ĐẠI HỌC HUẾ

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

BÁO CÁO ĐỒ ÁN

Học kỳ II, năm học 2023 - 2024

Học phần:

HỌC MÁY 2

Số phách
(Do hội đồng chấm thi ghi)

Thừa Thiên Huế, tháng 06 năm 2024

ĐẠI HỌC HUẾ

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

BÁO CÁO ĐỒ ÁN

Học kỳ II, năm học 2023 - 2024

Học phần:

HỌC MÁY 2

Giảng viên hướng dẫn: TS. Nguyễn Đăng Trị
Lớp: Khoa học dữ liệu và trí tuệ nhân tạo - K3
Sinh viên thực hiện:
Nguyễn Văn Minh Khánh – 22E1020015
Trịnh Quốc Dân – 22E1020014

(ký và ghi rõ họ tên)

Số phách
(Do hội đồng chấm thi ghi)

Thừa Thiên Huế, tháng 06 năm 2024

LỜI CẢM ƠN

“Em đã rất cố gắng và nỗ lực trong bài báo cáo đồ án này. Tuy nhiên, sẽ không thể thực hiện được nếu không có sự hỗ trợ, giúp đỡ ân cần của giảng viên bộ môn – Học máy 2 cũng như Ban giám hiệu Khoa Kỹ thuật và Công nghệ - Đại học Huế vì đã tạo điều kiện về cơ sở vật chất, như môi trường học tập thân thiện, giúp em phát huy hết khả năng học tập và rèn luyện nhân cách một cách hiệu quả.

Em muốn bày tỏ lòng biết ơn chân thành đối với thầy Nguyễn Đăng Trị và toàn thể giáo viên của Khoa Kỹ thuật và Công nghệ - Đại học Huế.

Em muốn bày tỏ lòng biết ơn đến gia đình và bạn bè vì đã luôn đồng hành, động viên và quan tâm em trên con đường học tập và trong cuộc sống.

Và lời cảm ơn đặc biệt cuối cùng em xin dành tặng cho bản thân chính mình vì đã không bỏ cuộc vào những lúc bản thân suy sụp, mệt mỏi nhất, cảm ơn bản thân đã luôn cố gắng để vượt qua những khó khăn tưởng chừng không thể bước tiếp, cảm ơn vì tất cả.”

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

MỤC LỤC

PHẦN 1. Tổng quan về mô hình Support Vector Machine - SVM

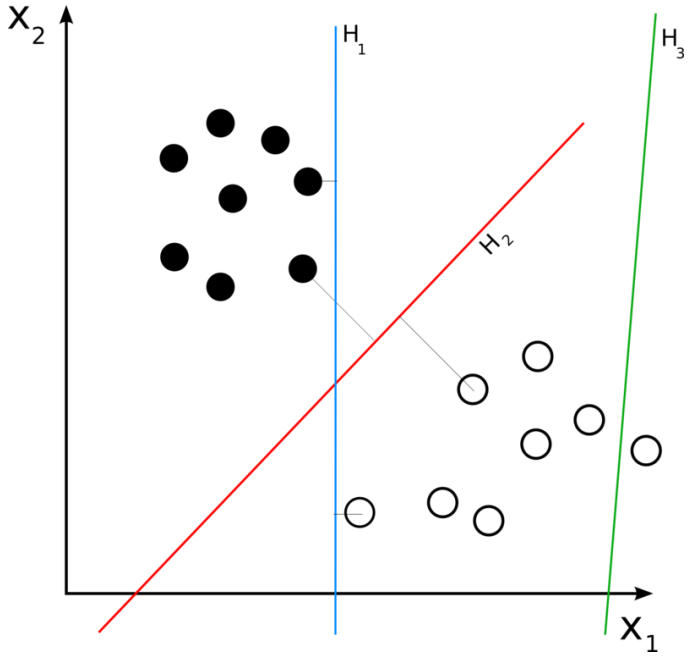
1.1 Giới thiệu

Mô hình thuật toán Support Vector Machine (SVM) là một mô hình học máy thuộc nhóm học có giám sát (Supervised Learning) và nó được sử dụng cho các bài toán phân loại và hồi quy.

SVM hoạt động bằng cách tìm kiếm một siêu phẳng (hyperplane) tối ưu trong không gian đặc trưng, nhằm phân tách các lớp dữ liệu với khoảng cách lớn nhất có thể. Điều này giúp đảm bảo khả năng tổng quát hóa tốt và giảm thiểu lỗi phân loại trên dữ liệu mới.

SVM còn có thể được xem là một mô hình tuyến tính, nghĩa là nó tìm kiếm một siêu phẳng tuyến tính để phân chia dữ liệu. Tuy nhiên, thông qua việc sử dụng các kỹ thuật kernel trick, SVM có thể mở rộng khả năng phân loại của mình đối với các dữ liệu phi tuyến tính bằng cách ánh xạ dữ liệu lên một không gian đặc trưng có chiều cao hơn.

Nền tảng của mô hình này nhằm giải quyết vấn đề cơ bản nhất là phân loại hai lớp dữ liệu. Đối với một tập dữ liệu ban đầu gồm hai lớp được tách biệt rõ ràng, câu hỏi đặt ra là nên chọn đường nào để đảm bảo khả năng tổng quát tốt nhất?

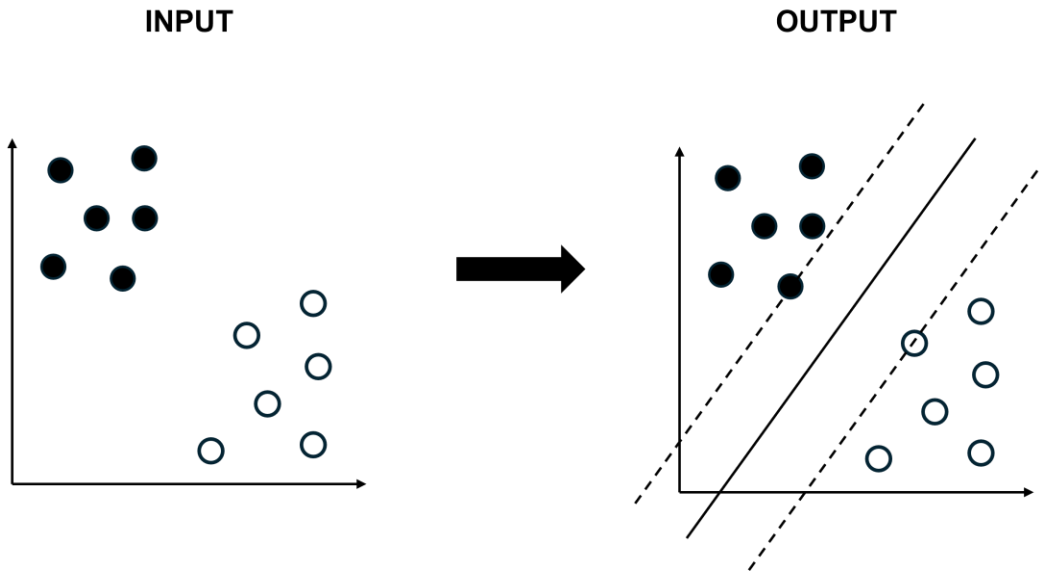


Hình 1:Phân chia 2 tập dữ liệu bằng svm - src:iostrean.co

1.2 Phát biểu bài toán cho mô hình SVM

Đầu vào: một tập dữ liệu gồm có 2 lớp, và mỗi điểm trong bộ dữ liệu được gán nhãn thuộc vào một trong hai lớp.

Đầu ra: Một đường thẳng (hoặc siêu phẳng trong không gian đa chiều) phân cách 2 lớp với khoảng cách từ đường thẳng đó tới điểm gần nhất của từng lớp, là lớn nhất có thể.



Hình 2: Đầu vào và đầu ra của tập dữ liệu - src:minhkhanh-coder

1.3 Lý thuyết toán học

1.3.1 Tập giả định của SVM

Tập giả định (hypothesis) của SVM là một tập hợp các giả định được sử dụng để dự đoán lớp của các điểm dữ liệu. Hàm giả định này trong SVM được biểu diễn dưới dạng:

$$y = h(x) = sign(w^T x + b) \quad (1)$$

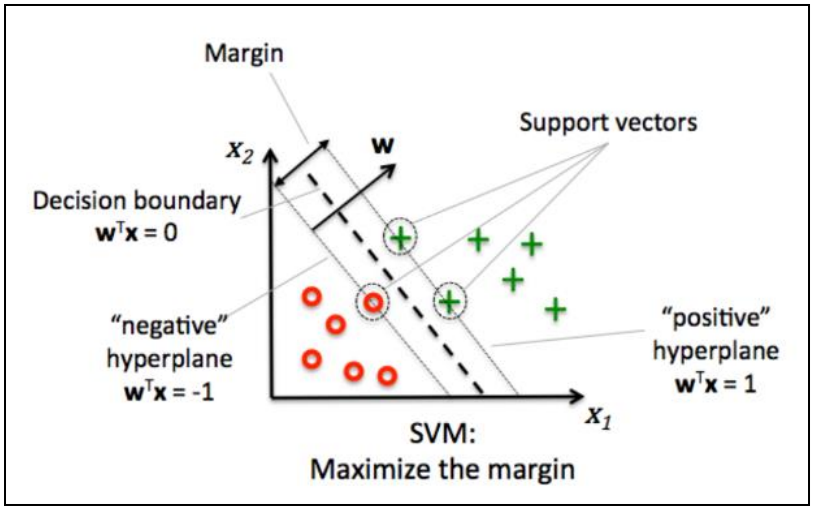
Với $x \in \mathbb{R}^d$, $w \in \mathbb{R}^d$, $b \in \mathbb{R}$. ta có:

- x là tập dữ liệu trong không gian d chiều.
- w, b là các tham số về siêu phẳng trong không gian.

Hàm $sign(x)$ là hàm dùng để xác định dấu, khi $x \geq 0$ thì $sign(x) = +1$, ngược lại $x < 0$ thì $sign(x) = -1$. Đây chính là cách phân lớp cho từng điểm dữ liệu trong SVM $y \in -1, +1$.

Hay

$$w^T x_{pos} + b = +1 \quad (2)$$
$$w^T x_{neg} + b = -1 \quad (3)$$



Hình 3: Tối đa hóa lề (margin) - src:ResearchGate

1.3.2 Bài toán tối ưu SVM

Mục tiêu chính của bài toán này là **tìm ra một mặt phẳng, sao cho mặt phẳng đó chia tách 2 phần dữ liệu ra với lề lớn nhất**. Từ đây ta có bài toán tối ưu:

Ta sẽ gọi những điểm x_n là những điểm dữ liệu gần nhất với mặt phẳng $w^T x + b$.

Và dùng công thức sau đây để tính khoảng cách giữa một điểm và một siêu phẳng trong SVM là:

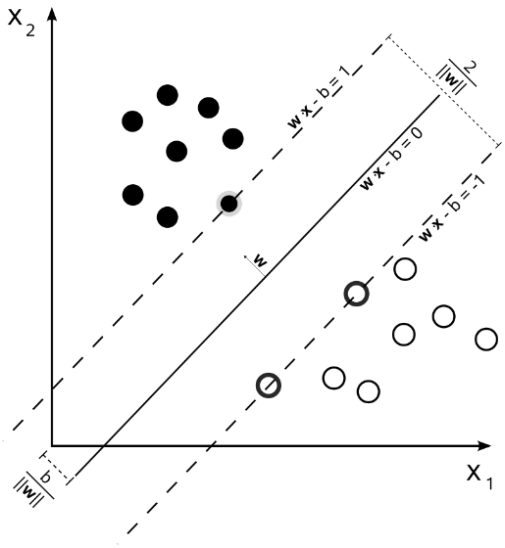
$$\frac{|w^T x_n + b|}{||w||_2} \tag{4}$$

Trong đó w là độ dài của vector được xác định bằng công thức:

$$\left|||w|\right|_2 = \sqrt{\sum_{i=1}^{d} w_i^2} \tag{5}$$

Tiếp theo để dễ dàng cho việc tính toán, ta sẽ normalize (chuẩn hóa) lại vector w sao cho:

$$|w^T x_n + b| = 1 \tag{6}$$



Hình 4: Hyperlane - src:iostream.co

Nếu trừ 2 phương trình tuyến tính của (2) và (3) với nhau, ta thu được:

$$\Rightarrow w^T(x_{pos} - x_{neg}) = 2 \tag{7}$$

Từ phương trình (4) và (7), ta được phương trình sau:

$$\frac{w^T(x_{pos} - x_{neg})}{||w||_2} = \frac{2}{||w||_2}$$

Ta rút ra được 2 nhận xét như sau:

- Những điểm x_n được phân vào lớp $+1$ sẽ có giá trị là $w^T x + b \geq 1$, ngược lại nếu được phân vào lớp -1, giá trị sẽ là $w^T x + b \leq -1$. Đó chính là cách phân lớp các điểm dữ liệu sau khi mà ta chuẩn hóa w.Với N là số lượng điểm dữ liệu, ta có $y_n(w^T x_n + b) \geq 1 \; \forall n = 1,2,3 \ldots N$.
- Độ lớn của **lề lớn nhất** cho một mặt phẳng $w^T x + b$ là:

$$\frac{2}{||w||_2}$$

Ta có thể phát biểu cho **"bài toán tối ưu cho SVM"** như sau:

$$maximize \frac{2}{||w||_2}$$
$$subject\ to\ y_n(w^T x_n + b) \geq 1 \; \forall \; n$$

Chuẩn hóa hàm mục tiêu ban đầu từ tối đa hóa sang tối thiểu hóa bằng cách nghịch đảo hàm mục tiêu:

$$\frac{1}{\frac{2}{||w||_2}} = \frac{1}{2}w^T w$$

Bài toán trên được viết lại thành:

$$minimize \frac{1}{2}w^T w \tag{8}$$

$$subject\ to\ y_n(w^T x_n + b) \geq 1\ \forall\ n = 1,2,3,\dots,N$$

1.3.3 Phương pháp tối ưu cho SVM

Để tối ưu cho SVM, ta dùng phương pháp nhân tử **Lagrange** vì nó giúp chuyển bài toán tối ưu ban đầu thành dạng tối ưu đối ngẫu, dễ giải quyết hơn. Bằng cách này, chúng ta có thể áp dụng các phương pháp tối ưu hiệu quả để tìm ra các hệ số tối ưu cho siêu phẳng phân chia, từ đó làm tăng hiệu suất của mô hình SVM và giải quyết bài toán phân loại dữ liệu một cách chính xác và hiệu quả.

Từ **mục 1.3.2**, ta có được hàm mục tiêu:

$$\frac{1}{2}w^T w$$

Điều kiện ràng buộc:

$$y_n(w^T x_n + b) - 1 \geq 0\ \forall\ n = 1,2,3,\dots,N$$

Để dựng hàm Lagrange, ra thêm biến α_n tương ứng với điều kiện $y_n(w^T x_n + b) - 1 \geq 0$:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}w^T w - \sum_{n=1}^{N} \alpha_n \left(y_n(w^T x_n + b) - 1\right)$$

$$(8) \Leftrightarrow Minimize\ with\ w, b;\ Maximize\ with\ \alpha: \alpha_n \geq \mathcal{L}(w, b, \alpha) \qquad (*)$$

Vậy là ta đã chuyển bài toán tối ưu từ dạng nguyên bản bài toán số (8) thành dạng đối ngẫu của bài toán $(*)$.

Và để giải bài toán (*) này, ta sẽ giải lần lượt từ Minimize cho w, b khi coi các α_n là hằng số. Rồi sau đó là giải Maximize với $\alpha_n \geq 0$ khi coi w, b là hằng số.

Minimize with w, b

$$Minimize\ with\ w, b = \frac{1}{2}w^T w - \sum_{n=1}^{N} \alpha_n \left(y_n(w^T x_n + b) - 1\right) \boldsymbol{with}\ w, b$$

Tính đạo hàm của w, b và đặt đạo hàm bằng 0, ta có:

$$\frac{\partial \mathcal{L}}{\partial w} = w - \sum_{n=1}^{N} \alpha_n y_n x_n = 0 \Leftrightarrow w = \sum_{n=1}^{N} \alpha_n y_n x_n \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{n=1}^{N} \alpha_n y_n = 0 \Leftrightarrow \sum_{n=1}^{N} \alpha_n y_n = 0 \tag{10}$$

Thay thế (9) và (10) vào $\mathcal{L}(w, b, \alpha)$ và rút gọn, ta có:

$$\mathcal{L}(\alpha) = \frac{1}{2}\left(\sum_{n=1}^{N} \alpha_n y_n x_n\right)^T \left(\sum_{n=1}^{N} \alpha_n y_n x_n\right) - \sum_{n=1}^{N} \alpha_n \left(y_n \left(\left(\sum_{n=1}^{N} \alpha_n y_n x_n\right)^T x_n + b\right) - 1\right)$$

$$= \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m - \left(\sum_{n=1}^{N} \alpha_n y_n x_n\right)\left(\sum_{n=1}^{N} \alpha_n y_n x_n\right)^T - b\sum_{n=1}^{N} \alpha_n y_n + \sum_{n=1}^{N} \alpha_n$$

$$= \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m - \sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m \; - b.0 + \sum_{n=1}^{N} \alpha_n$$

$$= -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m + \sum_{n=1}^{N} \alpha_n$$

$$= \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m$$

Vậy là bài toán tối ưu của ta chỉ còn lại:

$$Maximize\ \mathcal{L}(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m \tag{**}$$

Với các $\alpha_n \geq 0\ \forall n = 1,2,\ldots,N$ **và điều kiện ràng buộc (10)** $\sum_{n=1}^{N} \alpha_n y_n = 0$.

Như vậy ta đã hoàn thành chuyển đổi bài toán từ dạng nguyên bản (8) về dạng đối ngẫu của bài toán $(*)$. Cuối cùng ta sẽ chỉ việc tìm ra các nghiệm α_n để hoàn tất quá trình tối ưu cho SVM.

Để tìm ra nghiệm α_n ta sẽ sử dụng quy hoạch toàn phương (Quadratic Programming). Ta tóm tắt lại các bước của việc tối ưu cho SVM như sau:

1. Giải các nghiệm α_n thỏa:

$$\alpha_n^* = argmin_\alpha \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\,\alpha_m\,y_n y_m x_n^T x_m - \sum_{n=1}^{N}\alpha_n$$

   Với các $\alpha_n \geq 0$ và $\sum_{n=1}^{N}\alpha_n y_n = 0$.

2. Có được các α_n, ta tính w theo (9):

$$w = \sum_{n=1}^{N}\alpha_n\,y_n x_n$$

   Thực tế khi giải ra α_n bằng quy hoạch toàn phương chỉ có vài nghiệm $\alpha_n > 0$, ta viết lại thành:

$$w = \sum\nolimits_{\alpha_n>0}\alpha_n y_n x_n$$

   Và với những $\alpha_n > 0$, ta lại có $\alpha_n(y_n(w^T x_n + b) - 1) = 0$ (Từ một trong những **điều kiện Karush-Kuhn-Tucker - KKT** để chuyển từ dạng nguyên bản sang đối ngẫu), suy ra:

$$y_n(w^T x_n + b) = 1$$

   Từ đây, ta có thể nói rằng những điểm x_n có các $\alpha_n > 0$ là những điểm nằm gần nhất với mặt phẳng $w^T x_n + b$ và có khoảng cách tới mặt phẳng là:

$$\frac{1}{\|w\|_2}$$

   Những điểm x_n này đóng góp vào việc tính toán ra vector pháp tuyến w của mặt phẳng, vậy nên ta gọi đó là những **support vector**.

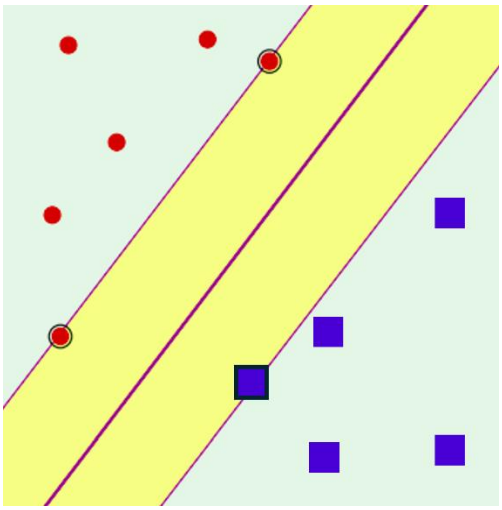
3. Tìm hệ số b.

   Để tìm hệ số b ta có thể dùng bất kỳ support vector nào vì $y_n(w^T x_n + b) = 1$.

PHẦN 2. Áp dụng mô hình SVM trong thực tế

2.1 Dữ liệu không khả tách tuyến tính

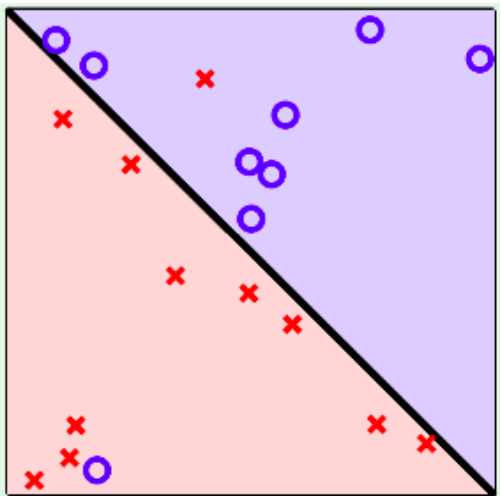
Trước hết, chúng ta có định nghĩa về linear separability (khả năng phân tách tuyến tính) của dữ liệu. Một dữ liệu được coi là khả tách tuyến tính khi chúng ta có thể dùng một đường thẳng để phân tách 2 lớp của nó rõ rệt, như được minh họa trong hình dưới đây:



Hình 5: Dữ liệu không khả tách tuyến tính - src: ashwanibhardwajcodevita16

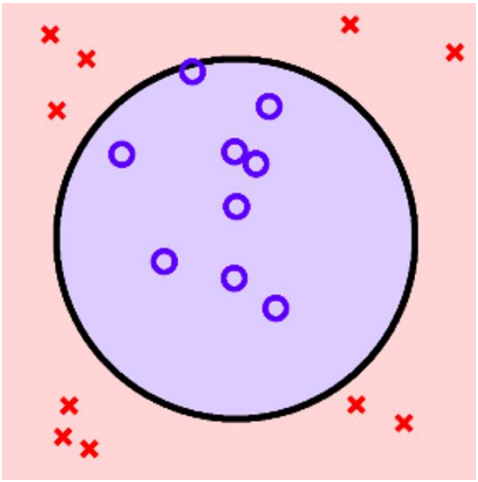
Mô hình SVM (Support Vector Machine) ban đầu được xây dựng dựa trên giả định rằng dữ liệu có thể được phân chia bởi một đường thẳng (trong trường hợp dữ liệu 2 chiều) hoặc một siêu phẳng (trong không gian nhiều chiều). Tuy nhiên, trong thực tế, không phải lúc nào dữ liệu cũng thỏa mãn giả định này. Có những trường hợp mà dữ liệu không thể được phân chia bằng một đường thẳng như vậy. Ví dụ như:

- **Trường hợp 1:** Dữ liệu không thể tách ra bằng 1 đường thẳng.



Hình 6: Dữ liệu không thể tách ra bằng 1 đường thẳng - src: jip.dev

- **Trường hợp 2:** Dữ liệu là phi tuyến tính.



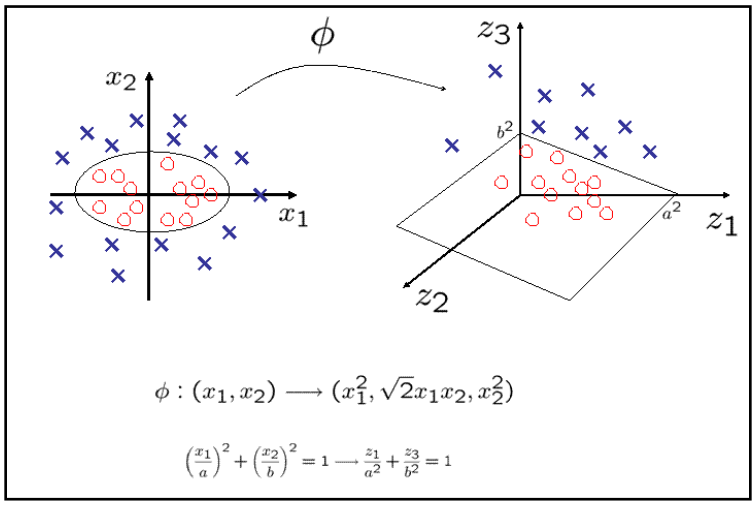
Hình 7: Dữ liệu là phi tuyến tính - src: jip.dev

Để xử lý các trường hợp dữ liệu không thể tách tuyến tính này, mô hình SVM cung cấp các công cụ để giải quyết bằng 2 phương pháp chính thường dùng là: **Biến đổi phi tuyến với Kernel, Soft-Margin (Lề mềm)**.

2.2 Biến đổi phi tuyến với các Kernel

Phương pháp này cho phép chúng ta ánh xạ dữ liệu từ không gian ban đầu sang một không gian khác có số chiều cao hơn, nơi mà việc tách các lớp trở nên dễ dàng hơn. Các kernel phổ biến như Polynomial Kernel, Gaussian Kernel, và Sigmoid Kernel thường được sử dụng để thực hiện biến đổi này.

Đây là phương pháp được áp dụng cho các dữ liệu phi tuyến, ta xét ví dụ sau:



Hình 8: Biến đổi phi tuyến từ không gian X sang Z - src: omega0.xyz

Với dữ liệu trong không gian X ở hình bên trái, chúng ta không thể sử dụng bất kỳ đường thẳng nào của SVM để phân tách hai phần dữ liệu. Tuy nhiên, ở hình bên phải, dữ liệu đã được biến đổi phi tuyến từ không gian X bằng cách điều chỉnh lại gốc tọa độ tại tâm và áp dụng ánh xạ từ $(x_1, x_2) \in \mathbf{X} \rightarrow (x_1^2, x_2^2) \in \mathbf{Z}$.

Trong SVM, ta không cần áp dụng các phép biến đổi phi tuyến lên từng điểm dữ liệu. Dựa vào bài toán tối ưu, ta có:

$$argmin_\alpha \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m - \sum_{n=1}^{N} \alpha_n$$

Ở đây, ta chỉ cần chú ý đến tích vô hướng của 2 điểm dữ liệu bất kì $x_n^T x_m$ để tạo ra ma trận Quadratic cho bài toán giải quy hoạch toàn phương để tìm nghiệm α_n.

Vậy nên, khi ta cần áp dụng một phép biến đổi $z = \Phi(x)$ vào cho không gian dữ liệu **X** để tạo không gian dữ liệu **Z**, ta chỉ cần thực hiện định nghĩa một hàm $K(x, x') = \Phi(x)^T \Phi(x') = z^T z'$.

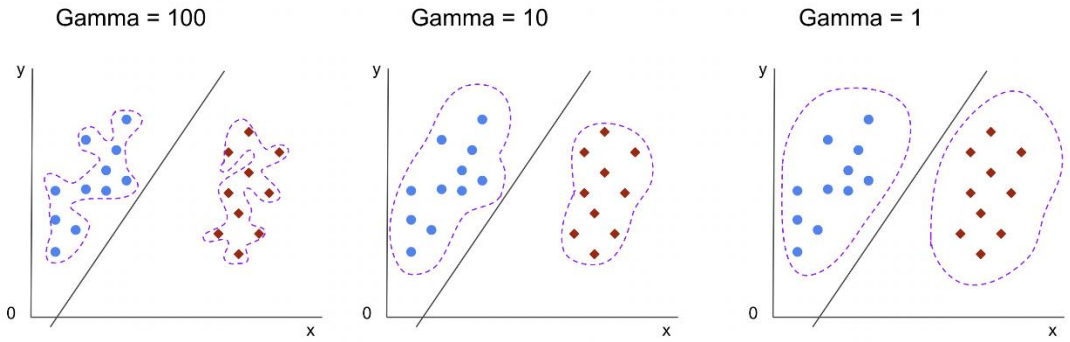
Bài toán tối ưu thành:

$$argmin_\alpha \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m K(x_n x_m) - \sum_{n=1}^{N} \alpha_n$$

Và K được gọi là hàm Kernel của SVM. Một số hàm Kernel thường được sử dụng như sau:

1. Kernel tuyến tính ban đầu của SVM: $K(x, x') = x^T x'$
2. Kernel đa thức (polynomial): $K(x, x') = (a(x^T x') + b)^Q$ với Q là bậc của đa thức, a, b là các hệ số.
3. Kernel RBF - Radial Basis Function: $K(x, x') = exp(\gamma |x - x'|^2)$ đây là một hàm kernel có vô hạn chiều.

Trong SVM, đường biên quyết định được xác định bởi các vectơ hỗ trợ, và giá trị của γ có ảnh hưởng đến độ cong và phân phối của các vectơ hỗ trợ. Giá trị γ càng lớn, đường biên càng phức tạp và uốn cong nhiều hơn để vừa khớp với dữ liệu. Trong khi đó, giá trị γ nhỏ sẽ dẫn đến một đường biên mềm hơn và ít uốn cong hơn.

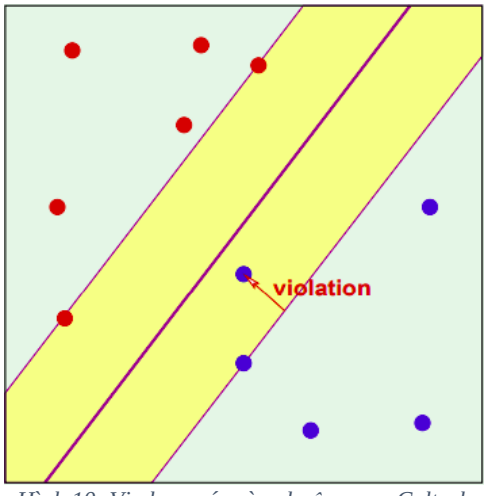


Hình 9: Low gamma – High gamma – src: stackabuse.com

2.3 Mô hình SVM với Lề mềm (Soft-margin)

Với mô hình SVM gốc, nó là mô hình SVM lề cứng (hard-margin), tập giả định $h(x)$ và độ lớn lề được tính từ các support vector x_n thỏa mãn các điều kiện ràng buộc. Có thể viết điều này một cách tổng quát cho mọi điểm trong tập dữ liệu là $y_n(w^T x_n + b) \geq 1, n = 1, 2, 3 \ldots N$.

Thông thường, mô hình lề cứng gặp khó khăn khi yêu cầu điều kiện quá nghiêm ngặt, đặc biệt trong các tập dữ liệu có nhiễu hoặc không thể phân tách hoàn hảo. Để giải quyết vấn đề này, mô hình SVM Lề mềm (Soft-margin SVM) được phát triển. Mô hình này cung cấp một cơ chế linh hoạt hơn để tìm đường thẳng phân tách dữ liệu và lề, cho phép một số điểm dữ liệu vi phạm điều kiện margin. Ta gọi ξ_n là một tham số vi phạm tương ứng với mỗi điểm x_n, SVM Soft-margin chỉ yêu cầu điều kiện $y_n(w^T x_n + b) \geq 1 - \xi_n$ với $\xi_n \geq 0$.



Hình 10: Vi phạm các ràng buộc - src: Caltech

Việc sử dụng lề mềm trong mô hình SVM là để cho phép mô hình chấp nhận sự không hoàn hảo trong việc phân loại dữ liệu. Trong thực tế, không phải lúc nào các điểm dữ liệu cũng rõ ràng và có thể phân loại một cách hoàn hảo bằng một đường ranh giới cứng.

Ta sẽ mô hình hóa lại việc các điểm dữ liệu "được phép vi phạm" ràng buộc $y_n(w^T x_n + b) \geq 1, n = 1,2,3 \ldots N$ như sau:

Với $\xi_n \geq 0$ là các tham số vi phạm tương ứng với mỗi điểm dữ liệu x_n, ta cần thỏa mãn: $y_n(w^T x_n + b) \geq 1 - \xi_n$.

Tổng giá trị vi phạm là $\sum_{n=1}^{N} \xi_n$.

Bài toán tối ưu được viết lại từ (8) thành:

$$minimize \frac{1}{2} w^T w + C \sum_{n=1}^{N} \xi_n \tag{11}$$

$$subject\ to\ y_n(w^T x_n + b) \geq 1\ - \xi_n, \xi_n \geq 0\ \forall\ n = 1,2,3, \ldots, N$$

Chú thích:

+ Giải thích thêm vì sao lại thêm đại lượng $C \sum_{n=1}^{N} \xi_n$ ở đây, chúng ta đang cực tiểu hóa $\frac{1}{2} w^T w$ tương đương với việc cực đại hóa độ lớn của lề.
+ Đại lượng $\sum_{n=1}^{N} \xi_n$ thể hiện cho mức độ sự vi phạm độ lớn của lề.
+ C là một tham số thể hiện tương quan giữa 2 mục tiêu tối ưu, là những đường cong phân chia các tập dữ liệu.

Chúng ta sẽ cảm giác thấy $C \sum_{n=1}^{N} \xi_n$ gần giống như Regularization (chính quy hóa) cho mô hình SVM, và thực tế chúng ta sẽ chứng minh sau đây:

Ta có hàm nhân tử Lagrange tương ứng với giải bài toán (11) là:

$$\mathcal{L}(w, b, \xi, \alpha, \beta) = \frac{1}{2} w^T w + C \sum_{n=1}^{N} \xi_n - \sum_{n=1}^{N} \alpha_n \left(y_n(w^T x_n + b) - 1 + \xi_n\right) - \sum_{n=1}^{N} \beta_n \xi_n$$

Hàm $\mathcal{L}$ cần minimize với w, b, ξ, và maximize với mọi $\alpha_n \geq 0$ và $\beta_n \geq 0$.

Tiếp tục giải ra bài toán này như trên, ta sẽ đưa ra các kết quả gần tương tự với SVM Hard-margin:

$$\frac{\partial \mathcal{L}}{\partial w} = w - \sum_{n=1}^{N} \alpha_n y_n x_n = 0$$

$$\frac{\partial \mathcal{L}}{\partial b} = - \sum_{n=1}^{N} \alpha_n y_n = 0$$

$$\frac{\partial \mathcal{L}}{\partial \xi_n} = C - \alpha_n - \beta_n = 0 \quad (12)$$

Tiếp tục thay thế các kết quả trên vào bài toán tối ưu cho các α, ta có:

$$Maximize\ \mathcal{L}(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n\, \alpha_m\, y_n y_m x_n^T x_m$$

Với $0 \leq \alpha_n \leq C$ (được suy ra từ (12)) với mọi $n = 1,2,3 \ldots N$ và $\sum_{n=1}^{N} \alpha_n y_n = 0$.

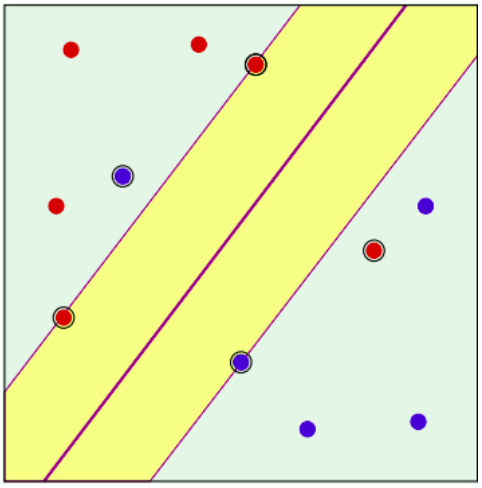
Tiếp tục giải bài toán trên bằng QP và việc ta có 1 giới hạn cho nhưng α_n được suy ra từ (12), với mỗi x_n là những support vector được tìm ra sẽ có các giá trị α_n tương ứng được giới hạn $0 \leq \alpha_n \leq C$. Nhưng ta vẫn sẽ phân chia các support vector vào 2 loại:

- Với những support vector x_n có $\alpha_n < C$, ta gọi chúng là những margin support vector (những support vector nằm trên lề) vì $\alpha_n < C \Rightarrow \beta_n > 0 \Rightarrow \xi_n = 0$ (để thỏa mãn cho bài toán maximize ở trên).

  $\rightarrow$ Đây chính là những vector thỏa mãn ràng buộc $y_n(w^T x_n + b) = 1$.

- Với những support vector x_n có $\alpha_n = C$, ta gọi chúng là những non-margin support vector (support vector không nằm trên lề): $\alpha_n = C \Rightarrow \xi_n > 0$.

  $\rightarrow$ Các support vector này có tính chất $y_n(w^T x_n + b) < 1$.



Hình 11: 2 loại super vector - src: ashwanibhardwajcodevita16

Nhận xét vai trò của tham số C trong Soft-margin:

– Tham số C là một tham số quan trọng trong SVM, quyết định mức độ ưu tiên giữa việc tối ưu hóa độ lớn của lề và việc giảm thiểu số lượng điểm dữ liệu bị phân loại sai.

- Khi C lớn, mỗi lỗi ξ_n sẽ được phạt nặng hơn, dẫn đến việc mô hình SVM sẽ cố gắng phân loại các điểm dữ liệu đúng hơn và chấp nhận lề nhỏ hơn. Ngược lại, khi C nhỏ, mô hình SVM có thể chấp nhận một lề lớn hơn nhưng phạt ít hơn cho các điểm dữ liệu bị phân loại sai. Nên chúng ta thường muốn nới lỏng bằng Soft-margin để tăng độ lớn của lề và chấp nhận độ lỗi lúc huấn luyện có thể cao hơn.
- C đóng vai trò gần tương tư như Regularization (chính quy hóa) cho SVM.
- Khi điều chỉnh tham số C, ta cần chú ý: **với giá trị C càng tăng lên**, nghĩa là các non-margin support (các điểm dữ liệu không nằm trên đường biên) vector x_n càng khó khăn đạt được giá trị $\alpha_n = C$, mô hình sẽ chuyển từ Soft-margin thành Hard-margin, độ lớn của lề khi tìm được sẽ **nhỏ lại**. Với các **giá trị C càng nhỏ dần**, nghĩa là ta sẽ ch5ấp nhận nhiều non-margin support vector hơn, **độ lớn của lề sẽ tăng lên**.

PHẦN 3. Đề xuất ứng dụng và mô phỏng thuật toán với thư viện scikit-learn

3.1 Đề xuất ứng dụng

SVM thường được áp dụng nhiều trong các tác vụ phân loại và dự báo, cũng như được nhiều công ty ứng dụng và triển khai trên môi trường production. Chúng ta có thể liệt kê một số ứng dụng của thuật toán SVM đó là:

1. **Phân loại hình ảnh:** phân loại các đối tượng trong hình ảnh, như nhận dạng khuôn mặt, phân loại đối tượng trong ảnh y khoa, hay phân loại sản phẩm trong ảnh bán hàng.
2. **Phát hiện gian lận:** sử dụng trong lĩnh vực tài chính để phát hiện gian lận trong giao dịch tín dụng hoặc giao dịch tài chính.
3. **Nhận diện văn bản:** sử dụng để nhận diện văn bản, ví dụ như phân loại email vào hộp thư rác và hộp thư đến.
4. **Dự đoán thị trường tài chính:** dự đoán xu hướng thị trường tài chính, như dự đoán giá cổ phiếu hoặc giá tiền điện tử.
5. **Phân loại văn bản:** phân loại văn bản, như phân loại tin tức vào các chủ đề khác nhau.
6. **Nhận diện tiếng nói:** nhận dạng lệnh trong hệ thống điều khiển giọng nói hoặc trong ứng dụng nhận dạng người nói.
7. **Dự đoán y tế:** được áp dụng để dự đoán các tình trạng y tế, như dự đoán nguy cơ mắc các bệnh lý dựa trên dữ liệu bệnh án.

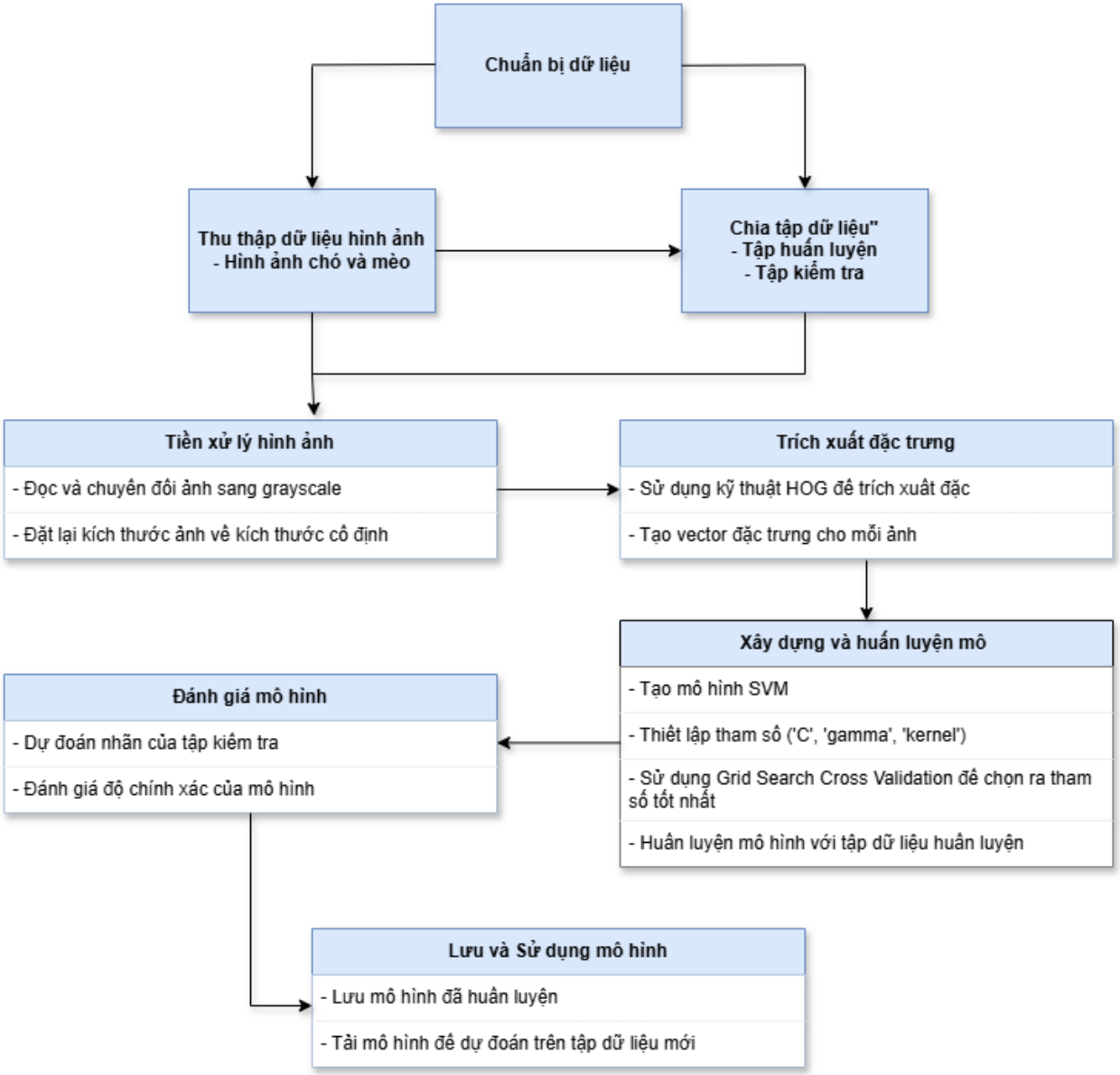
3.2 Mô phỏng thuật toán với thư viện scikit-learn

3.2.1 Phân loại hình ảnh

Link source code: Click Here

Trong phần này, chúng ta sẽ trình bày cách áp dụng SVM vào việc phân loại hình ảnh chó và mèo, từ khâu tiền xử lý dữ liệu đến việc huấn luyện, đánh giá mô hình và dự đoán hình ảnh dựa trên model đã được luyện tập.

Lược đồ mô tả quy trình phân loại hình ảnh bằng thuật toán SVM:



Hình 12: Quy trình phân loại hình ảnh bằng thuật toán SVM

Điểm đánh giá mô hình phân loại hình ảnh chó và mèo:

```
Best parameters found: {'C': 1, 'gamma': 0.1, 'kernel': 'rbf'}
Accuracy: 0.7071428571428572
Model saved to svm_dog_cat_model.pkl
```

Hình 13: Điểm đánh giá mô hình phân loại hình ảnh

Kết quả sau khi dự đoán:

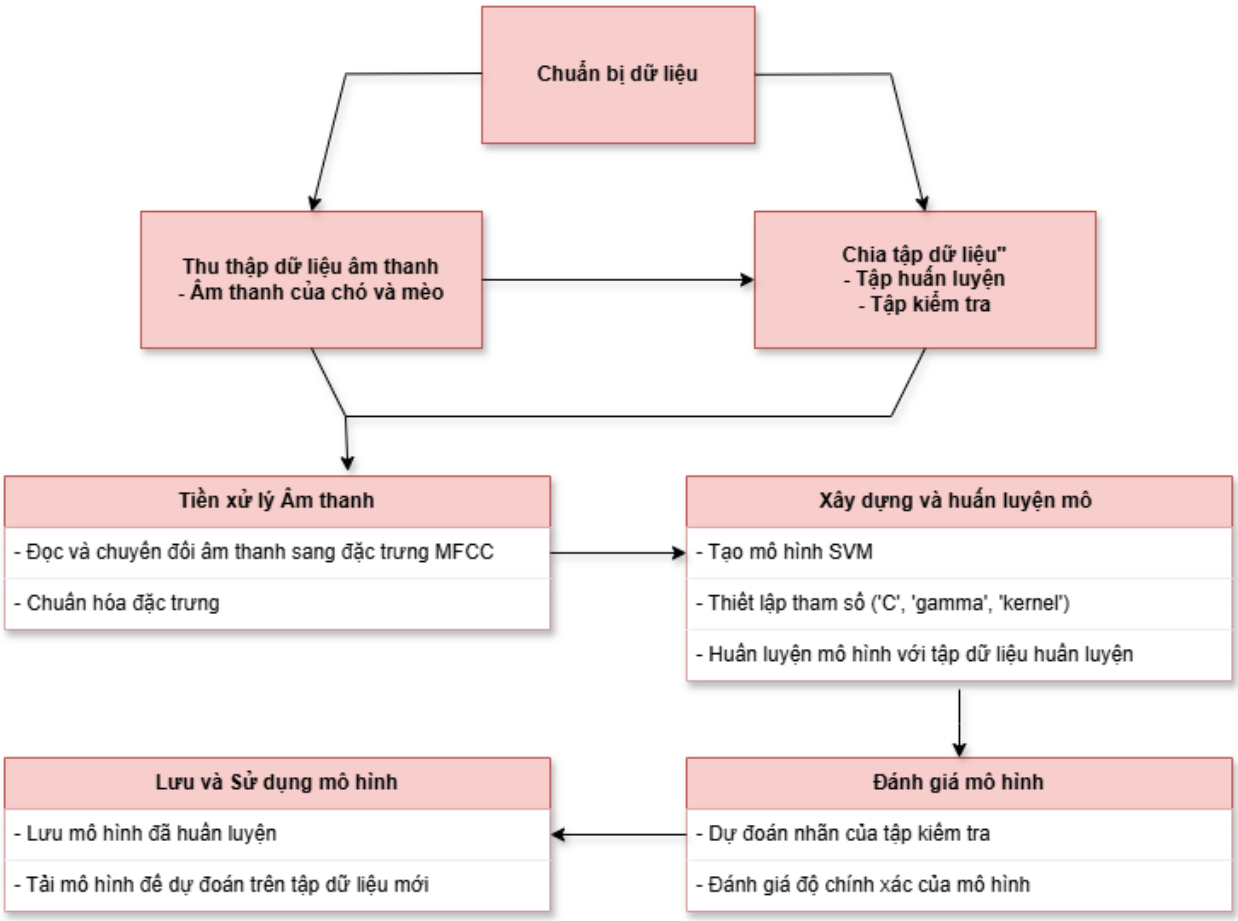
Hình 14: Kết quả phân loại chó và mèo

3.2.2 Phân loại âm thanh

Link source code: Click Here

Tiếp theo, chúng ta sẽ khám phá quá trình phân loại âm thanh giữa chó và mèo bằng cách sử dụng các kỹ thuật xử lý tín hiệu âm thanh và học máy bằng thuật toán SVM. Chúng ta sẽ tìm hiểu về các bước từ việc thu thập và tiền xử lý dữ liệu âm thanh, đến trích xuất đặc trưng và xây dựng mô hình phân loại, cùng với ứng dụng thực tế của việc phân loại âm thanh trong cuộc sống hàng ngày.

Lược đồ mô tả quy trình phân loại âm thanh bằng thuật toán SVM:



Hình 15: Quy trình phân loại âm thanh bằng thuật toán SVM

Điểm đánh giá mô hình phân loại âm thanh của chó và mèo:

Accuracy: 0.9104477611940298

	precision	recall	f1-score	support
cat	0.88	0.97	0.93	39
dog	0.96	0.82	0.88	28
accuracy			0.91	67
macro avg	0.92	0.90	0.91	67
weighted avg	0.91	0.91	0.91	67

Hình 16: Điểm đánh giá mô hình phân loại âm thanh

Kết quả sau khi dự đoán:

```
File: dataset/test\cat\cat_61.wav => dự đoán Label: cat
File: dataset/test\cat\cat_66.wav => dự đoán Label: cat
File: dataset/test\cat\cat_67.wav => dự đoán Label: cat
File: dataset/test\cat\cat_75.wav => dự đoán Label: cat
File: dataset/test\cat\cat_76.wav => dự đoán Label: cat
File: dataset/test\cat\cat_79.wav => dự đoán Label: cat
File: dataset/test\cat\cat_82.wav => dự đoán Label: cat
File: dataset/test\cat\cat_85.wav => dự đoán Label: cat
File: dataset/test\cat\cat_86.wav => dự đoán Label: cat
File: dataset/test\cat\cat_88.wav => dự đoán Label: cat
File: dataset/test\cat\cat_90.wav => dự đoán Label: cat
File: dataset/test\dog\dog_barking_112.wav => dự đoán Label: dog
File: dataset/test\dog\dog_barking_12.wav => dự đoán Label: dog
File: dataset/test\dog\dog_barking_15.wav => dự đoán Label: dog
File: dataset/test\dog\dog_barking_19.wav => dự đoán Label: dog
File: dataset/test\dog\dog_barking_24.wav => dự đoán Label: cat
File: dataset/test\dog\dog_barking_3.wav => dự đoán Label: cat
File: dataset/test\dog\dog_barking_34.wav => dự đoán Label: dog
File: dataset/test\dog\dog_barking_43.wav => dự đoán Label: dog
File: dataset/test\dog\dog_barking_44.wav => dự đoán Label: dog
File: dataset/test\dog\dog_barking_45.wav => dự đoán Label: dog
```

Hình 17: Kết quả phân loại âm thanh

TÀI LIỆU THAM KHẢO

[1] Learning From Data – Caltech

[2] SVM Introduction – Bùi Văn Hợp

[3] Python Machine Learning Packt Publishing 2015 – Sebastian Raschka

[4] ChatGPT

KẾT QUẢ KIỂM TRA ĐẠO VĂN