|  |
| --- |
| **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HỒ CHÍ MINH**  **KHOA CÔNG NGHỆ THÔNG TIN**  -----o0o----    **ĐỒ ÁN ĐỀ TÀI:**  **TÌM HIỂU KỸ THUẬT PHÂN LỚP SUPPORT VECTOR MACHINES VÀ XÂY DỰNG ỨNG DỤNG**  **Học phần: TRÍ TUỆ NHÂN TẠO**  **Thành phố Hồ Chí Minh, tháng 12 năm 2022** |

|  |
| --- |
| **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HỒ CHÍ MINH**  **KHOA CÔNG NGHỆ THÔNG TIN**  -----o0o----    **ĐỒ ÁN ĐỀ TÀI:**  **TÌM HIỂU KỸ THUẬT PHÂN LỚP SUPPORT VECTOR MACHINES VÀ XÂY DỰNG ỨNG DỤNG**  **Học phần: TRÍ TUỆ NHÂN TẠO**  Nhóm: 4  Danh sách sinh viên thực hiện:  STT HỌ TÊN MSSV  1 Dương Thị Phương Thi 2001207084  2 Trần Ngọc Hiếu Thảo 2001207001  3 Nguyễn Tô Bảo Ngọc 2001206998  Giảng viên hướng dẫn: ThS. Huỳnh Thị Châu Lan  **Thành phố Hồ Chí Minh, tháng 12 năm 2022** |

# PHÂN CÔNG NHIỆM VỤ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **HỌ TÊN** | **MSSV** | **NHIỆM VỤ** | **MỨC ĐỘ HOÀN THÀNH** |
| 1 | Dương Thị Phương Thi | 2001207084 | CHƯƠNG 2 | 100% |
| 2 | Trần Ngọc Hiếu Thảo | 2001207001 | CHƯƠNG 3 | 100% |
| 3 | Nguyễn Tô Bảo Ngọc | 2001206998 | CHƯƠNG 1 | 100% |

# LỜI CẢM ƠN

Nhóm chúng em xin chân thành gửi lời cảm ơn đến Ban lãnh đạo trường Đại học Công nghiệp Thực phẩm Thành phố Hồ Chí Minh nói chung và đặc biệt là các thầy cô bộ môn công nghệ nói riêng đã tạo điều kiện cho chúng em có cơ hội học tập, tiếp xúc với các phần mềm mang tính ứng dụng thực tế cao để chúng em có thể dần làm quen với môi trường làm việc sau này.

Đặc biệt, chúng em xin chân thành cảm ơn Huỳnh Thị Châu Lan. Nhờ sự giúp đỡ tận tình và góp ý của cô từ khi bắt đầu đến lúc kết thúc đồ án. Qua đó chúng em đã hoàn thành đúng thời hạn và tích lũy được nhiều kiến thức chuyên ngành hơn.

Trong quá trình làm đồ án, chúng em đã cố gắng hoàn thành hết mức có thể tuy nhiên kiến thức và độ hiểu biết còn có hạn nên việc chúng em mắc phải những sai sót nhất định là không thể tránh khỏi. Rất mong sự chia sẻ, cảm thông và được cô và các bạn đóng góp ý kiến tận tình để nhóm chúng em học thêm nhiều kinh nghiệm và sẽ hoàn thành tốt hơn bài báo cáo tốt nghiệp sắp tới.

Nhóm 7 chúng em xin chân thành cảm ơn!

Thành phố Hồ Chí Minh, ngày 03/12/2022

Nhóm OK

# 

MỤC LỤC

[PHÂN CÔNG NHIỆM VỤ 1](#_Toc122357614)

[LỜI CẢM ƠN 2](#_Toc122357615)

[PHẦN MỞ ĐẦU 3](#_Toc122357616)

[*1. Lý do chọn đề tài* 3](#_Toc122357617)

[*2. Mục tiêu nghiên cứu* 3](#_Toc122357618)

[*3. Nội dung nghiên cứu* 3](#_Toc122357619)

[*4. Phương pháp nghiên cứu* 3](#_Toc122357620)

[*5. Bố cục bài đề tài* 3](#_Toc122357621)

[PHẦN NỘI DUNG 4](#_Toc122357622)

[CHƯƠNG 1: TÌM HIỂU VỀ SUPPORT VECTOR MACHINES 4](#_Toc122357623)

[1. Giới thiệu 4](#_Toc122357624)

[1.1 Định nghĩa 4](#_Toc122357625)

[1.2 Ứng dụng 4](#_Toc122357626)

[2. Ý tưởng phương pháp 4](#_Toc122357627)

[2.1 Khoảng cách từ một điểm tới một siêu mặt phẳng 4](#_Toc122357628)

[2.2 Nhắc lại bài toán phân chia hai lớp dữ liệu 5](#_Toc122357629)

[3. Cơ sở lý thuyết 7](#_Toc122357630)

[3.1 Xây dựng bài toán tối ưu SVM 7](#_Toc122357631)

[3.2 Xác định lớp cho một điểm dữ liệu mới 9](#_Toc122357632)

[CHƯƠNG 2 : CÁC BÀI TOÁN PHÂN LỚP 10](#_Toc122357633)

[1. Bài Toán Tối Ưu (Hard – margin) 10](#_Toc122357634)

[2. Tệp dữ liệu phân chia tuyến tính có nhiễu (Soft Margin) 11](#_Toc122357635)

[3. Tập dữ liệu phân chia phi tuyến (Sử dụng phương pháp Kernel) 12](#_Toc122357636)

[CHƯƠNG 3 : ỨNG DỤNG 14](#_Toc122357637)

[1. Ứng dụng SVM vào phân loại hoa. 14](#_Toc122357638)

[1.1 Mô tả bài toán phân loại hoa 14](#_Toc122357639)

[1.2 Tập dữ liệu Iris Flowers 14](#_Toc122357640)

[1.3 Biểu diễn tập dữ liệu bằng đồ thị 2D 15](#_Toc122357641)

[1.4 Phân lớp sử dụng SVM với các Kernel khác nhau 17](#_Toc122357642)

[1.5 Dự đoán tập data mới 19](#_Toc122357643)

[1.6 Hướng phát triển 22](#_Toc122357644)

[2. Ứng dung SVM vào phân loại hạt thóc. 23](#_Toc122357645)

[2.1 Chuẩn bị dữ liệu 23](#_Toc122357646)

[2.2 Phân chia thành tập train và tập test 26](#_Toc122357647)

[2.3 Huấn luyện và Đánh giá 27](#_Toc122357648)

[2.4 Sử dụng đặc trưng HOG 28](#_Toc122357649)

[2.5 Trích xuất đặc trưng 29](#_Toc122357650)

[2.6 Huấn luyện và phân loại bằng SVM 31](#_Toc122357651)

[2.7 Kết Luận 31](#_Toc122357652)

[TÀI LIỆU THAM KHẢO 32](#_Toc122357653)

# PHẦN MỞ ĐẦU

## 1. Lý do chọn đề tài

Hiện nay, với một lượng lớn các dữ liệu thì phân lớp dữ liệu có vai trò rất quan trọng, là một trong những bài toán luôn thời sự trong lĩnh vực xử lý dữ liệu văn bản. Một yêu cầu cơ bản được đặt ra là cần tăng tính hiệu quả của thuật toán phân lớp, nâng cao giá trị của các độ đo hồi tưởng, chính xác của thuật toán. Mặt khác, nguồn tài nguyên về ví dụ học có nhãn không phải luôn được đáp ứng vì vậy cần có các thuật toán phân lớp sử dụng các ví dụ chưa có nhãn. Phân lớp bán giám sát đáp ứng được hai yêu cầu nói trên

Vấn đề phân lớp và dự đoán là khâu rất quan trọng trong học máy và trong khai phá dữ liệu, phát hiện trí thức. Kỹ thuật Support Vector Machines (SVM) được đánh giá là công cụ mạnh và tinh vi nhất hiện nay cho những bài toán phân lớp phi tuyến. Từ các công trình khoa học được công bố cho thấy phương pháp SVM có khả năng phân loại khá tốt đối với bài toán phân loại văn bản cũng như trong nhiều ứng dụng khác.

## 2. Mục tiêu nghiên cứu

Đề tài thảo luận mục đích sẽ nghiên cứu phần bài toán phân lớp quan điểm, cơ sở lý thuyết của phương pháp SVM và các vấn đề liên quan.

Trình bày hướng áp dụng kỹ thuật SVM cũng nhờ những cải tiến, mở rộng của nó vào giải quyết một số các bài toán ứng dụng trong thực tiễn.

Trình bày tổng quan về bài toán phân lớp quan điểm và cụ thể là bài toán phân lớp phân cực để phân chia các loại hoa trọng tập dữ liệu được đào tạo Iris.

Tìm hiểu dữ liệu quan điểm và viết chương trình thử nghiệm với kỹ thuật Support Vector Machines (SVM) đã giải thích ở trên.

## 3. Nội dung nghiên cứu

Phân tích, nghiên cứu và tìm hiểu kỹ thuật phân lớp support vector machines và xây dựng ứng dụng

## 4. Phương pháp **nghiên cứu**

Phương pháp nghiên cứu lý luận: nghiên cứu và tìm hiểu kỹ thuật phân lớp support vector machine và xây dựng ứng dụng qua các tài liệu, văn kiện, văn bản,…

Phương pháp nghiên cứu xử lý dữ kiện: xử lý kết quả thu thập được từ việc nghiên cứu, điều tra, khảo sát.

Phương pháp phân tích tổng hợp: phân tích các thông tin thu thập được nhằm phát hiện và khai thác các khía cạnh khác nhau; tổng hợp lại các thông tin đã cố nhằm phát hiện ra sự sai lệch, nêu ra những tác động nhằm đưa ra biện pháp phù hợp về bản chất của vấn đề nghiên cứu.

## 5. Bố cục bài đề tài

Ngoài phần mở đầu và kết luận, nội dung của đề tài được trình bày trong 2 chương có cấu trúc như sau:

Chương 1: Tìm hiểu về support vector machine

Chương 2: Xây dựng chương trình ứng dụng SVM vào thực tiễn

Ứng dụng với phân loại hoa với các loại kernel

Ứng dụng với phân loại hạt giống kết hợp phương pháp HOG với SVM

# 

# PHẦN NỘI DUNG

# CHƯƠNG 1: TÌM HIỂU VỀ SUPPORT VECTOR MACHINES

## 1. Giới thiệu

### 1.1 Định nghĩa

Support vector machine (SVM) là một trong những thuật toán phân lớp phổ biến và hiệu quả. Ý tưởng đứng sau SVM khá đơn giản, nhưng để hiểu được cách tìm nghiệm của nó, chúng ta cần một chút kiến thức về tối ưu và duality.

### 1.2 Ứng dụng

* Nhận dạng tiếng nói
* Phân loại văn bản
* Phân tích dữ liệu gen
* Phân tích dữ liệu Marketing

## 2. Ý tưởng phương pháp

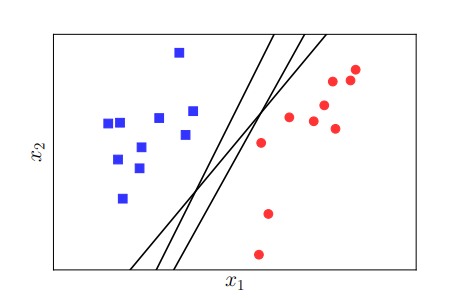
### 2.1 Khoảng cách từ một điểm tới một siêu mặt phẳng

Trong không gian hai chiều, khoảng cách từ một điểm có toạ độ (x0, y0) tới đường thẳng có phương trình được xác định bởi

Trong không gian ba chiều, khoảng cách từ một điểm có toạ độ (x0, y0, z0) tới một mặt phẳng có phương trình được xác định bởi

Hơn nữa, nếu bỏ dấu trị tuyệt đối ở tử số, ta có thể xác định được điểm đó nằm về phía nào của đường thẳng hay mặt phẳng đang xét. Những điểm làm cho biểu thức trong dấu giá trị tuyệt đối mang dấu dương nằm về cùng một phía (tạm gọi là phía dương), những điểm làm cho giá trị này mang dấu âm nằm về phía còn lại (gọi là phía âm). Những điểm nằm trên đường thẳng/mặt phẳng sẽ làm cho tử số có giá trị bằng 0, tức khoảng cách bằng 0.

Các công thức này có thể được tổng quát lên cho trường hợp không gian d chiều. Khoảng cách từ một điểm (vector) có toạ độ tới siêu mặt phẳng (hyperplane) có phương trình được xác định bởi



**Hình 2.1:** *Hai lớp dữ liệu đỏ và xanh là linearly separable. Có vô số các đường thằng có thể phân tách chính xác hai lớp dữ liệu này*

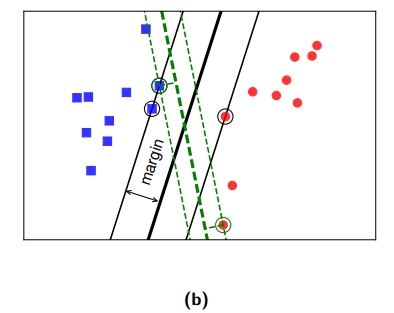
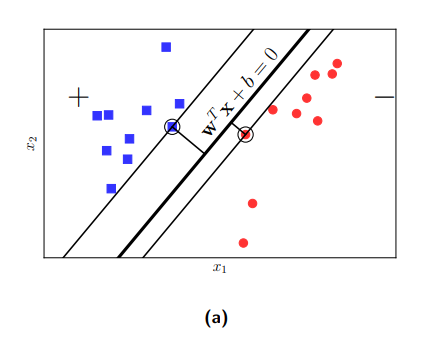
### 2.2 Nhắc lại bài toán phân chia hai lớp dữ liệu

Chúng ta cùng quay lại với bài toán phân lớp Perceptron Learning Algorithm (PLA). Giả sử rằng có hai lớp dữ liệu được mô tả bởi các điểm (feature vector) trong không gian nhiều chiều, hơn nữa, hai lớp dữ liệu này là linearly separable, tức tồn tại một siêu phẳng phân chia chính xác hai lớp đó. Hãy tìm một siêu phẳng phân chia hai lớp đó đó, tức tất cả các điểm thuộc một lớp nằm về cùng một phía của siêu phẳng đó và ngược phía với toàn bộ các điểm thuộc lớp còn lại. Chúng ta đã biết rằng, thuật toán PLA có thể làm được việc này nhưng nó có thể cho chúng ta vô số nghiệm như *Hình 2.1.*

Có một câu hỏi được đặt ra ở đây. Trong vô số các mặt phân chia đó, đâu là mặt tốt nhất. Trong ba đường thẳng minh họa trong *Hình 2.1*, có hai đường thẳng khá lệch về phía lớp màu đỏ. Điều này có thể khiến cho lớp màu đỏ không vui vì lãnh thổ bị lấn nhiều quá. Việc này có thể khiến cho các điểm màu đỏ trong tương lai bị phân lớp lỗi thành điểm màu xanh. Liệu có cách nào để tìm được đường phân chia mà cả hai lớp đều cảm thấy công bằng và hạnh phúc nhất hay không?

Để trả lời câu hỏi này, chúng ta cần tìm một tiêu chuẩn để đo sự hạnh phúc của mỗi lớp. Nếu ta định nghĩa mức độ hạnh phúc của một lớp tỉ lệ thuận với khoảng cách gần nhất từ một điểm của lớp đó tới đường/mặt phân chia, ở *Hình 2.2a*, lớp màu đỏ sẽ không được hạnh phúc cho lắm vì đường phân chia gần nó hơn lớp màu xanh rất nhiều. Chúng ta cần một đường phân chia sao cho khoảng cách từ điểm gần nhất của mỗi lớp (các điểm được khoanh tròn) tới đường phân chia là như nhau, như thế thì mới công bằng. Khoảng cách như nhau này được gọi là biên hoặc lề (margin).

Đã có công bằng rồi, chúng ta cần thịnh vượng nữa. công bằng mà cả hai đều kém hạnh phúc như nhau thì chưa phải là thịnh vượng cho lắm.



**Hình 2.2:** *Ý tưởng của SVM. Margin của một lớp được định nghĩa là khoảng cách từ các điểm gần nhất của lớp đó tới mặt phân chia. Margin của hai lớp phải bằng nhau và lớn nhất có thể.*

Xét tiếp *Hình 2.2b* khi khoảng cách từ đường phân chia tới các điểm gần nhất của mỗi lớp là như nhau. Xét hai cách phân chia bởi đường nét liền màu đen và đường nét đứt màu lục, đường nào sẽ làm cho cả hai lớp *hạnh phúc hơn?* Rõ ràng đó phải là đường nét liền màu đen vì nó tạo ra một *margin* rộng hơn.

Việc *margin* rộng hơn sẽ mang lại hiệu ứng phân lớp tốt hơn vì *sự phân chia giữa hai lớp là rạch ròi hơn*. Bài toán tối ưu trong SVM chính là bài toán đi tìm đường phân chia sao cho *margin* giữa hai lớp là lớn nhất. Đây cũng là lý do vì sao SVM còn được gọi là *maximum margin classifier*. Nguồn gốc của tên gọi *support vector machine* sẽ sớm được làm sáng tỏ.

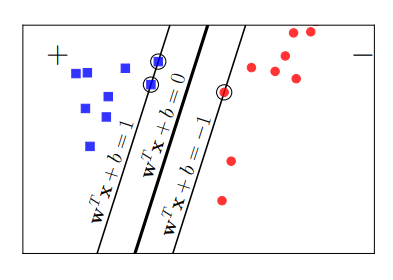
## 3. Cơ sở lý thuyết

### 3.1 Xây dựng bài toán tối ưu SVM

Giả sử rằng các cặp dữ liệu trong tập huấn luyện là với vector xi ∈ Rd thể hiện đầu vào của một điểm dữ liệu và là nhãn của điểm dữ liệu đó, d là số chiều của dữ liệu và N là số điểm dữ liệu. Giả sử rằng nhãn của mỗi điểm dữ liệu được xác định bởi hoặc giống như trong PLA.

Để dễ hình dung, chúng ta cùng làm với các ví dụ trong không gian hai chiều. Giả sử rằng các điểm màu xanh có nhãn là 1, các điểm tròn đỏ có nhãn là -1 và mặt là mặt phân chia giữa hai lớp (*Hình 3.1*). Hơn nữa, lớp màu xanh nằm về phía dương, lớp màu đỏ nằm về phía âm của mặt phân chia. Nếu ngược lại, ta chỉ cần đổi dấu của w và b. Ta cần đi tìm siêu phẳng được mô tả bởi các hệ số w và b.

Ta quan sát thấy một điểm quan trọng như sau. Với cặp dữ liệu (xn, yn) bất kỳ, khoảng cách từ điểm đó tới mặt phân chia là



**Hình 3.1:** *Giả sử mặt phân chia có phương trình. Không mất tính tổng quát, bằng cách nhân các hệ số w và b với các hằng số phù hợp, ta có thể giả sử rằng điểm gần nhất của lớp màu xanh tới mặt này thoả mãn Khi đó, điểm gần nhất của lớp đỏ thoả mãn*

Điều này có thể được nhận thấy vì theo giả sử ở trên, yn luôn cùng dấu với phía của xn. Từ đó suy ra yn cùng dấu với ,vì vậy tử số luôn là một đại lượng không âm. Với mặt phân chia này, margin được tính là khoảng cách gần nhất từ một điểm (trong cả hai lớp, vì cuối cùng margin của cả hai lớp sẽ như nhau) tới mặt đó, tức là

Bài toán tối ưu của SVM chính là việc tìm w và b sao cho margin này đạt giá trị lớn nhất:

(3.1)

Có một nhận xét quan trọng là nếu ta thay vector hệ số w bởi kw và b bởi kb trong đó k là một hằng số dương bất kỳ thì mặt phân chia không thay đổi, tức khoảng cách từ từng điểm đến mặt phân chia không đổi, tức margin không đổi. Vì vậy, ta có thể giả sử

với những điểm nằm gần mặt phân chia nhất (được khoanh tròn trong *Hình 3.1*)

Như vậy, với mọi n ta luôn có

Vậy bài toán tối ưu (3.1) có thể đưa về bài toán tối ưu có ràng buộc có dạng

Thỏa mãn :

(3.2)

Bằng một biến đổi đơn giản, ta có thể đưa bài toán này về dạng

Thỏa mãn :

(3.3)

Ở đây, chúng ta đã lấy nghịch đảo hàm mục tiêu, bình phương nó để được một hàm khả vi, và nhân với để biểu thức đạo hàm đẹp hơn.

Có một quan sát rất quan trọng. Trong bài toán (3.3), hàm mục tiêu là một norm, nên là một hàm lồi. Các hàm bất đẳng thức ràng buộc là các hàm tuyến tính theo w và b, nên chúng cũng là các hàm lồi. Vậy bài toán tối ưu (3.3) có hàm mục tiêu là lồi, và các hàm ràng buộc cũng là lồi, nên nó là một bài toán lồi. Hơn nữa, nó là một quadratic programming vì hàm mục tiêu là một quadratic form. Thậm chí, hàm mục tiêu là strictly convex vì và mathbfI là ma trận đơn vị – là một ma trận xác định dương. Từ đây có thể suy ra nghiệm cho SVM là duy nhất.

Đến đây thì bài toán này có thể giải được bằng các công cụ hỗ trợ tìm nghiệm cho quadratic programing, ví dụ CVXOPT. Tuy nhiên, việc giải bài toán này trở nên phức tạp khi số chiều d của không gian dữ liệu và số điểm dữ liệu N tăng lên cao. Thay vào đó, người ta thường giải bài toán đối ngẫu của bài toán này. Thứ nhất, bài toán đối ngẫu có những tính chất thú vị hơn khiến nó được giải một cách hiệu quả hơn. Thứ hai, trong quá trình xây dựng bài toán đối ngẫu, người ta thấy rằng SVM có thể được áp dụng cho những bài toán mà dữ liệu không nhất thiết linearly separable, như chúng ta sẽ thấy ở các chương sau của phần này.

### 3.2 Xác định lớp cho một điểm dữ liệu mới

Sau khi đã tìm được mặt phân cách , nhãn của bất kỳ một điểm nào sẽ được xác định đơn giản bằng

# CHƯƠNG 2 : CÁC BÀI TOÁN PHÂN LỚP

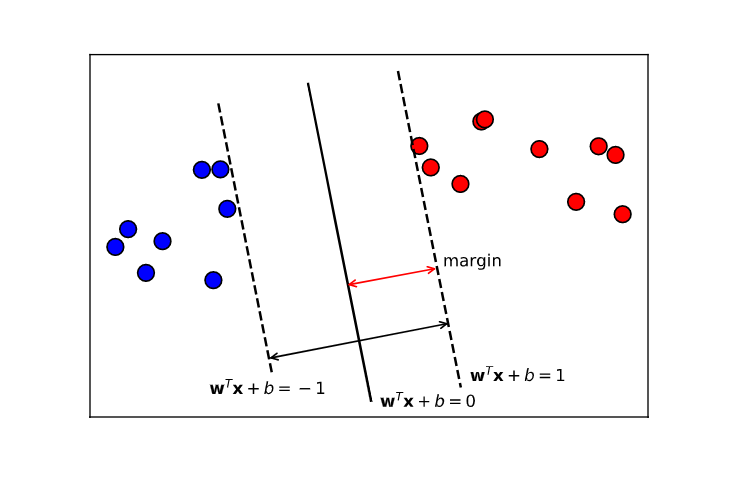
## 1. Bài Toán Tối Ưu (Hard – margin)

Bài toán tối ưu của SVM chính là việc tìm w và b sao cho margin này đạt giá trị lớn nhất

Với mọi n ta luôn có :

Vậy bài toán có thể đưa về bài toán tối ưu có ràng buộc

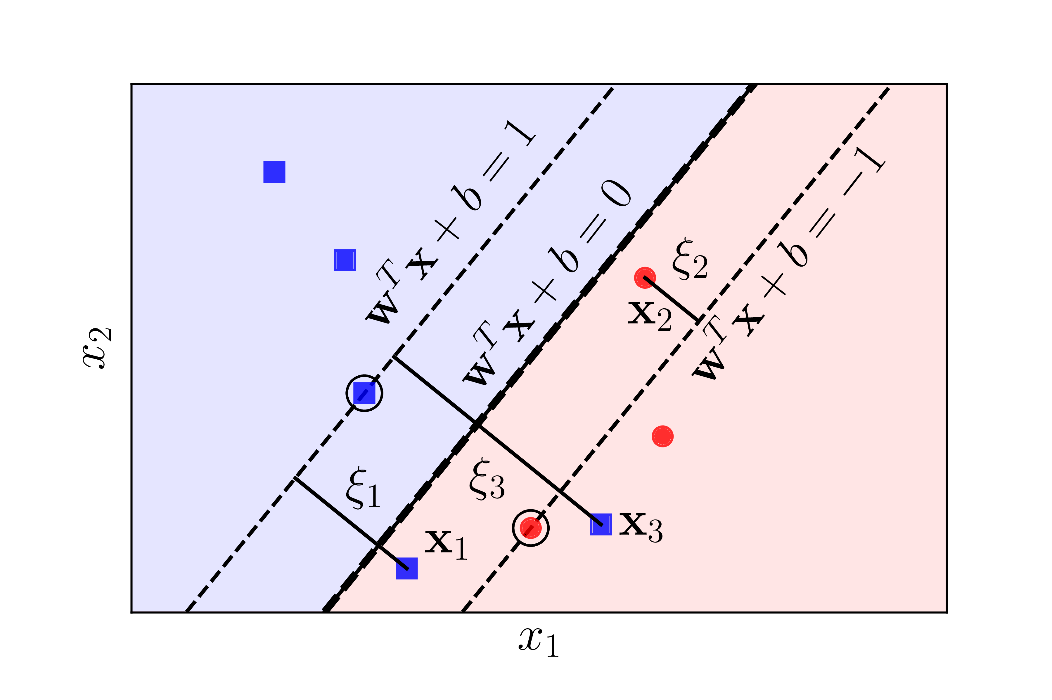
Thỏa mãn :



## 2. Tệp dữ liệu phân chia tuyến tính có nhiễu (Soft Margin)

Sử dụng biến Slack :

* Với những điểm nằm trong vùng an toàn
* Những điểm nằm trong vùng không an toàn nhưng vẫn đúng phía so với đường ranh giới
* Những điểm nằm ngược phía với class của chúng



Từ đó ta có hàm mục tiêu :

Thỏa mãn ràng buộc :

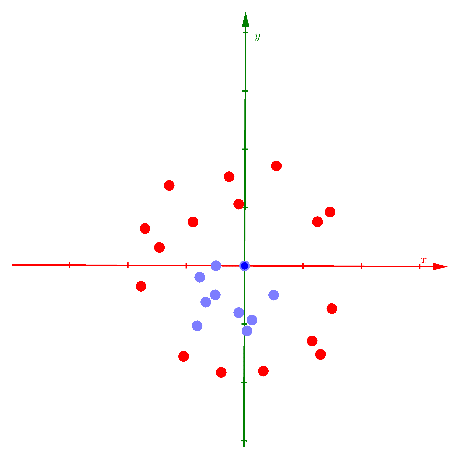
Với tham số C được quy ước :

* C lớn : cho phép sai lệch nhỏ, thu được margin nhỏ
* C nhỏ : cho phép sai lệch lớn, thu được margin lớn

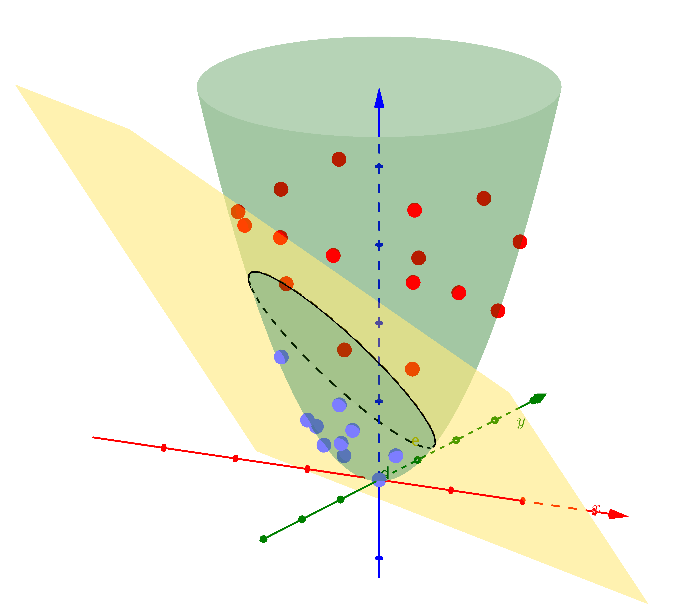
## 3. Tập dữ liệu phân chia phi tuyến (Sử dụng phương pháp Kernel)

Ví dụ về phương pháp Kernel trong không gian 2 chiều :

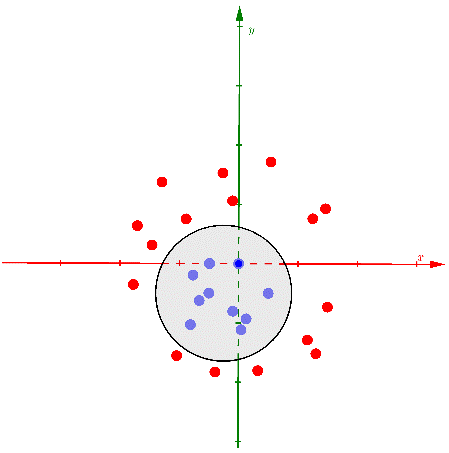
Dữ liệu của 2 lớp không phân biệt tuyến tính trong không gian 2 chiều



Nếu thêm chiều thứ 3 là một hàm số của 2 chiều còn lại các điểm dữ liệu đã được phân bố trên một mặt parabolic và trở nên tuyến tính và mặt màu vàng là mặt phân chia

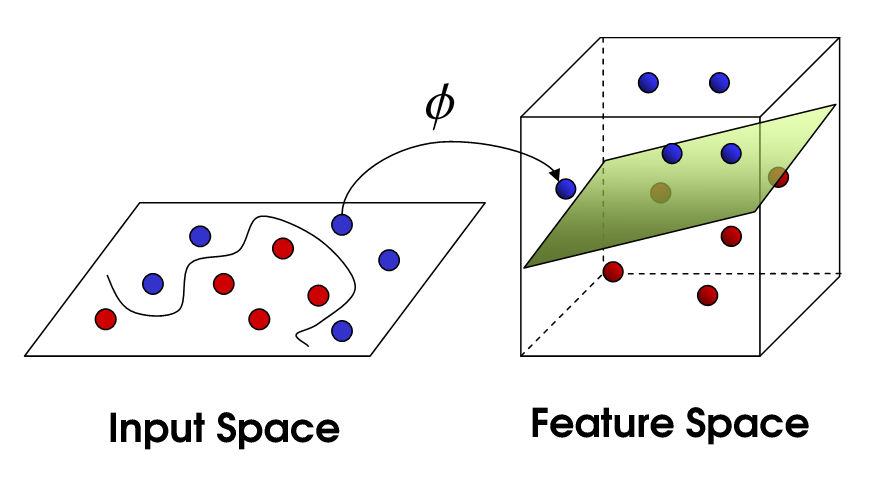


Giao của mặt phẳng phân chia và parabolic là 1 đường elip, khi chiếu toàn bộ xuông mặt phẳng 2 chiều ban đầu, sẽ tìm ra được đường phân 2 lớp



Phương pháp Kernel SVM là phương pháp đi tìm một hàm số biến đổi dữ liệu từ không gian đặc trưng ban đầu thành dữ liệu trong một không gian mới bằng một hàm số

|  |  |
| --- | --- |
| **Hàm** | **Công Thức** |
| Đa thức (Polynomial Kernels) |  |
| Gaoxo (Gaussian Kernels) |  |
| Sigmoid(Sigmoid Kernels) |  |



# CHƯƠNG 3: ỨNG DỤNG

## 1. Ứng dụng SVM vào phân loại hoa.

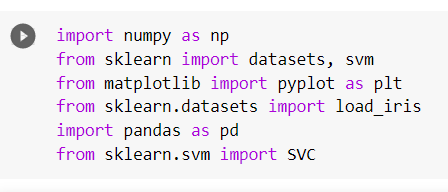
### 1.1 Mô tả bài toán phân loại hoa

Khi mô tả một bài toán thì chúng ta cần tìm ra cơ sở để có thể giải quyết bài toán đó. Như bài toán phân loại hoa vậy thì đối tượng mà chúng ta cần quan tâm đến đó là Hoa. Vậy làm sao chúng ta có thể phân loại ra các loài hoa với nhau?

Đến với câu hỏi này thì chúng ta cùng điểm lại trong quá trình chúng ta lớn lên thì chúng ta đã tích lũy cho mình những đặc điểm riêng của mỗi loài hoa. Ví dụ như là hoa Huệ thì sẽ có màu trắng cùng với mùi thơm đặc trưng của nó, hoa Hồng màu đỏ và có gai, và còn nhiều loài hoa khác nữa. Vậy thì giờ đây chúng ta đã có một tập dữ liệu các loài hoa và cũng chính là căn cứ để chúng ta có thể nhận dạng các loài hoa đó. Nguyên tắc học của máy tính cũng như vậy, chúng ta cần phải cung cấp cho nó một tập dữ liệu và huấn luyện để nó có thể căn cứ vào tập dữ liệu mà chúng ta cung cấp để nhận dạng được bông hoa đó thuộc loài hoa nào. Mấu chốt quan trọng là nằm ở tập dữ liệu và huấn luyện nó. Cho nên chúng ta sẽ sử dụng một tập dữ liệu đã có sẵn về hoa vô cùng nổi tiếng đó là tập dữ liệu Iris. Sau đây chúng ta hãy đi tìm hiểu sâu hơn về tập dữ liệu này.

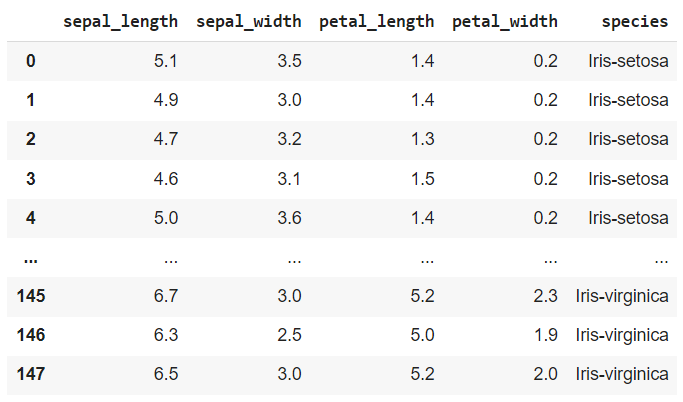
### 1.2 Tập dữ liệu Iris Flowers

Thư viện sử dụng

****

Tập dữ liệu Iris Flowers hay còn có tên gọi khác là Fisher’s Iris, được tạo ra do ông Ronald Fisher thu nhập và tổng hợp. Tập dữ liệu này chứa gồm 50 mẫu về 3 loài hoa khác nhau của họ Iris là (Iris setosa, Iris virginica và Iris versicolor).

Với mỗi mẫu hoa này, chúng ta thu nhập được 4 thuộc tính đó là chiều dài, chiều rộng, đài hoa, cánh hoa với đơn vị là centimet. Để mà chúng ta có thể sử dụng được cái tập dữ liệu này thì chúng ta sẽ sử dụng datasets trong sklearn.

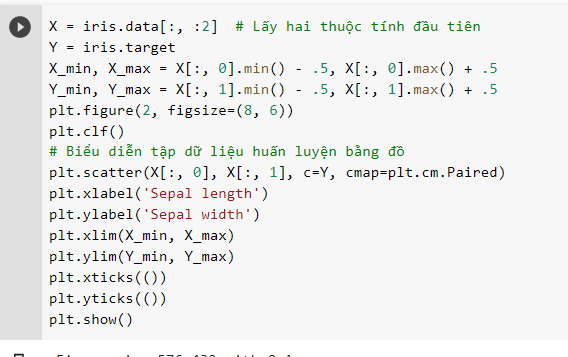


Sau khi mà chúng ta đã có tập dữ liệu thì chúng ta đã có thể chạy thuật toán SVM ngay để có thể tiến hành phân loại. Tuy nhiên, chúng ta có thể nhìn trực quan hơn về tập dữ liệu này. Để hiện thị cho nó trực quan nhất thì chúng ta nên dùng biểu đồ với thư viện matplotlib. Chúng ta xem thử biểu diễn trên đồ thị khi ta đưa tập dữ liệu mà chúng ta đã cung cấp cho máy tính.

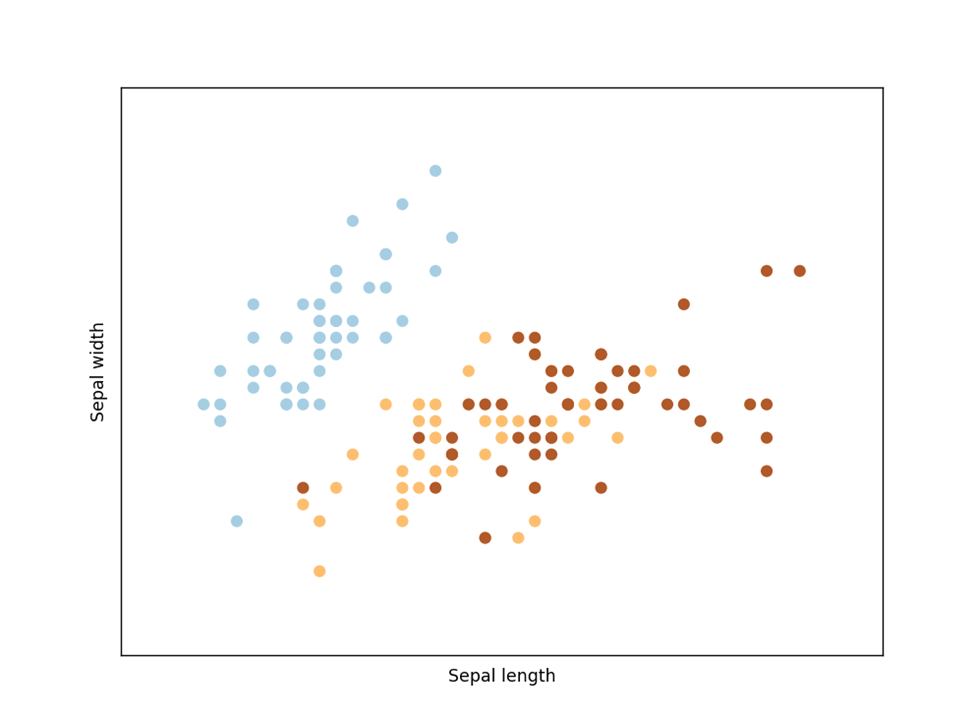
### 1.3 Biểu diễn tập dữ liệu bằng đồ thị 2D

Chúng ta thử tưởng tượng tập dữ liệu của chúng ta là một tập hợp của 150 điểm dữ liệu sẽ tương ứng với 50 điểm mỗi loại. Chúng ta lấy từ datasets bằng hàm sau :

Lúc này thì chúng ta đã có một tập hợp các điểm dữ liệu. Mỗi điểm dữ liệu sẽ bao gồm 4 thuộc tính như đã nói ở trên. Tuy nhiên để có thể biểu diễn trong đồ thị hai chiều thì cần phải giảm bớt số thuộc tính biểu diễn. Ở đây giả sử chọn hai thuộc tính đầu tiên là độ rộng và chiều cao của đài hoa. Hàm xử lý vẽ đồ thị 2D như sau:



Và đây là kết quả của nó :



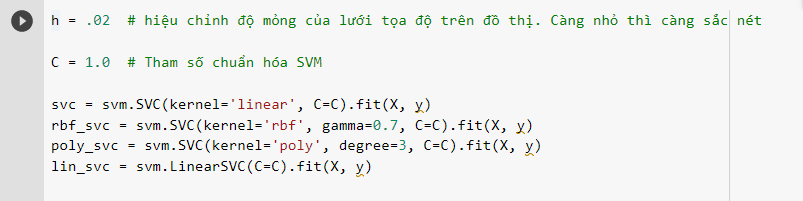
Chúng ta có thể thấy rằng những điểm dữ liệu với hai thuộc tính trên đồ thị hai chiều. Các điểm này đều được phân biệt bằng mắt thường với 3 màu khác nhau. Tuy nhiên, để máy tính có thể phân biệt được lại là một câu chuyện hoàn toàn khác.

### 1.4 Phân lớp sử dụng SVM với các Kernel khác nhau

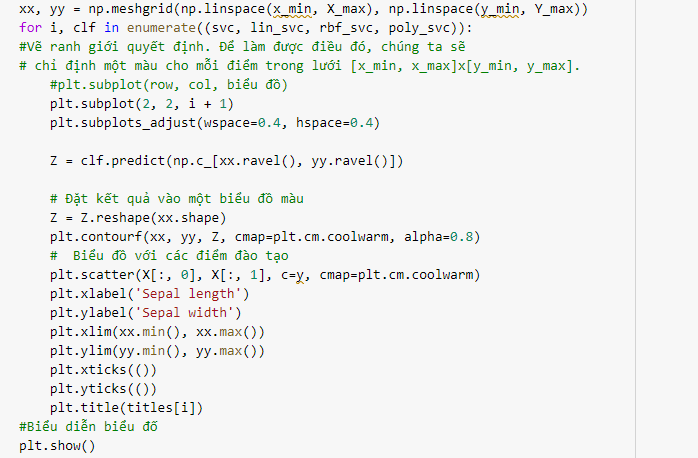
Với tập dữ liệu **Iris** chúng ta cần phải phân loại các bông hoa thành 3 lớp dữ liệu. Và sử dụng SVM với các phương pháp khác nhau sẽ cho hiệu quả phân lớp khác nhau. Cũng tương tự như trên, chúng ta chỉ xem xét đến 2 thuộc tính đầu tiên của tập dữ liệu, tức là phân lớp trong không gian 2 chiều. Chúng ta sử dụng các Kernel khác nhau bao gồm:

* SVC with linear kernel
* LinearSVC (linear kernel
* SVC with RBF kernel
* SVC with polynomial (degree 3) kernel

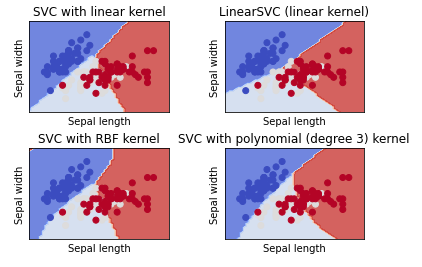
Đầu tiên chúng ta cần huấn luyện dữ liệu :



Tiếp theo là việc đẩy các mô hình thu được vào đồ thị

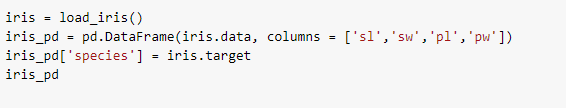


Bây giờ sau khi vẽ các đồ thị trên là chúng ta có thể so sánh hiệu quả phân lớp của các Kernel khác nhau rồi đấy. Kết quả như sau:

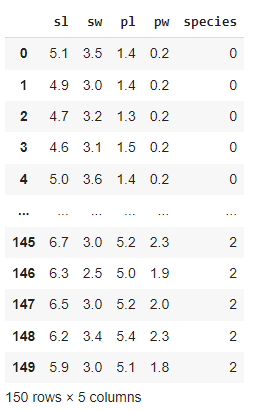


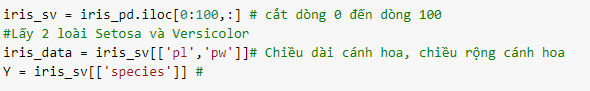
Bằng việc phân lớp chúng ta có thể chia mặt phẳng tọa độ thành các phần khác nhau. Khi có một điểm dữ liệu mới chúng ta có thể dựa vào tọa độ của chúng để **phán đoán** xem nó thuộc lớp nào.

### 1.5 Dự đoán tập data mới

Với phần này, chúng ta sẽ đi vào bước chuẩn bị model

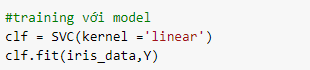
Sau khi chuẩn bị xong ta sẽ có tệp dữ liệu như sau:



Tiếp đến, chúng ta sẽ test data mới với chỉ 2 loại hoa 

Cắt data từ dòng 0 đến 100, lấy 2 loại setosa và versicolor cùng với 2 giá trị chiều dài cánh hoa(pl), chiều rộng cánh hóa(pw). Y được gán phân loại theo loài.

Tiếp theo ta sẽ training model với linear kernel:



Trước khi dự đoán ta nhắc lại công thức hàm mục tiêu:

Sau đây, phần quan trọng nhất, chúng ta sẽ tiếp hành dự đoán đưa vào thông số x1 là chiều dài cánh hoa, x2 là chiều rộng cánh hoa của công thức hàm mục tiêu:

Với đường hyperplane bằng

Vì sự phân chia không đổi của w và b, tức là khoảng cách từng điểm đến mặt phẳng không đổi, margin không đổi. Vì thế ta có:

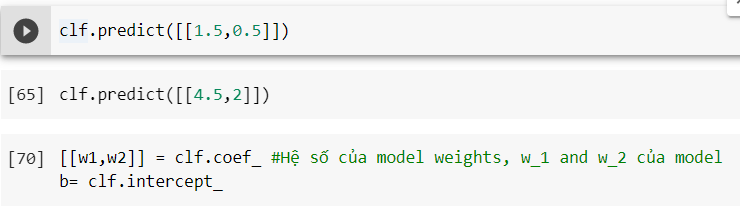
và

Ứng dụng vào data mới ta có, và là chiều dài cánh hoa, là chiều rộng cánh hoa. Trong , là , là được thế vào phương trình chia những loài hoa như sau:

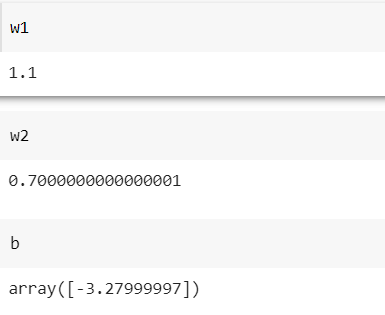
Nếu thì Y là loài ‘Setosa’

Nếu thì Y là loài ‘Versicolor’

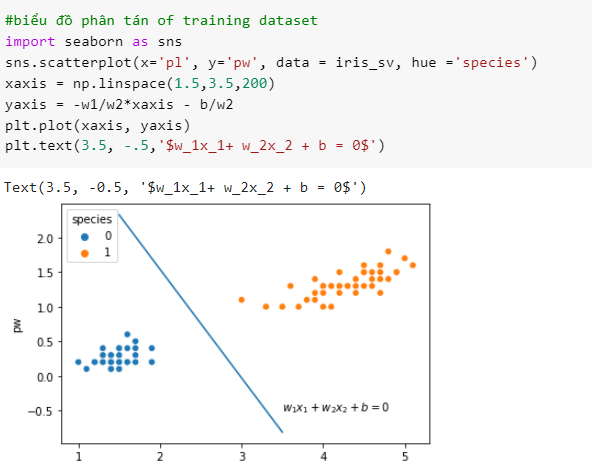
Tiếp theo, ta sẽ dự đoán với



Lấy hệ số từ model đang test. Ta có kết quả như sau:



Để trực quan hơn, chúng ta sẽ biểu diễn nó trên biểu đồ phân tán.



Với dự đoán dữ liệu điểm mới: [1.5,0.5]

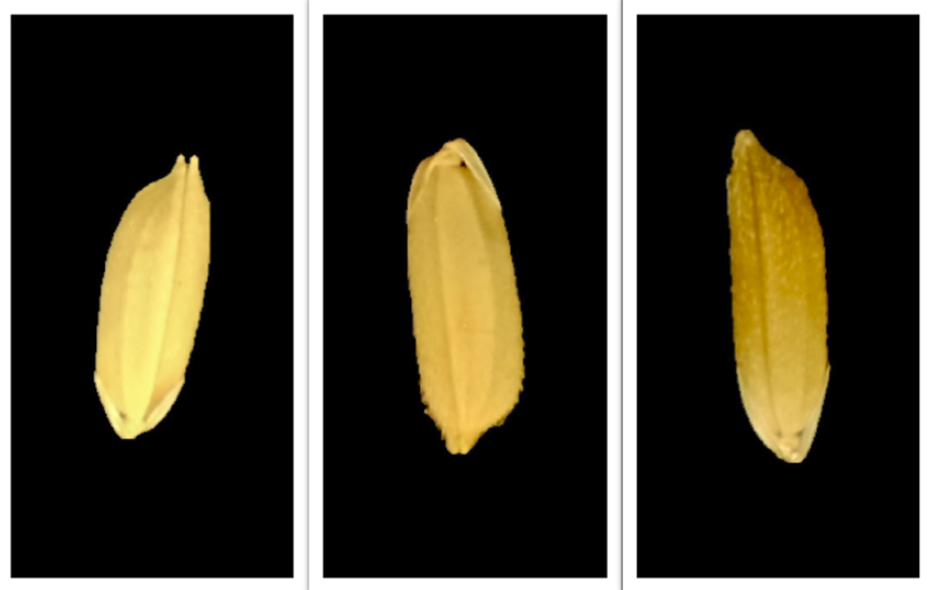
Tương tự với [4.5, 2]

* 4.5 + 0,7. 2 – 2,327999

### 1.6 Hướng phát triển

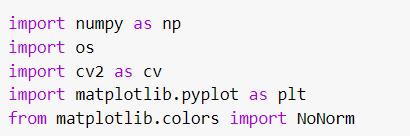
Chúng ta hoàn toàn có thể tự mình custom ra một Kernel riêng. Điều đó làm cho SVM sẽ trở nên linh hoạt hơn. Và trong phạm vi bài này thì chúng ta chỉ dùng ở các Kernel tuyến tính và phi tuyến.

## 2. Ứng dung SVM vào phân loại hạt thóc.

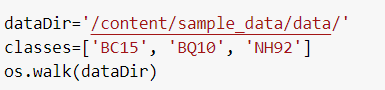
****

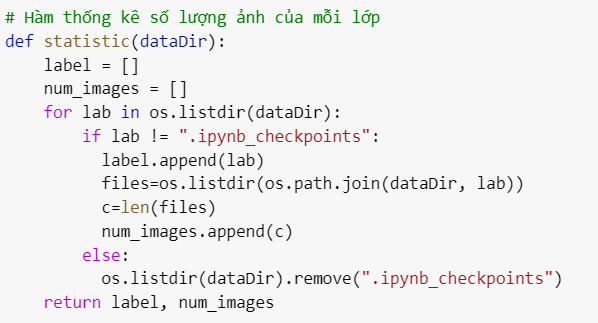
Đây là 3 hạt thóc của ba giống lúa khác nhau, lần lượt từ trái qua phải là BC15, BQ10 và NH92.

### 2.1 Chuẩn bị dữ liệu

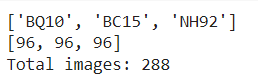
****

Để thuận tiện cho việc thống kê số lượng samples của từng class chúng ta sử dụng hàm dưới đây:



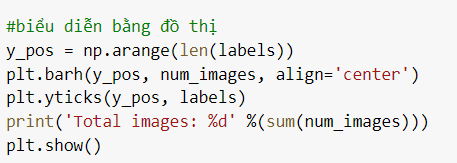


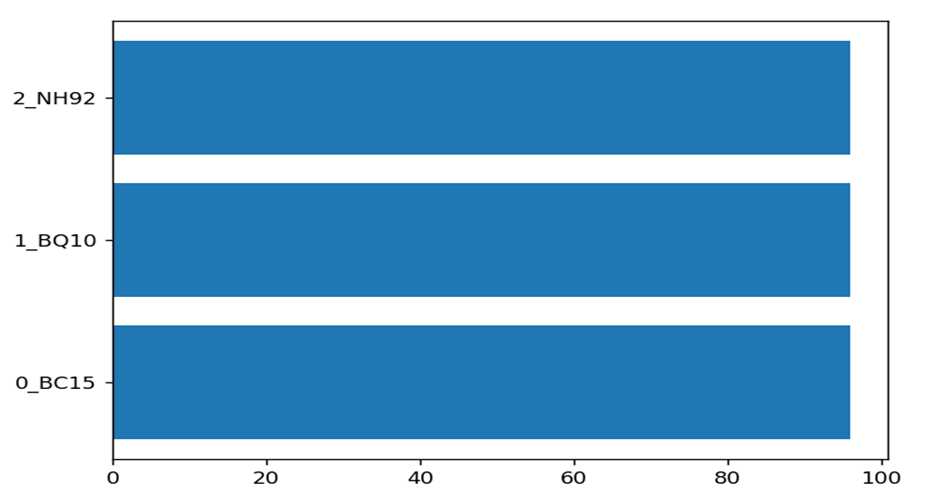
Sau khi chạy qua hàm tôi thu được kết quả thống kê như sau:



Tập dữ liệu mà tôi sử dụng gồm 3 class, số lượng samples của mỗi class là 96 ảnh.

Để có cái nhìn trực quan hơn về số lượng tôi sử dụng matplotlib để





### 2.2 Phân chia thành tập train và tập test

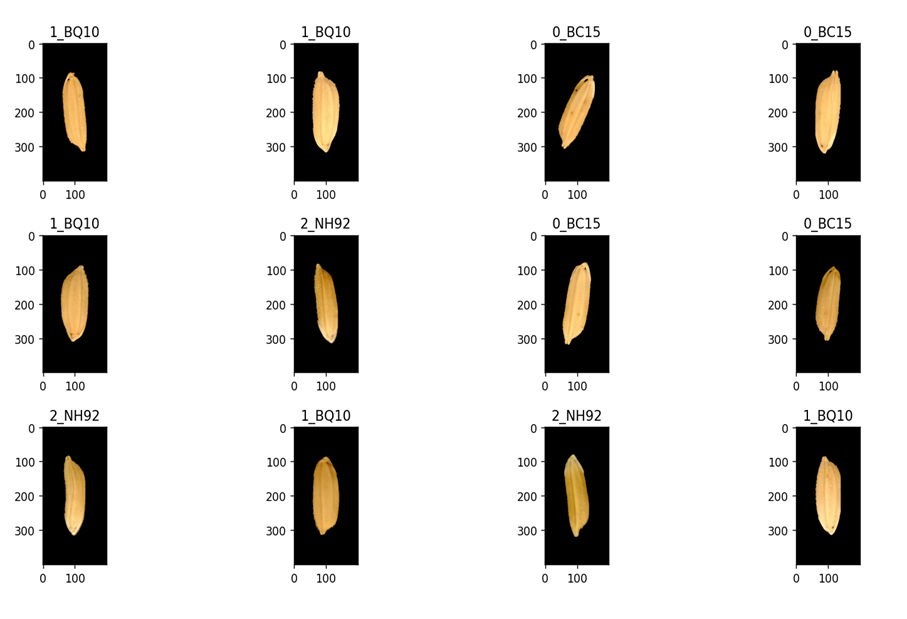
Sau quá trình thống kê dữ liệu, tôi sẽ chia dữ liệu thành 2 tập **Train** và **Test** theo tỉ lệ quen thuộc 70% cho train và 30% cho test. Để chia dữ liệu thành 2 tập thì tôi định nghĩa hàm dưới đây:



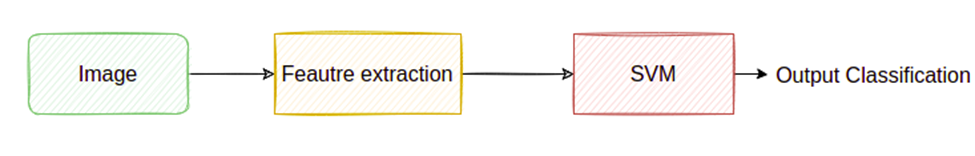
Sau khi chia train test qua hàm trên thì số lượng dữ liệu cho mỗi tập là:

* Train: 201 ảnh
* Test: 87 ảnh

Dưới đây là một số ảnh trong tập train: tương ứng với giống lúa BC15, BQ10, NH92.



### 2.3 Huấn luyện và Đánh giá

Quá trình huấn luyện cũng như phân loại khi sử dụng SVM về cơ bản sẽ có các bước như hình dưới đây:

* **Bước 1:** Tiền xử lý ảnh đầu vào.
* **Bước 2:** Trích xuất đặc trưng của ảnh.
* **Bước 3:** Đưa đặc trưng vừa trích xuất qua SVM để tiến hành phân loại.

Ở trong bước 2 có rất nhiều các phương pháp dùng để trích xuất đặc trưng của ảnh đầu vào. Nghe đến trích xuất đặc trưng thì các bạn có thể liên tưởng đến việc sử dụng một mô hình CNN để trích xuất ra những đặc trưng (deep features), và rồi dùng các đặc trưng đã trích xuất dựa trên CNN để đưa vào SVM nhằm mục đích phân loại thì kỹ thuật này gọi là Transfer Learning.

Ngoài việc sử dụng đặc trưng trích xuất từ CNN thì còn rất nhiều các kỹ thuật khác như trích xuất đặc trưng HOG(Histogram of Oriented Gradients) , đặc trưng về màu sắc của đối tượng ....

Trong bài hôm nay chúng ta sẽ sử dụng 2 đặc trưng là HOG và Color histogram

### 2.4 Sử dụng đặc trưng HOG

Sơ qua chút lý thuyết về HOG, HOG hay **Histogram of Oriented Gradients** là một thuật toán để trích xuất thuộc tính hình ảnh.Bản chất của phương pháp HOG là sử dụng thông tin về sự phân bố của các cường độ gradient (intensity gradient) hoặc của hướng biên (edge directions) để mô tả các đối tượng cục bộ trong ảnh.

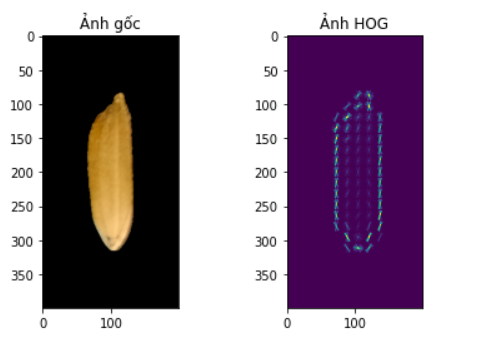
### 2.5 Trích xuất đặc trưng

Sau khi đã hình dung được HOG là gì thì sau đây sẽ định nghĩa 1 hàm dùng để trích xuất đặc trưng HOG từ ảnh đầu vào

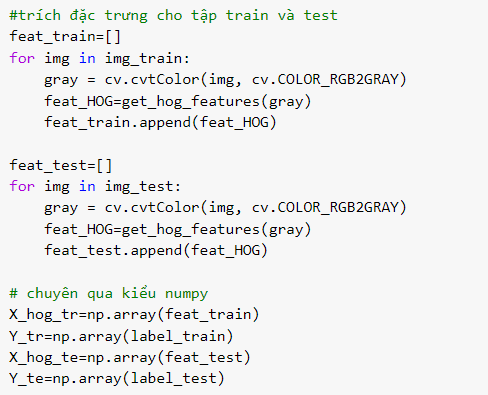


Giải thích qua một chút về hàm trích xuất đặc trưng HOG ở trên thì trong hàm này tôi sử dụng luôn hàm hog của thư viện skimage.

Chúng ta có thể để ý trong hàm này tôi sủ dụng biến bool vis để lựa chọn trả về ảnh đặc trưng HOG cùng với vector đặc trưng hoặc chỉ trả về vector đặc trưng. Nếu vis=True thì sẽ nhận được cái ảnh HOG kiểu như hình dưới đây (bên trái là ảnh gốc, bên phải là ảnh HOG)

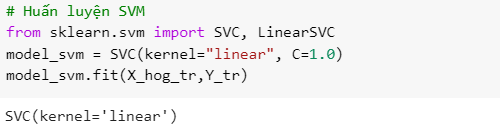


Sau khi đã định nghĩa hàm trích xuất đặc trưng, chúng ta sẽ sử dụng hàm này để trích xuất và lưu lại đặc trưng của tập train và test bằng đoạn code dưới đây :

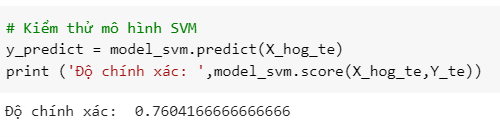


### 2.6 Huấn luyện và phân loại bằng SVM

Đến bước này thì việc của chúng ta khá là đơn giản, chỉ cần định nghĩa ra mô hình SVM và sau đó đưa dữ liệu vào để huấn luyện thôi..



Trong thử nghiệm lần này thì tôi dùng SVM với kernel là *linear*.



### 2.7 Kết Luận

Qua bài này chúng ta đã sử dụng SVM để giải quyết một bài toán khá thú vị là phân loại hạt thóc. Có thể thấy việc lựa chọn đặc trưng cho các thuật toán học máy có giám sát (Supervised learning) là khá quan trọng, nó ảnh hưởng rất lớn đến chất lượng mô hình (như trong bài này là sự khác biệt khi lựa chọn giữa đặc trưng HOG và đặc trưng Color Histogram).Và tất nhiên bài toán phân loại thóc trong bài này chỉ mang tính chất thử nghiệm, dữ liệu cho bài toán khá đẹp với những hạt thóc được đặt trong nền đen tránh được các ảnh hưởng do nền khi trích xuất đặc trưng.

**KẾT LUẬN**

Với bài toán phân lớp nhị phân mà hai lớp dữ liệu là linearly separable, có vô số các mặt phân cách phẳng giúp phân chia hai lớp đó. Khoảng cách gần nhất từ một điểm dữ liệu tới mặt phân cách ấy được gọi là margin của bộ phân lớp với ranh giới là mặt phẳng đó

Support Vector Machine là bài toán đi tìm mặt phân cách sao cho margin có được là lớn nhất, đồng nghĩa với việc các điểm dữ liệu có một khoảng cách an toàn tới mặt phân cách.

Với cái đề án đề tài này thì tụi em đã đạt được :

Tổng quan về bài toán phân lớp quan điểm và cụ thể là bài toán phân lớp phân cực để phân chia các loại hoa trọng tập dữ liệu được đào tạo Iris.

Tìm hiểu dữ liệu quan điểm và viết chương trình thử nghiệm với kỹ thuật Support Vector Machines (SVM) đã giải thích ở trên.

# TÀI LIỆU THAM KHẢO

## [1]. Ứng dụng Support Vector Machine trong bài toán phân loại hoa

## <https://viblo.asia/p/ung-dung-support-vector-machine-trong-bai-toan-phan-loai-hoa-PdbGnLXBkyA>

## [2]. Thử sử dụng Support Vector Machine để phân loại hạt thóc

## <https://viblo.asia/p/thu-su-dung-support-vector-machine-de-phan-loai-hat-thoc-gAm5yrdwKdb>

## 