## **2.1 Bài toán phân loại bệnh cây đậu nành sử dụng kỹ thuật SVM**

### **2.1.1 Nguồn dữ liệu**

- Nguồn : https://archive.ics.uci.edu/

-Bộ dữ liệu: soybean-large.csv

-Gồm 19 lớp ,35 thuộc tính ,307 dòng

+35 thuộc tính gồm:

1. date

2. plant-stand

3. precip

4. temp

5. hail

6. crop-hist

7. area-damaged

8. severity

9. seed-tmt

10. germination

11. plant-growth

12. leaves

13. leafspots-halo

14. leafspots-marg

15. leafspot-size

16. leaf-shread

17. leaf-malf

18. leaf-mild

19. stem

20. lodging

21. stem-cankers

22. canker-lesion

23. fruiting-bodies

24. external decay

25. mycelium

26. int-discolor

27. sclerotia

28. fruit-pods

29. fruit spots

30. seed

31. mold-growth

32. seed-discolor

33. seed-size

34. shriveling

35. roots

+19 Lớp :

diaporthe-stem-canker: 10

charcoal-rot: 10

rhizoctonia-root-rot: 10

phytophthora-rot : 40

brown-stem-rot: 20

powdery-mildew: 10

downy-mildew: 10

brown-spot: 40

bacterial-blight: 10

bacterial-pustule: 10

purple-seed-stain: 10

anthracnose: 20

phyllosticta-leaf-spot:10

alternarialeaf-spot: 40

frog-eye-leaf-spot: 40

diaporthe-pod-&-stem-blight: 6

cyst-nematode: 6

herbicide-injury: 4

Tổng :307

### **2.1.2 Bài toán học máy**

Bài toán phân loại bệnh cây đậu nành sử dụng kỹ thuật SVM

Phương pháp đánh giá: Đánh giá mô hình học dựa trên kết quả dự đoán (với độ đo đơn giản Accuracy

#### **2.2.1 Xử lý dữ liệu**

-Bộ dữ liệu: soybean-large.csv

Dữ liệu mất mát : Có

Dùng hàm dropna() để xử lý :

+Hàm Dropna(**axis*=0*, how*='any'*, thresh*=None*, subset*=None*, inplace*=False***):

Loại bỏ các giá trị bị thiếu.

**Thông số**

**Axis:** {0 hoặc 'chỉ mục', 1 hoặc 'cột'}, mặc định là 0

-Xác định xem các hàng hoặc cột chứa các giá trị bị thiếu có bị xóa hay không.

-0, hoặc 'chỉ mục': Bỏ các hàng chứa các giá trị bị thiếu.

-1 hoặc 'cột': Bỏ các cột chứa giá trị bị thiếu.

**How:** {‘any’, ‘all’}, mặc định ‘any’

-Xác định xem hàng hoặc cột có bị xóa khỏi DataFrame không khi chúng ta có ít nhất một NA hoặc tất cả NA.

-' any': Nếu có bất kỳ giá trị NA nào, hãy bỏ hàng hoặc cột đó.

-'all': Nếu tất cả các giá trị là NA, hãy bỏ hàng hoặc cột đó.

**Thresh:** int, tùy chọn

-Yêu cầu nhiều giá trị không phải NA.

**Subset: *array-like, optional***

**-**Các nhãn dọc theo trục khác để xem xét, ví dụ: nếu bạn đang bỏ hàng, đây sẽ là danh sách các cột cần bao gồm.

**Inplace: *bool, default False***

**-**Nếu Đúng, thực hiện thao tác tại chỗ và trả về None.

Code : df = df.dropna()

Kết quả

+Dữ liệu trước xử lý :307

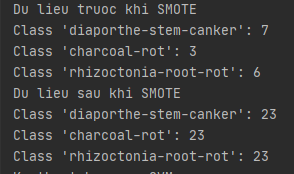
+Sau khi xử lý : 266

-Cân bằng dữ liệu SMOTE:

sm = SMOTE(random\_state = 2,k\_neighbors=1)

X\_train\_res, y\_train\_res = sm.fit\_sample(X\_train, y\_train)

+Kết quả

* 

#### **2.2.2 Huấn luyện**

-Sử dụng kỹ thuật học máy SVM:

**THUẬT TOÁN SVM**

**-Giới thiệu**

Bài toán phân lớp (*Classification*) và dự đoán (*Prediction*) là hai bài toán cơ bản và có rất nhiều ứng dụng trong tất cả các lĩnh vực nhơ: học máy, nhận dạng, trí tuệ nhân tạo, .v.v . Trong khóa luận này, chúng em sẽ đi sâu nghiên cứu phương pháp Support Vector Machines (SVM), một phương pháp rất hiệu quả hiện nay.

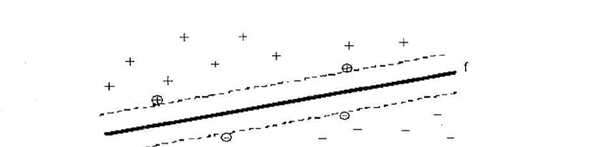
Phương pháp SVM được coi là công cụ mạnh cho những bài toán phân lớp phi tuyến tính được các tác giả Vapnik và Chervonenkis phát triển mạnh mẽ năm 1995. Phương pháp này thực hiện phân lớp dựa trên nguyên lý Cực tiểu hóa Rủi ro có Cấu trúc SRM (*Structural Risk Minimization*), được xem là một trong các phương pháp phân lớp giám sát không tham số tinh vi nhất cho đến nay. Các hàm công cụ đa dạng của SVM cho phép tạo không gian chuyển đổi để xây dựng mặt phẳng phân lớp

**- 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

**- Ý tưởng của phương pháp**

Cho trước một tập huấn luyện, được biểu diễn trong không gian vector, trong đó mỗi tài liệu là một điểm, phương pháp này tìm ra một siêu phẳng quyết định tốt nhất có thể chia các điểm trên không gian này thành hai lớp riêng biệt tương ứng là lớp + và lớp -. Chất lượng của siêu phẳng này được quyết định bởi khoảng cách (gọi là biên) của điểm dữ liệu gần nhất của mỗi lớp đến mặt phẳng này. Khi đó, khoảng cách biên càng lớn thì mặt phẳng quyết định càng tốt, đồng thời việc phân loại càng chính xác.

Mục đích của phương pháp SVM là tìm được khoảng cách biên lớn nhất, điều này được minh họa như sau:

**-Nội dung phương pháp**

+Cơ sở lý thuyết

|  |  |
| --- | --- |
|  |  |
|  |  |

SVM thực chất là một bài toán tối ưu, mục tiêu của thuật toán này là tìm được một không gian F và siêu phẳng quyết định f trên F sao cho sai số phân loại là thấp nhất.

Cho tập mẫu (x1,y1), (x2, y2), … (xf, yf )} với xi  Rn , thuộc vào hai lớp nhãn: yi {-1,1} là nhãn lớp tương ứng của các xi (-1 biểu thị lớp I, 1 biểu thị lớp

II).

Ta có, phương trình siêu phẳng chứa vectơ xitrong không gian:

xi .w + b = 0

+1, Xi . W + b > 0

Đặt f(Xi) = sign (Xi . W + b) =

-1, Xi . W + b < 0

Như vậy, f(Xi) biểu diễn sự phân lớp của Xivào hai lớp như đã nêu. Ta nói yi= +1 nếu Xi € lớp I và yi = -1 nếu Xi€ lớp II . Khi đó, để có siêu phẳng f ta sẽ phải giải bài toán sau:

ur

Tìm min *w* với W thỏa mãn điều kiện sau:

yi(sin (Xi.W + b)) ≥ 1 với i € 1,n

Bài toán SVM có thể giải bằng kỹ thuật sử dụng toán tử Lagrange để biến đổi về thành dạng đẳng thức. Một đặc điểm thú vị của SVM là mặt phẳng quyết định chỉ phụ thuộc các Support Vector và nó có khoảng cách đến mặt phẳng quyết

ur

định là 1/ *w* . Cho dù các điểm khác bị xóa đi thì thuật toán vẫn cho kết quả giống như ban đầu. Đây chính là điểm nổi bật của phương pháp SVM so với các phương pháp khác vì tất cả các dữ liệu trong tập huấn luyện đều được dùng để tối ưu hóa kết quả.

***TÓM LẠI:*** trong trương hợp nhị phân phân tách tuyến tính, việc phân lớp được thực hiện qua hàm quyết định *f(x) = sign(<w.x> + b),* hàm này thu được bằng việc thay đổi vectơ chuẩn *w*, đây là vectơ để cực đại hóa viền chức năng

Việc mở rộng SVM để phân đa lớp hiện nay vẫn đang được đầu tư nghiên cứu. Có một phương pháp tiếp cận để giải quyết vấn để này là xây dựng và kết hợp nhiều bộ phân lớp nhị phân SVM (Chẳng hạn: trong quá trình luyện với SVM, bài toán phân m lớp có thể được biến đổi thành bài toán phân 2\*m lớp, khi đó trong mỗi hai lớp, hàm quyết định sẽ được xác định cho khả năng tổng quát hóa tối đa). Trong phương pháp này có thể đề cập tới hai cách là *một-đổi-một, một-đối-tất cả*

+Bài toán nhiều phân lớp với 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.

Code:

from sklearn.svm import SVC

model2 = SVC().fit(X\_train,y\_train)

### **2.1.3 Đánh giá**

*-accuracy* (độ chính xác). Cách đánh giá này đơn giản tính tỉ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử.

-Với bài toán phân loại mà tập dữ liệu của các lớp là chênh lệch nhau rất nhiều, có một phép đo hiệu quả thường được sử dụng là Precision-Recall.

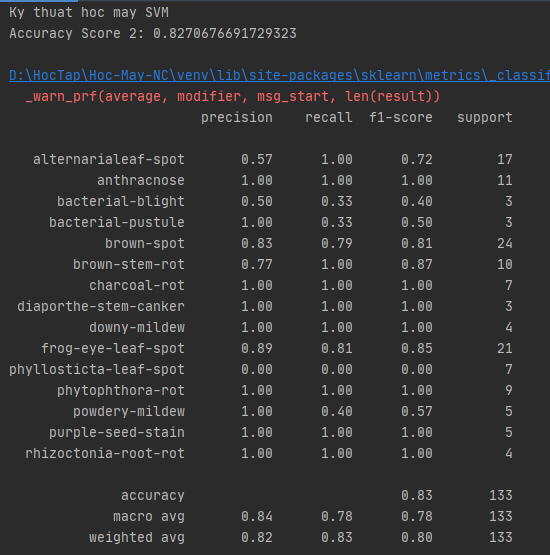
+**Precision** được định nghĩa là tỉ lệ số điểm **true positive** trong số những điểm **được phân loại là *positive*** (TP + FP).

**+Recall** được định nghĩa là tỉ lệ số điểm **true positive** trong số những điểm **thực sự là *positive*** (TP + FN).

+Precision cao đồng nghĩa với việc độ chính xác của các điểm tìm được là cao. Recall cao đồng nghĩa với việc True Positive Rate cao, tức tỉ lệ bỏ sót các điểm thực sự là thấp.

-F1-score, là *harmonic mean* của precision và recall (giả sử rằng hai đại lượng này khác không)

\*Kết quả đánh giá



\*Kết quả đánh giá với SMOTE (cân bằng dữ liệu)

