput: Chất lượng rượu (đạt(yes) hoặc không đạt(no)).
* Tóm tắt công việc thực hiện của bài toán:
  + Đọc tập dữ liệu của bài toán
* Chia dữ liệu thành 2 phần: 70% cho tập train, 30% cho tập test
* Sử dụng phương pháp:
  + - thuật toán Perceptron để tìm ra
    - ID3 xây dựng cây quyết định dựa vào hàm “entropy”
    - CART xây dựng cây quyết định dựa vào chỉ số “gini index”
* Huấn luyện mô hình và dự đoán nhãn

## 2. Mô tả tập dữ liệu của bài toán

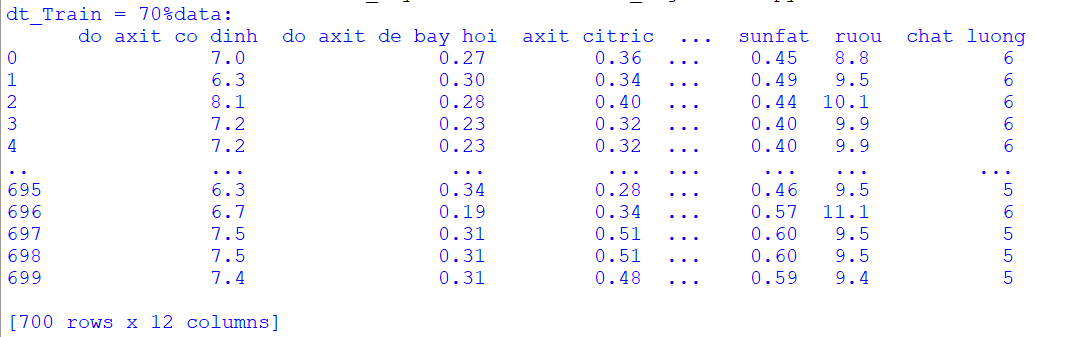
* Số lượng vector dữ liệu(số lượng mẫu rượu Vinho Verde): 1000
* Mỗi vector dữ liệu(mỗi mẫu rượu Vinho Verde) gồm 11 thông tin: Độ axit cố định, Độ axit dễ bay hơi, Axit citric, Đường dư, Clorua, Lưu huỳnh đioxit tự do, Tổng lưu huỳnh đioxit, Tỷ trọng, pH, Sunfat, Rượu.
* Nhãn lớp của dữ liệu (thông tin cần dự đoán) là chất lượng rượu, mỗi vector dữ liệu sẽ tương ứng với 1 giá trị chất lượng rượu (đạt hoặc không đạt).
* Tập dữ liệu X [11,1000] là 1 ma trận dữ liệu đầu vào với mỗi hàng của X là 1 vector dữ liệu gồm các thông tin: Độ axit cố định, Độ axit dễ bay hơi, Axit citric, Đường dư, Clorua, Lưu huỳnh đioxit tự do, Tổng lưu huỳnh đioxit, Tỷ trọng, pH, Sunfat, Rượu.



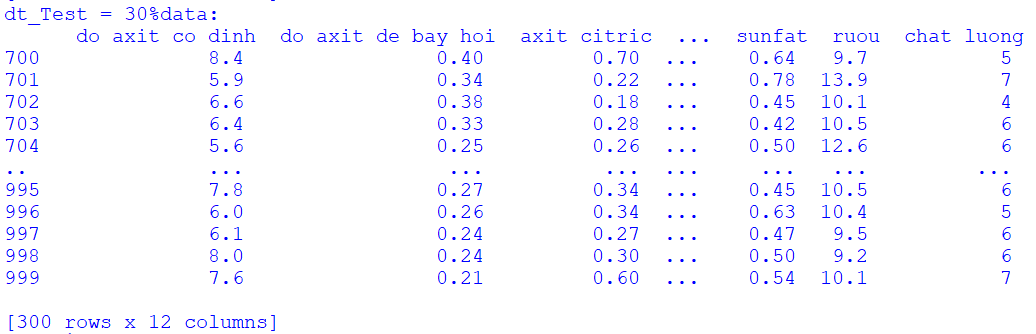
* Nhãn lớp Y là 1 vector có số dòng là 1000 và có 1 cột là “chất lượng”.



* Chia tập dữ liệu thành 2 phần: 70% dùng để huấn luyện mô hình, 30% dùng để kiểm tra sự phù hợp của mô hình.
  + 70% Dữ liệu sd để huấn luyện:

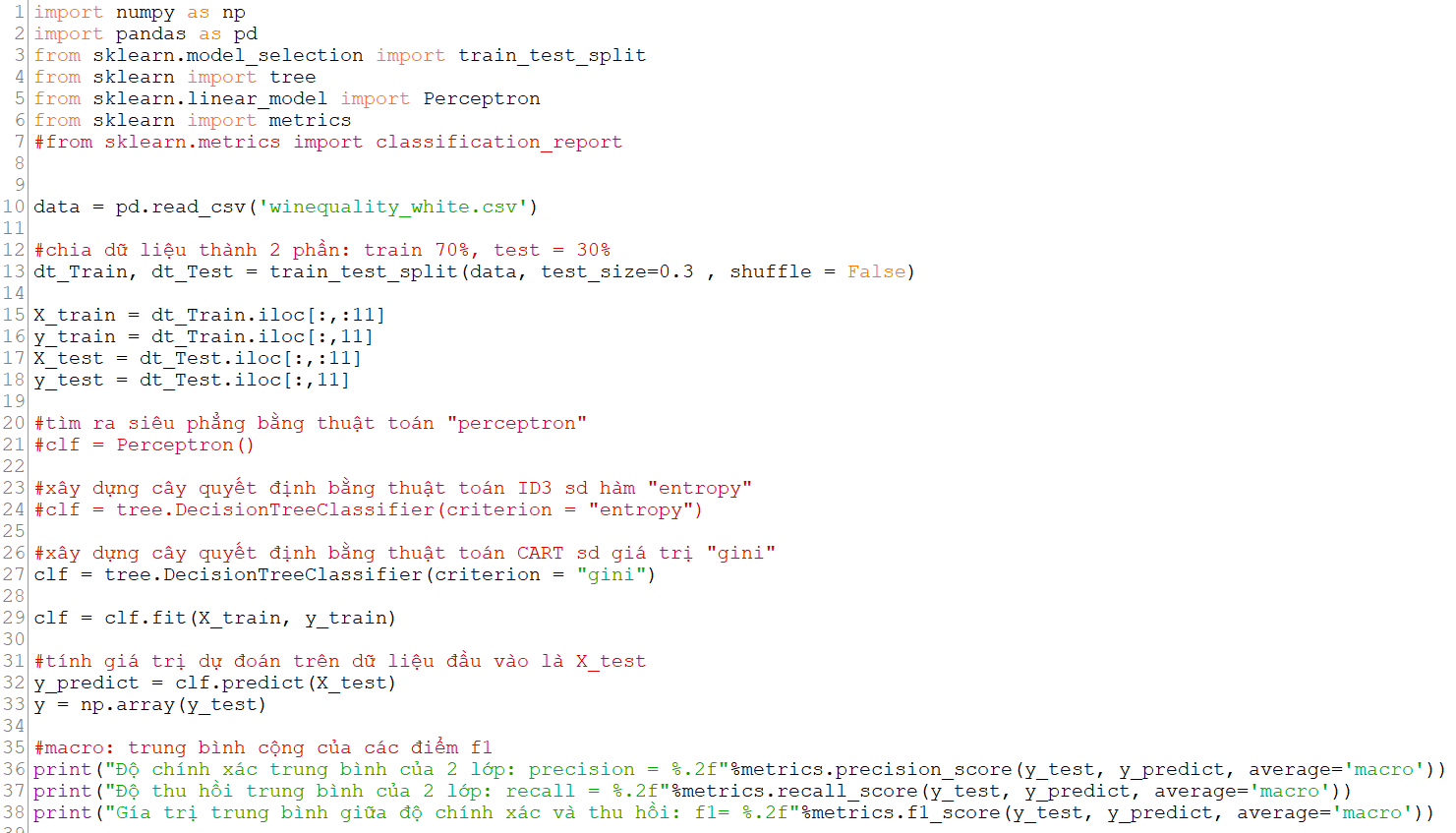


* + 30% Dữ liệu sd để kiểm tra sự phù hợp của mô hình:



## 3. Viết ứng dụng

* Viết code của các thuật toán Perceptron, ID3 và CART để giải quyết bài toán.



## 4. Phân tích kết quả của chương trình

a) Thuật toán Perceptron

* Tỷ lệ dự đoán đúng trên tập test: 0.39
* Tỷ lệ dự đoán sai trên tập test: 0.61

b) thuật toán ID3

* Tỷ lệ dự đoán đúng trên tập test: 0.69
* Tỷ lệ dự đoán sai trên tập test: 0.31

c) thuật toán Cart

* Tỷ lệ dự đoán đúng trên tập test: 0.73
* Tỷ lệ dự đoán sai trên tập test:0.27
* Thuật toán phù hợp nhất với bài toán là CART vì có tỉ lệ dự đoán đúng cao nhất trong 3 thuật toán.

## 5. Kết luận

* Hiểu được cách tìm siêu phẳng bằng thuật toán perceptron để dự đoán nhãn của mẫu dữ liệu cho trước.
* Hiểu được các xây dựng cây quyết định dựa trên thuật toán ID3 và CART để dự đoán nhãn của mẫu dữ liệu cho trước.

## 6. Tài liệu tham khảo

* <https://archive.ics.uci.edu/ml/datasets/Wine+Quality>
* <https://machinelearningcoban.com/2016/12/28/linearregression/>