Tiểu luận Ai - jll gj gj

Ứng dụng AI trong Kinh doanh quản lý (Đại học Kinh tế Quốc dân)

MỤC LỤC

LỜI NÓI ĐẦU

Trong thế kỷ 21 đầy thách thức và tiến bộ của công nghệ thông tin, việc phân loại và dự đoán dữ liệu đã trở thành một phần quan trọng của nghiên cứu và ứng dụng thực tế. Trong lĩnh vực Machine Learning, có một phương pháp mạnh mẽ và linh hoạt đã được chứng minh là hiệu quả trong nhiều ứng dụng khác nhau, đó là Support Vector Machine (SVM). SVM là một mô hình học máy được sử dụng rộng rãi cho các tác vụ như phân loại, hồi quy và phát hiện bất thường. Phương pháp này được phát triển ban đầu bởi Vladimir Vapnik và đồng nghiệp vào những năm 1990 và từ đó đã trở thành một trong những công cụ quan trọng nhất trong học máy. Bằng cách nắm vững kiến thức về SVM, chúng ta có thể áp dụng nó vào nhiều tình huống thực tế, từ phân loại ảnh y tế đến dự đoán thị trường tài chính. Điều này không chỉ mở ra cánh cửa cho các ứng dụng mới mẻ mà còn giúp chúng ta hiểu sâu hơn về cách máy tính học và ra quyết định. Trong tiểu luận này sẽ tìm hiểu về phương pháp support vector machine và ứng dụng thuật toán support vector machine qua ngôn ngữ lập trình Python

CHƯƠNG 1: GIỚI THIỆU PHÁT BIỂU ĐỀ TÀI

1. Phát biểu đề tài

- **Đề tài:** Chương trình phần mềm cài đặt thuật toán phân lớp máy vector hỗ trợ SVM (Support Vector Machine).
- **Nội Dung Đề Tài:** Tìm hiểu về thuật toán phân lớp máy vector (SVM) và ứng dụng SVM sử dụng ngôn ngữ lập trình Python
- **Yêu cầu:** Hiểu được thuật toán phân lớp máy vector, viết Code ứng dụng cho thuật toán.
- **Lý do chọn đề tài:** Vấn đề phân lớp và dự đoán là khâu rất quan trọng trong học máy và trong khai phá dữ liệu, phát hiện tri thức. Kỹ thuật Support Vector Machine (SVM) được đánh giá là mạnh và tinh vi nhất hiện nay cho những bài toán phân lớp phi tuyến. Nhiều những ứng dụng đã và đang được xây dựng trên kỹ thuật SVM rất hiệu quả.
- **Mục đích, đối tượng và phạm vi nghiên cứu:**
- Nghiên cứu bài toán phân lớp quan điểm, cơ sở lý thuyết của phương pháp SVM và các vấn đề liên quan. Phân tích những giải pháp cho phép mở rộng và cải tiến để nâng cao hiệu quả ứng dụng của SVM. Đưa kỹ thuật mờ vào SVM cho phép phân chia không gian dữ liệu một các tốt hơn, nhằm loại bỏ những vùng không được phân lớp bằng SVM thông thường.
- Trình bày hướng áp dụng kỹ thuật SVM Cũng như những cải tiến, mở rộng của nó vào giải quyết một số các bài toán ứng dụng trong thực tiễn.
- Ứng dụng thuật toán phân lớp Vector SVM thông qua ngôn ngữ lập trình Python

2. Ý nghĩa khoa học và thực tiễn:

- SVM là một phương pháp phân lớp hiện đại và hiệu quả, nắm chắc phương pháp này sẽ tạo nền tảng giúp chúng ta trong việc phát triển các giải pháp phân loại và dự đoán,.., xây dựng được những ứng dụng quan trọng trong thực tế.

CHƯƠNG 2: NỘI DUNG CỦA ĐỀ TÀI NGHIÊN CỨU

I.Tìm hiểu về Support Vector Machine

1.1 Phát biểu bài toán

Support Vector Machine (SVM) là kỹ thuật mới đối với việc phân lớp dữ liệu, là phương pháp học sử dụng không gian giả thuyết các hàm tuyến tính trên không gian đặc trưng nhiều chiều, dựa trên lý thuyết tối ưu và lý thuyết thống kê.

Trong kỹ thuật SVM không gian dữ liệu nhập ban đầu sẽ được ánh xạ vào không gian đặc trưng và trong không gian đặc trưng này mặt siêu phẳng phân chia tối ưu sẽ được xác định.

Ta có tập S gồm *e* các mẫu học $S = \{(x_1,y_1), (x_2,y_2), (x_3,y_3).....(x_e,y_e)\} \subseteq (X \times Y)^e$

Với một Vector đầu vào *n* chiều $\vec{x}_i \in R^n$ thuộc lớp I hoặc lớp II (tương ứng nhãn $y_i = 1$ đối với lớp I và $y_i = -1$ đối với lớp II). Một tập mẫu học được gọi là tầm thường nếu tất cả các nhãn là bằng nhau.

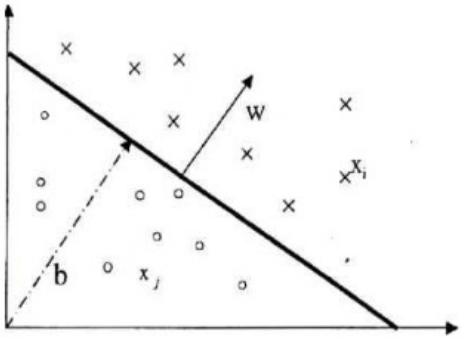
Đối với các dữ liệu phân chia tuyến tính, chúng ta có thể xác định được siêu phẳng $f(x)$ mà nó có thể chia tập dữ liệu. Khi đó, với mỗi siêu phẳng nhận được ta có: $f(x) \geq 0$ nếu đầu vào x thuộc lớp dương, và $f(x) < 0$ nếu x thuộc lớp âm

$$f(x) = w.x + b = \sum_{j=1}^{n} w_i x_i + b$$

$$y_i f(x_i) = y_i (w.x_i + b) \geq 0 \ , \ i=1,....,l$$

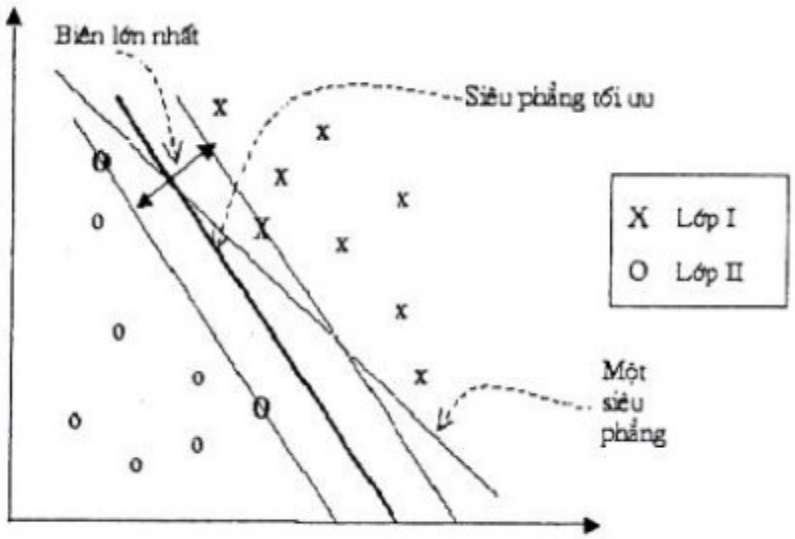
Trong đó *w* là vector pháp tuyến *n* chiều và *b* là giá trị ngưỡng

Vector pháp tuyến *w* xác định chiều của siêu phẳng $f(x)$, còn giá trị ngưỡng *b* xác định khoảng cách giữa siêu phẳng và gốc.



Hình 2. 1: Phân tách theo siêu phẳng (w,b) trong không gian 2 chiều của tập mẫu

Siêu phẳng có khoảng cách với dữ liệu gần nhất là lớn nhất (tức có biến lớn nhất) được gọi là siêu phẳng tối ưu



Hình 2. 2: Siêu phẳng tối ưu

Mục đích đặt ra ở đây là tìm ra được một ngưỡng (*w*, *b*) phân chia tập mẫu vào các lớp có nhãn 1 (lớp I) và -1 (lớp II) nếu ở trên với khoảng cách là lớn nhất

1.1.1 Trình bày tóm tắt về phân lớp dữ liệu

- **Phân lớp dữ liệu** là một kỹ thuật trong khai phá dữ liệu được sử dụng rộng rãi nhất và được nghiên cứu mở rộng hiện nay.
- **Mục đích:** Để dự đoán những nhãn phân lớp cho các bộ dữ liệu hoặc mẫu mới.

  Đầu vào: Một tập các mẫu dữ liệu huấn luyện, với một nhãn phân lớp cho mỗi mẫu dữ liệu

  Đầu ra: Bộ phân lớp dựa trên tập huấn luyện, hoặc những nhãn phân lớp

Phân lớp dữ liệu dựa trên tập huấn luyện và các giá trị trong một thuộc tính phân lớp và dùng nó để xác định lớp cho dữ liệu mới

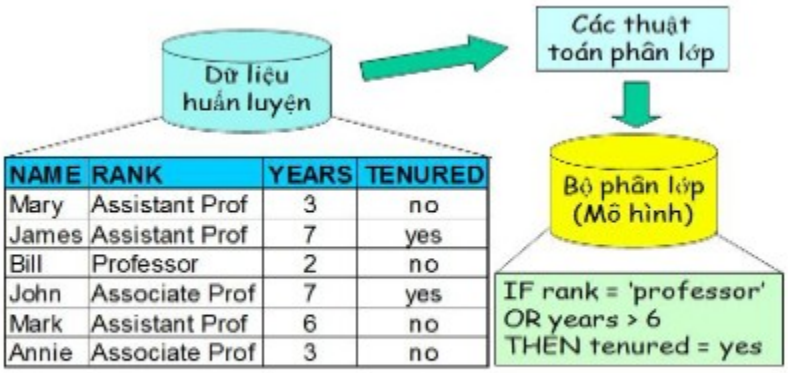
Kỹ thuật phân lớp dữ liệu được tiến hành bao gồm 2 bước:

Bước 1: Xây dựng mô hình từ tập huấn luyện

Bước 2: Sử dụng mô hình - kiểm tra tính đúng đắn của mô hình và dùng nó để phân lớp dữ liệu mới.

Bước 1: Xây dựng mô hình

- Mỗi bộ/mẫu dữ liệu được phân vào một lớp được xác định trước.
- Lớp của mỗi bộ/mẫu dữ liệu được xác định bởi thuộc tính gán nhãn lớp
- Tập các bộ/mẫu dữ liệu huấn luyện - tập huấn luyện - được dùng để xây dựng mô hình
- Mô hình được biểu diễn bởi các luật phân lớp, các cây quyết định hoặc các công thức toán học

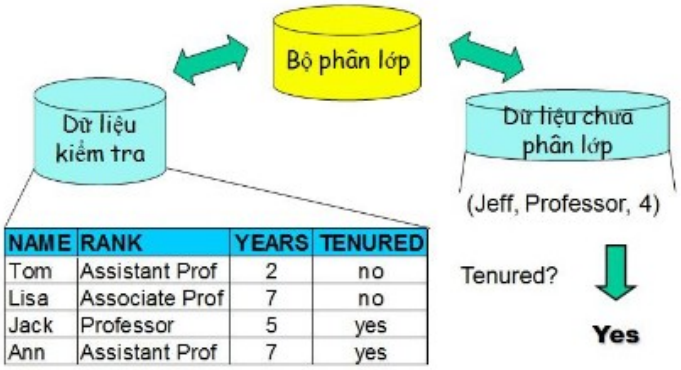


NAME	RANK	YEARS	TENURED
Mary	Assistant Prof	3	no
James	Assistant Prof	7	yes
Bill	Professor	2	no
John	Associate Prof	7	yes
Mark	Assistant Prof	6	no
Annie	Associate Prof	3	no

Hình 2.3: Ví dụ xây dựng mô hình

Bước 2: Sử dụng mô hình

- Phân lớp cho những đối tượng mới hoặc chưa được phân lớp
- Đánh giá độ chính xác của mô hình
    - Lớp biết trước của một bộ/mẫu dữ liệu đem kiểm tra được so sánh với kết quả thu được từ mô hình
    - Tỉ lệ chính xác bằng phần trăm các bộ/mẫu dữ liệu được phân lớp đúng bởi mô hình trong số các lần kiểm tra



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Lisa	Associate Prof	7	no
Jack	Professor	5	yes
Ann	Assistant Prof	7	yes

Hình 2.4: Sử dụng mô hình

1.1.2 Tại sao lại sử dụng thuật toán SVM trong phân lớp dữ liệu

- SVM rất hiệu quả để giải quyết bài toán dữ liệu có số chiều lớn (ảnh của dữ liệu biểu diễn gene, protein, tế bào)
- SVM giải quyết vấn đề *overfitting* rất tốt (dữ liệu có nhiễu và tách rời nhóm hoặc dữ liệu huấn luyến quá ít)
- Là phương pháp phân lớp nhanh
- Có hiệu suất tổng hợp tốt và hiệu suất tính toán cao

1.2 Thuật toán SVM

1.2.1 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 như: học máy, nhận dạng, trí tuệ nhân tạo,... . Trong khóa luận này, sẽ đi nghiên cứu phương pháp Support Vector Machine (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 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.

1.2.2 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 (*Supervised Learning*) được sử dụng cho phân lớp dữ liệu

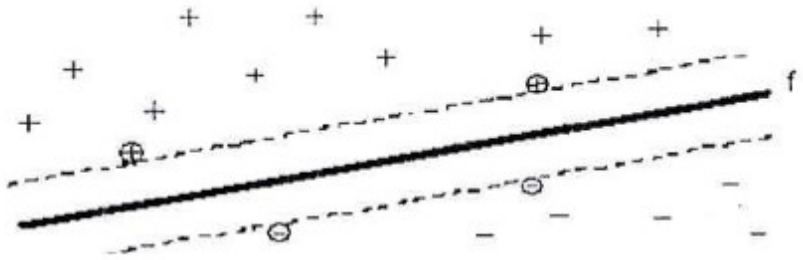
Là một phương pháp thử nghiệm, đưa ra một trong những phương pháp mạnh và chính xác nhất trong số các thuật toán nổi tiếng về phân lớp dữ liệu

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

1.2.3 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 bằng 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:



Hình 2. 5: Siêu phẳng phân chia dữ liệu học thành 2 lớp + và - với khoảng cách biên lớn nhất. Các điểm gần nhất (điểm được khoanh tròn) là các Support Vector.

1.2.4 Nội dung phương pháp

1.2.4.1 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 $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\}$ với $\forall i \in R^n$, thuộc vào hai lớp nhãn: $y_i \in \{-1, 1\}$ là nhãn lớp tương ứng của các x_i (-1 biểu thị lớp I, 1 biểu thị lớp II).

Ta có phương trình siêu phẳng của vector x_i trong không gian:

$$\vec{X_i}.\vec{W} + b = 0$$

$$\text{Đặt } f(\vec{X_i}) = \text{sign}\,(\vec{X_i}.\vec{W} + b) = \begin{cases} +1, \vec{X_i}.\vec{W} + b > 0 \\ -1, \vec{X_i}.\vec{W} + b < 0 \end{cases}$$

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

$$\text{Tìm min } \|\vec{w}\| \text{ với W thỏa mãn điều kiện sau:}$$

$$y_i(\sin\,(\vec{X_i}.\vec{W} + b)) \geq 1 \text{ với } \forall\, i \in \overrightarrow{1,n}$$

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 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.

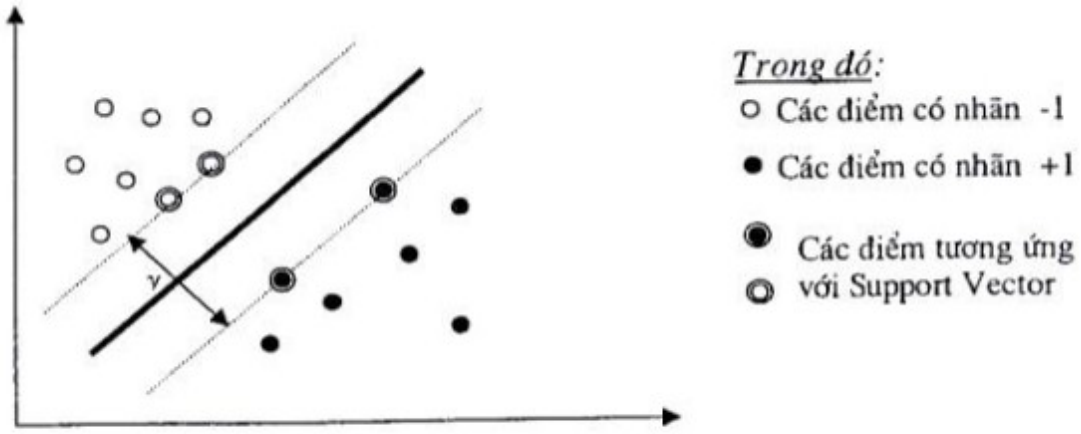
TÓM LẠI: Trong trường hợp nhị phân tách tuyến tính, việc phân lớp được thực hiện qua hàm quyết định *f(x) = sign(<w.x> + b)*, hàm này thu được bằng việc thay đổi vector chuẩn *w*, đây là vector để cực đại hóa viền chức năng.

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 đề này là xây dựng và kết hợp nhiều bộ phận 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-đối-một, một-đối-tất cả.*

1.2.4.2 Bài toán phân 2 lớp với SVM

Bài toán đặt ra là: Xác định hàm phân lớp để phân lớp các mẫu trong tương lai, nghĩa là với một mẫu dữ liệu mới x_i thì cần phải xác định x_i được phân vào lớp +1 hay lớp -1

Để xác định hàm phân lớp dựa trên phương pháp SVM, ta sẽ tiến hành tìm hai siêu phẳng song song sao cho khoảng cách *y* giữa chúng là lớn nhất có thể để phân tách hai lớp này ra làm hai phía. Hàm phân tách tương ứng với phương trình siêu phẳng nằm giữa hai siêu phẳng tìm được.



Hình 2. 6: Minh họa bài toán 2 phân lớp bằng phương pháp SVM

Các điểm mà nằm trên hai siêu phẳng phân tách được gọi là các Support Vector. Các điểm này sẽ quyết định đến hàm phân tách dữ liệu

1.2.4.3 Bài toán nhiều phân lớp với SVM

Để phân nhiều lớp thì kỹ thuật SVM nguyên thủy sẽ chia không gian dữ liệu thành hai 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-lớp sẽ là:

$$f_i(x) = w^i_i x + b_i$$

Những phần tử *x* là support Vector sẽ thỏa điều kiện

$$f_i(x) = \begin{cases} +1 \text{ nếu thuộc lớp i} \\ -1 \text{ nếu thuộc phần còn lại} \end{cases}$$

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 phân loại có k lớp (k>2), chiến lược "*một-đối-một*" sẽ tiến hành k(k-1)/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.

1.2.4.4 Các bước chính của phương pháp SVM

Phương pháp SVM yêu cầu dữ liệu được diễn tả như 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. Thường 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 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.

Kiểm thử tập dữ liệu Test.

II.So sánh và một số cải tiến

Một số phương pháp như mạng neuron, fuzzy logic, mạng fuzzy-neuron,... , cũng được sử dụng thành công để giải quyết bài toán phân lớp. Ưu điểm của các phương pháp này là không cần xác định mô hình toán đối của đối tượng (giải quyết tốt với các hệ thống lớn và phức tạp).

SVM có 2 đặc trưng cơ bản:

- Nó luôn kết hợp với các dữ liệu có ý nghĩa về mặt vật lý, do vậy dễ dàng giải thích được một cách tường minh.
- Cần một tập các mẫu huấn luyện rất nhỏ.

Phương pháp SVM hiện nay được xem là một công cụ mạnh và tinh vi nhất hiện nay cho những bài toán phân lớp phi tuyến. Nó có một số biến thể như C - SVC, V - SVC. Cải tiến mới nhất hiện nay của phương pháp SVM đã được công bố là thuật toán NNSRM (Nearest Neighbor Structural Risk Minimization) là sự kết hợp giữa hai kỹ thuật SVM và Nearest Neighbor.

CHƯƠNG III: ỨNG DỤNG VÀ CHƯƠNG TRÌNH

I.Tổng quan về xây dựng phần mềm

1.Khai báo những thư viện cần thiết

- Khai báo thư viện numpy, pandas, matplotlib, seaborn: Các thư viện này được sử dụng để làm việc với dữ liệu và vẽ biểu đồ.
- Khai báo thư viện sklearn: Thư viện chứa nhiều công cụ và thuật toán machine learning, bao gồm cả SVM và các công cụ đánh giá mô hình.
- Khai báo thư viện ipywidgets: Thư viện cho phép tạo các phần tương tác trong notebook.
- Khai báo thư viện IPython.display: Được sử dụng để hiển thị các widget và output trong notebook.

2.Tải lên dữ liệu có sẵn để tương tác

- Dữ liệu Iris dataset được tải từ sklearn.datasets và chuyển thành DataFrame sử dụng pandas để dễ dàng xử lý.
- Một cột mới được thêm vào DataFrame để lưu trữ nhãn của mỗi mẫu.

3.Xây dựng các Widget để tương tác

- Sử dụng ipywidgets để tạo các phần tương tác cho người dùng, bao gồm dropdown menu, thanh trượt và các nút.

- Người dùng có thể chọn kernel, điều chỉnh các siêu tham số của SVM và kích hoạt các chức năng khác nhau.

4.Xây dựng thuật toán SVM

- Hàm train_svm được tạo để huấn luyện một mô hình SVM dựa trên các siêu tham số được chọn và hiển thị kết quả đánh giá.
- Dữ liệu được chia thành tập huấn luyện và tập kiểm tra, sau đó chuẩn hóa và giảm chiều bằng PCA.
- Mô hình SVM được huấn luyện và đánh giá trên tập kiểm tra.

5.Widget để hiển thị hình minh họa (plot_decision_boundary)

- Hàm plot_decision_boundary nhận một mô hình SVM đã được huấn luyện và dữ liệu huấn luyện, sau đó vẽ biên phân loại của mô hình trên không gian 2D sau khi áp dụng PCA.
- Để vẽ biên phân loại, đầu tiên chúng ta tạo một lưới điểm trên không gian 2D sau khi giảm chiều dữ liệu bằng PCA.
- Tiếp theo, chúng ta dự đoán lớp cho từng điểm trong lưới bằng mô hình SVM.
- Sau đó, chúng ta vẽ contour plot để hiển thị biên phân loại trên không gian 2D.

6.Widget hiển thị ma trận tương quan (show_correlation_matrix)

- Hàm show_correlation_matrix vẽ ma trận tương quan của các đặc trưng trong dữ liệu Iris.
- Để vẽ ma trận tương quan, chúng ta sử dụng corr() để tính ma trận tương quan giữa tất cả các cặp đặc trưng.
- Sau đó, chúng ta sử dụng heatmap từ thư viện seaborn để hiển thị ma trận tương quan dưới dạng một biểu đồ heatmap.

7.Widget hiển thị phân phối của các đặc trưng (show_feature_distribution)

- Hàm show_feature_distribution vẽ các biểu đồ phân phối của các đặc trưng trong dữ liệu Iris.
- Đầu tiên, chúng ta tạo một grid để hiển thị các biểu đồ.
- Tiếp theo, chúng ta sử dụng vòng lặp để vẽ biểu đồ phân phối của từng đặc trưng bằng sns.histplot.
- Cuối cùng, chúng ta thiết lập tiêu đề và tên trục cho từng biểu đồ.

8.Widget hiển thị mức độ quan trọng của các đặc trưng (show_feature_importance):

- Hàm show_feature_importance sử dụng mô hình Random Forest để ước lượng mức độ quan trọng của các đặc trưng và vẽ biểu đồ thanh để hiển thị.
- Mức độ quan trọng của các đặc trưng được tính bằng phương pháp feature_importances_ của mô hình Random Forest.
- Sau đó, chúng ta sắp xếp các đặc trưng theo mức độ quan trọng giảm dần và vẽ biểu đồ thanh để hiển thị.

9.Widget so sánh hiệu suất của các mô hình (compare_models)

- Hàm compare_models so sánh hiệu suất giữa mô hình SVM và mô hình Random Forest bằng cách sử dụng cross-validation và vẽ biểu đồ để thể hiện kết quả.
- Chúng ta chạy mỗi mô hình trên dữ liệu sử dụng cross-validation với 5 folds để ước lượng độ chính xác.
- Kết quả độ chính xác trung bình của mỗi mô hình được lưu trữ và vẽ biểu đồ so sánh giữa chúng bằng sns.barplot.

II.Hướng cài đặt và sử dụng chương trình

1.Khởi động chương trình

- Khởi động chương trình trên Google colab hoặc Jupyter notebook
- Chạy đoạn mã của chương trình
- Tương tác với các Widget để xử lý dữ liệu

2.Chọn kernel

- Widget "Select Kernel" cho phép bạn chọn loại kernel cho mô hình SVM, bao gồm linear, polynomial và radial basis function (RBF). Chọn một trong các lựa chọn từ dropdown menu.

3.Điều chỉnh siêu tham số

- Các widget "C Parameter", "Gamma Parameter", và "Degree Parameter" cho phép bạn điều chỉnh các siêu tham số tương ứng của mô hình SVM. Bạn có thể di chuyển thanh trượt hoặc nhập giá trị trực tiếp vào ô textbox để điều chỉnh các giá trị này.

4.Huấn luyện mô hình

- Sau khi đã chọn kernel và điều chỉnh các siêu tham số, nhấn nút "Train SVM Model" để huấn luyện mô hình SVM với các tham số đã chọn.
- Kết quả của quá trình huấn luyện, bao gồm độ chính xác trên tập huấn luyện và tập kiểm tra, sẽ được hiển thị.

5.Hiển thị biên phân loại

- Sau khi huấn luyện mô hình, widget "Plot Decision Boundary" sẽ hiển thị biên phân loại của mô hình trên không gian 2D sau khi áp dụng phương pháp giảm chiều PCA.
- Bạn có thể thay đổi kích thước và độ phân giải của biểu đồ bằng cách điều chỉnh các tham số trong widget.

6.Hiển thị ma trận tương quan

- Widget "Show Correlation Matrix" sẽ hiển thị ma trận tương quan của các đặc trưng trong dữ liệu Iris dưới dạng một heatmap.
- Ma trận tương quan sẽ giúp bạn hiểu rõ hơn về mối quan hệ giữa các đặc trưng trong dữ liệu.

7.Hiển thị phân phối đặc trưng

- Widget "Show Feature Distribution" sẽ hiển thị phân phối của từng đặc trưng trong dữ liệu Iris dưới dạng các biểu đồ histogram.
- Bạn có thể thay đổi kích thước và số lượng cột của grid để xem biểu đồ cho từng đặc trưng.

8.Hiển thị mức độ quan trọng của các đặc trưng

- Widget "Show Feature Importance" sẽ hiển thị mức độ quan trọng của từng đặc trưng trong dữ liệu Iris dưới dạng biểu đồ thanh.
- Biểu đồ này sẽ giúp bạn xác định đặc trưng nào quan trọng nhất đối với mô hình SVM đã huấn luyện.

9.So sánh hiệu suất của các mô hình

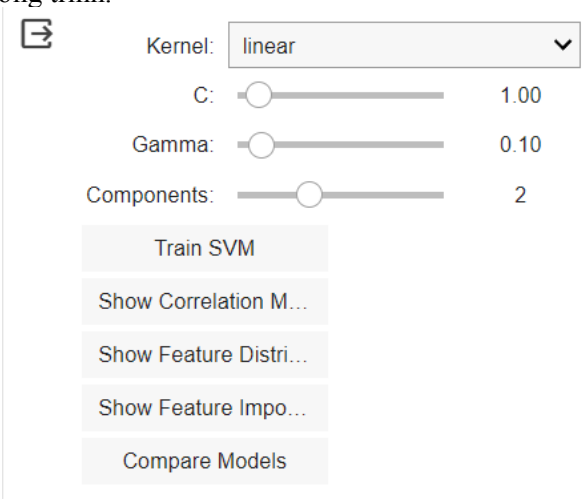
- Widget "Compare Models" sẽ so sánh hiệu suất giữa mô hình SVM và mô hình Random Forest bằng cách sử dụng kỹ thuật cross-validation.

- Kết quả của quá trình so sánh sẽ được hiển thị dưới dạng biểu đồ cột, giúp bạn so sánh độ chính xác của hai mô hình trên dữ liệu Iris.

III. Chi tiết cách sử dụng và các chức năng

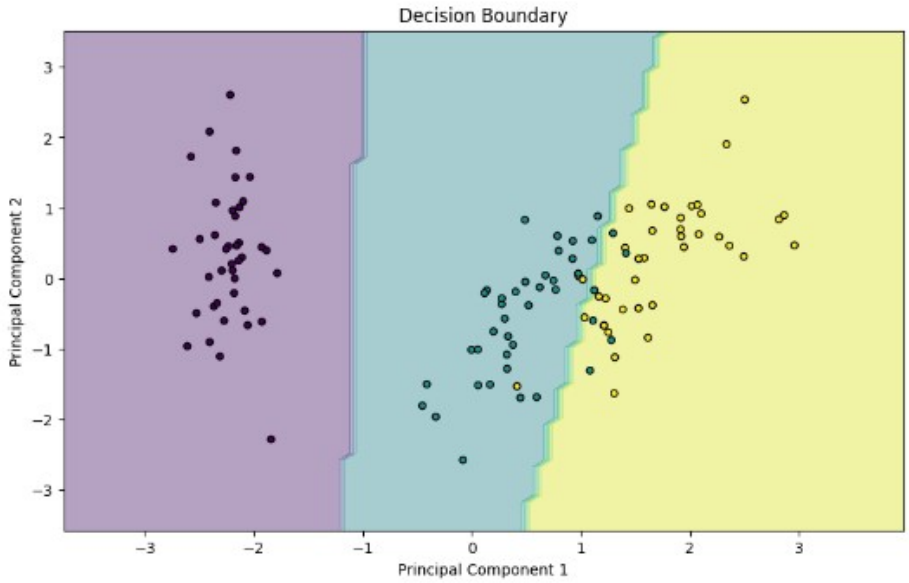
1. Sử dụng chương trình

- **Chạy mã:**
  - Nhấn Ctrl + Enter hoặc nhấp vào nút "RUN" bên cạnh mỗi ô code để chạy mã.
  - Widget tương tác sẽ xuất hiện dưới các ô code, cho phép bạn tương tác với các chức năng của chương trình.



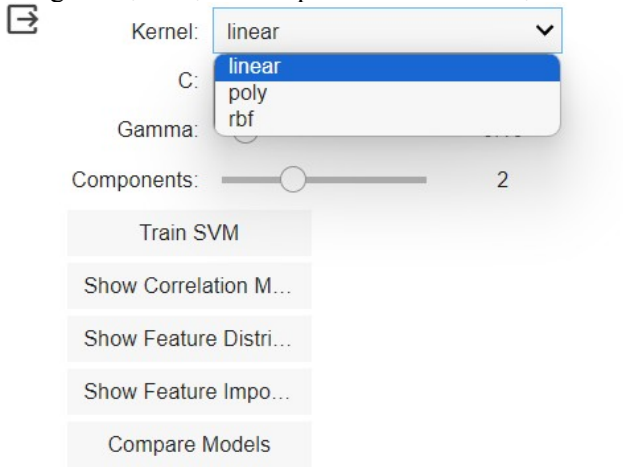
- **Tương tác với widget:**
  - Chọn các tùy chọn từ các dropdown menu hoặc điều chỉnh các thanh trượt để điều chỉnh các siêu tham số của mô hình SVM.
  - Nhấn nút để huấn luyện mô hình và hiển thị kết quả.
  - Khám phá các chức năng khác như vẽ biên phân loại, hiển thị ma trận tương quan, phân phối đặc trưng và mức độ quan trọng của các đặc trưng.
- **Thử nghiệm các tùy chọn:**
  - Thử nghiệm với các kernel khác nhau như linear, polynomial và RBF.
  - Thay đổi các siêu tham số như C parameter, gamma parameter và degree parameter để xem làm thế nào chúng ảnh hưởng đến hiệu suất của mô hình.
- **Nhận xét và phân tích kết quả:**
  - Quan sát biên phân loại được vẽ ra và cố gắng hiểu rõ cách mô hình phân loại dữ liệu.
  - Xem xét ma trận tương quan để đánh giá mối quan hệ giữa các đặc trưng.
  - Thẩm định phân phối của các đặc trưng và mức độ quan trọng của chúng đối với mô hình SVM.

```
Accuracy: 0.9
Confusion Matrix:
[[10  0  0]
 [ 0  7  2]
 [ 0  1 10]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       0.88      0.78      0.82         9
           2       0.83      0.91      0.87        11

    accuracy                           0.90        30
   macro avg       0.90      0.90      0.90        30
weighted avg       0.90      0.90      0.90        30
```

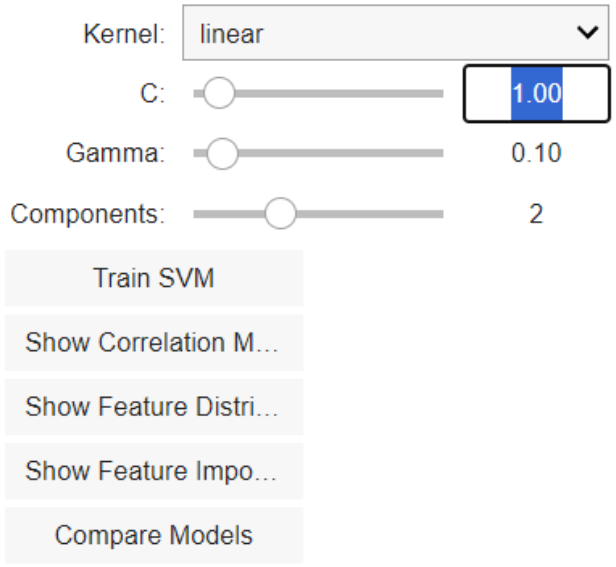


2.Chi tiết cách sử dụng các chức năng Widget

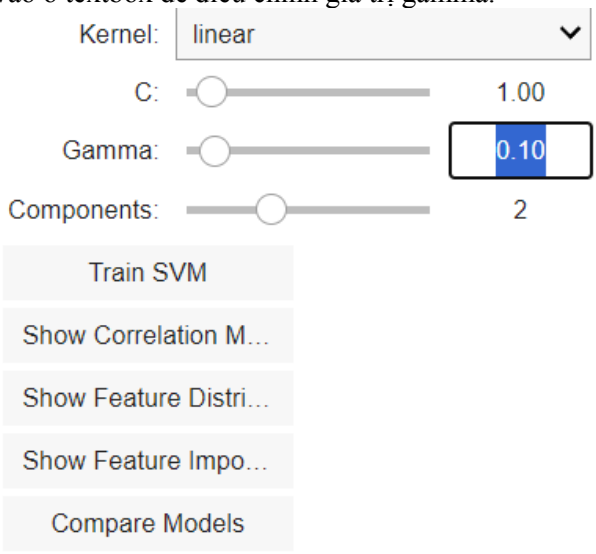
- **Select Kernel Widget:**
  - Widget này cho phép bạn chọn loại kernel cho mô hình SVM, bao gồm linear, polynomial và radial basis function (RBF).
  - Chọn một trong các lựa chọn từ dropdown menu để chọn kernel.



- **C Parameter Widget:**
  - Widget này cho phép bạn điều chỉnh siêu tham số C của mô hình SVM.
  - Di chuyển thanh trượt hoặc nhập giá trị trực tiếp vào ô textbox để điều chỉnh giá trị C.

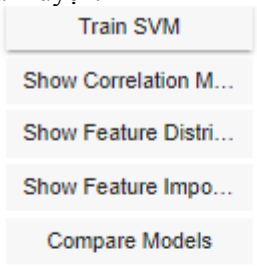


- **Gamma Parameter Widget:**
  - Widget này cho phép bạn điều chỉnh siêu tham số gamma của mô hình SVM.
  - Tương tự như C Parameter Widget, bạn có thể di chuyển thanh trượt hoặc nhập giá trị trực tiếp vào ô textbox để điều chỉnh giá trị gamma.



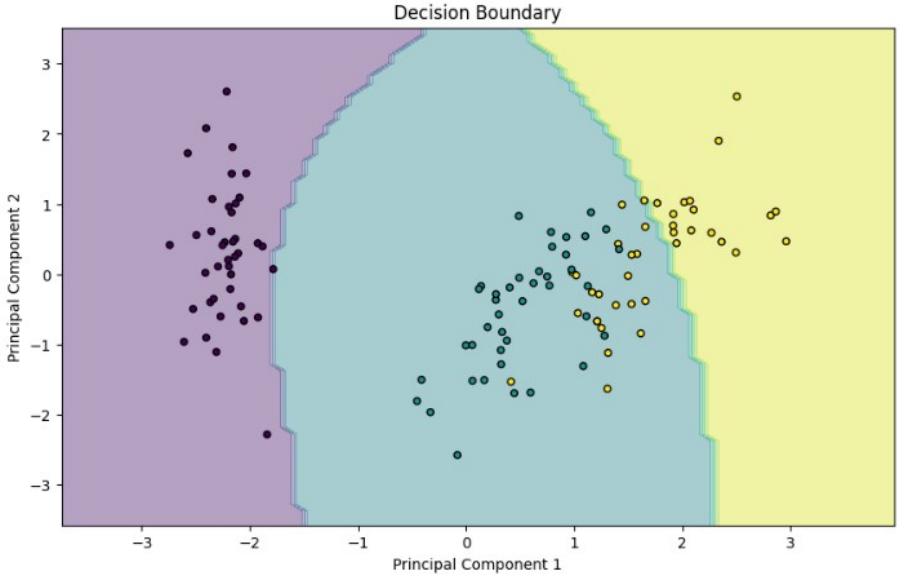
- **Degree Parameter Widget:**
  - Widget này chỉ xuất hiện khi bạn chọn kernel là polynomial.
  - Cho phép bạn điều chỉnh siêu tham số degree của kernel polynomial.
  - Di chuyển thanh trượt hoặc nhập giá trị trực tiếp vào ô textbox để điều chỉnh giá trị degree.

- **Train SVM Model Button:**
  - Nút này được sử dụng để huấn luyện mô hình SVM với các siêu tham số đã chọn.
  - Sau khi chọn kernel và điều chỉnh các siêu tham số, nhấn nút này để bắt đầu quá trình huấn luyện.

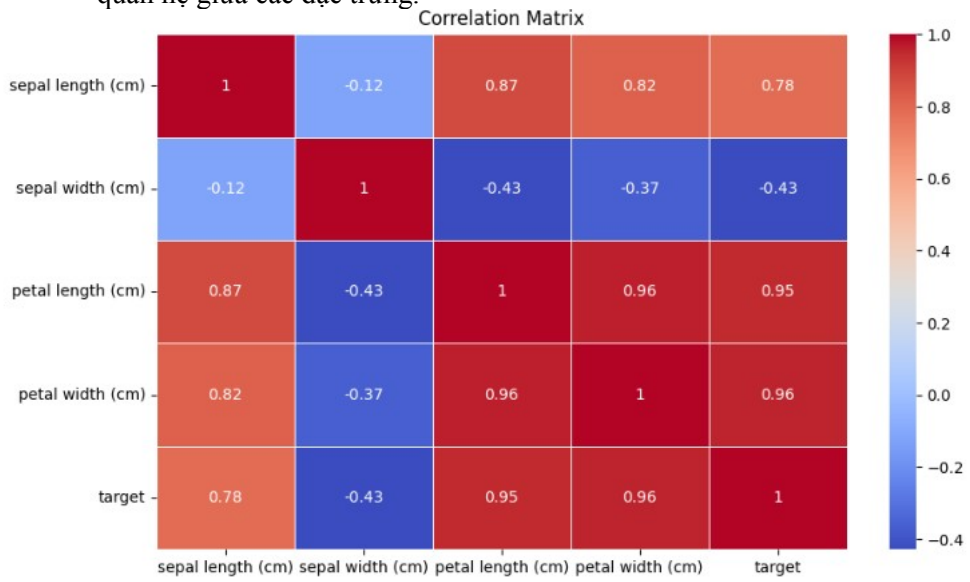



```
Accuracy: 0.8333333333333334
Confusion Matrix:
[[10  0  0]
 [ 0  9  0]
 [ 0  5  6]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       0.64      1.00      0.78         9
           2       1.00      0.55      0.71        11

    accuracy                           0.83        30
   macro avg       0.88      0.85      0.83        30
weighted avg       0.89      0.83      0.83        30
```

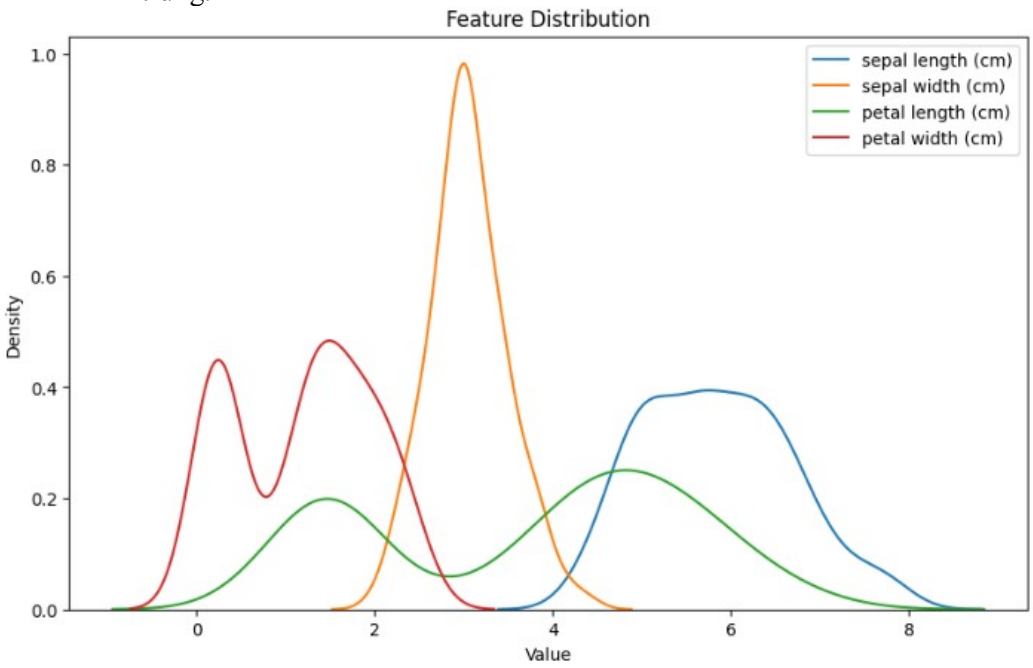
- **Plot Decision Boundary Button:**
  - Nút này được sử dụng để vẽ biên phân loại của mô hình SVM sau khi đã huấn luyện.
  - Biên phân loại sẽ được vẽ trên không gian 2D sau khi áp dụng phương pháp giảm chiều PCA.



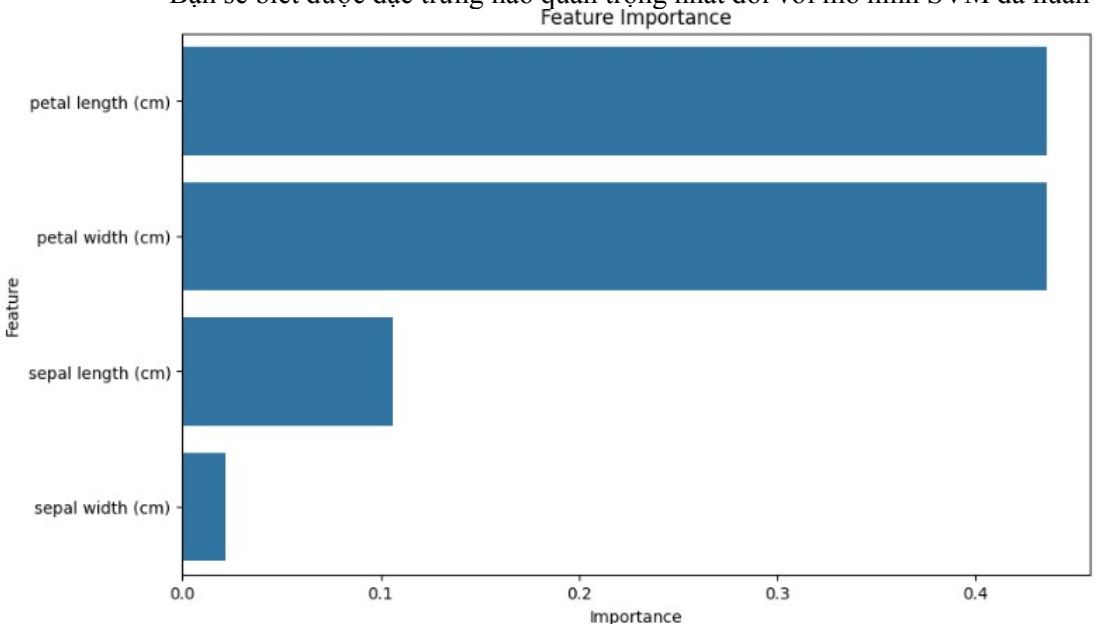
- **Show Correlation Matrix Button:**
  - Nút này được sử dụng để hiển thị ma trận tương quan của các đặc trưng trong dữ liệu.
  - Ma trận tương quan sẽ được hiển thị dưới dạng một heatmap để bạn có thể nhận biết mối quan hệ giữa các đặc trưng.



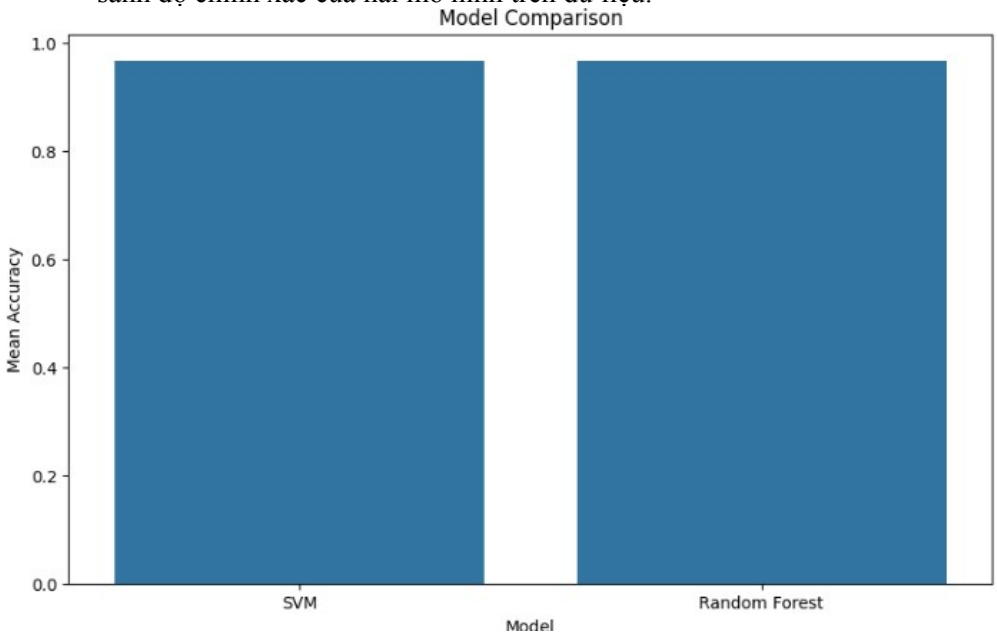
- **Show Feature Distribution Button:**
  - Nút này được sử dụng để hiển thị phân phối của từng đặc trưng trong dữ liệu dưới dạng các biểu đồ histogram.
  - Bạn có thể thay đổi kích thước và số lượng cột của grid để xem biểu đồ cho từng đặc trưng.



- **Show Feature Importance Button:**
  - Nút này được sử dụng để hiển thị mức độ quan trọng của từng đặc trưng trong dữ liệu dưới dạng biểu đồ thanh.
  - Bạn sẽ biết được đặc trưng nào quan trọng nhất đối với mô hình SVM đã huấn luyện.



- **Compare Models Button:**
  - Nút này được sử dụng để so sánh hiệu suất giữa mô hình SVM và mô hình Random Forest bằng cách sử dụng kỹ thuật cross-validation.
  - Kết quả của quá trình so sánh sẽ được hiển thị dưới dạng biểu đồ cột để bạn có thể so sánh độ chính xác của hai mô hình trên dữ liệu.



CHƯƠNG IV: ĐÁNH GIÁ KẾT QUẢ NGHIÊN CỨU VÀ KẾT LUẬN

TÀI LIỆU THAM KHẢO

https://machinelearningcoban.com/2017/04/09/smv/
https://scikit-learn.org/stable/modules/svm.html
https://www.datacamp.com/tutorial/svm-classification-scikit-learn-python