**Thuật Toán SVM (Support Vector Machine)**

Thuật toán SVM **(Support Vector Machine)** là một trong những thuật toán phân loại mạnh mẽ và phổ biến trong machine learning. SVM được sử dụng cho cả các tác vụ phân loại và hồi quy.

SVM hoạt động bằng cách tạo ra siêu mặt phẳng (hyperplane) trong không gian nhiều chiều để phân tách các lớp dữ liệu. Đối với bài toán phân loại nhị phân, SVM cố gắng tìm ra siêu mặt phẳng tối ưu sao cho khoảng cách từ siêu mặt phẳng này đến các điểm dữ liệu gần nhất (các vector hỗ trợ, support vectors) là lớn nhất.

Các loại kernel trong **Support Vector Machine** (SVM) là các phương pháp để biến đổi không gian đặc trưng của dữ liệu, giúp SVM phát hiện và phân loại các mẫu không phân chia tuyến tính trở nên dễ dàng hơn. Dưới đây là một số loại kernel phổ biến:

1. **Linear Kernel**: Đây là kernel cơ bản nhất và thường được sử dụng khi dữ liệu có thể phân chia tuyến tính.
2. **Polynomial Kernel**: Kernel đa thức sử dụng đa thức để ánh xạ dữ liệu vào không gian cao chiều. Nó có hai tham số chính là bậc của đa thức và hệ số tự do.
3. **Radial Basis Function** (RBF) Kernel: Kernel RBF là một trong những kernel phổ biến nhất. Nó ánh xạ dữ liệu vào không gian vô hạn chiều bằng cách sử dụng hàm Gaussian. Nó có tham số sigma (gamma trong Sklearn) để điều chỉnh độ rộng của hàm Gaussian.
4. **Sigmoid Kernel**: Kernel sigmoid ánh xạ dữ liệu vào không gian cao chiều bằng cách sử dụng hàm sigmoid. Nó có thể hữu ích trong một số trường hợp đặc biệt.
5. **Custom Kernel**: Bạn cũng có thể định nghĩa các kernel tùy chỉnh để phù hợp với đặc tính cụ thể của dữ liệu.

Mỗi kernel có những ưu điểm và hạn chế riêng. Việc lựa chọn kernel phù hợp phụ thuộc vào cấu trúc dữ liệu và đặc tính của bài toán cụ thể mà bạn đang làm việc

* **Ưu điểm của SVM:**

- Hiệu suất tốt trong không gian cao chiều: SVM hiệu quả trong việc xử lý không gian đa chiều, có nhiều đặc trưng. Điều này giúp SVM phù hợp trong việc xử lý các vấn đề phân loại dữ liệu có nhiều chiều.

- Hỗ trợ lựa chọn hàm kernel: SVM có thể sử dụng các hàm kernel khác nhau để thích ứng với đa dạng dữ liệu, cho phép nó làm việc tốt với dữ liệu phi tuyến.

- Chống overfitting: Bằng cách sử dụng margin, SVM giúp giảm nguy cơ overfitting trong việc phân loại dữ liệu.

- Hiệu suất tốt với dữ liệu nhỏ: SVM thường hoạt động tốt khi có ít dữ liệu hơn so với một số thuật toán học máy khác.

* **Nhược điểm của SVM:**

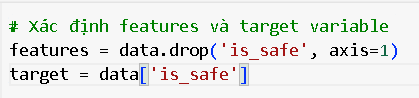
- Khả năng lựa chọn tham số: SVM yêu cầu lựa chọn các tham số quan trọng như hàm kernel, cost function, và gamma. Việc lựa chọn tham số phù hợp có thể ảnh hưởng đến hiệu suất của mô hình.

- Khó khăn khi xử lý dữ liệu lớn: SVM yêu cầu thời gian tính toán lớn khi xử lý dữ liệu lớn với số lượng mẫu lớn.

- Độ phức tạp tính toán: Trong một số trường hợp, việc tính toán của SVM có thể tốn nhiều thời gian, đặc biệt khi số lượng mẫu lớn.

- Khó hiểu kết quả: Khi sử dụng kernel, mô hình trở nên phức tạp hơn và đôi khi khó để hiểu cách mà mô hình phân loại các mẫu.

1. Xác định features và target variable



- features đang được xác định bằng cách loại bỏ cột 'is\_safe' từ dữ liệu. Đây có thể là cột chứa nhãn hoặc biến mục tiêu mà bạn muốn dự đoán. Vì vậy, mục tiêu của mô hình là phân loại hoặc dự đoán giá trị của 'is\_safe'.

- target đang được xác định bằng cột 'is\_safe', giả sử rằng đây là nhãn hoặc biến mục tiêu mà mô hình của bạn sẽ học để dự đoán hoặc phân loại.

1. Chia dữ liệu thành tập huấn luyện và tập kiểm tra



- features chứa các đặc trưng của dữ liệu.

- target là biến mục tiêu bạn muốn dự đoán hoặc phân loại.

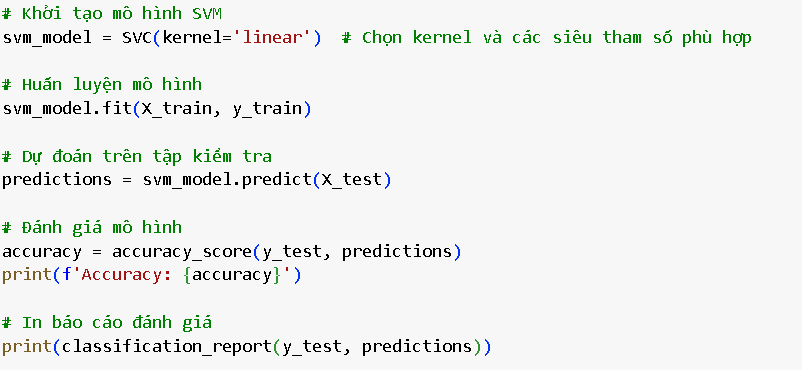
- Hàm train\_test\_split từ thư viện Scikit-learn (Sklearn) được sử dụng để chia dữ liệu thành tập huấn luyện và tập kiểm tra. Cụ thể:

- test\_size = 0.2 xác định tỷ lệ dữ liệu sẽ được chia cho tập kiểm tra là 20%.

- random\_state = 42 là giá trị seed để đảm bảo việc chia dữ liệu này có thể tái tạo được, tức là mỗi khi chạy, kết quả sẽ giống nhau.

=> Kết quả của hàm này sẽ là X\_train, X\_test, y\_train, và y\_test lần lượt chứa tập dữ liệu huấn luyện và kiểm tra cho đặc trưng và biến mục tiêu tương ứng. Điều này cho phép bạn huấn luyện mô hình trên X\_train, y\_train, và sau đó đánh giá mô hình trên X\_test, y\_test để đo lường hiệu suất của mô hình trên dữ liệu mới.

1. Khởi tạo và đánh giá mô hình
   1. Mã khởi tạo mô hình SVM và đánh giá



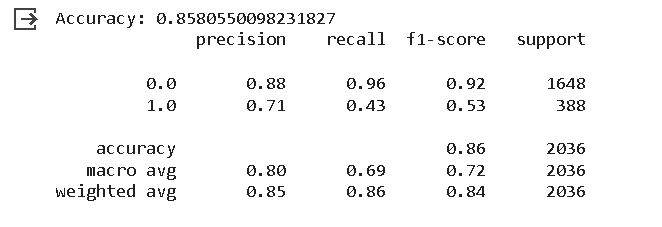
- Khởi tạo mô hình SVM: Được thực hiện thông qua SVC trong Scikit-learn, với sử dụng kernel 'linear' (tuyến tính). Đây là phần quan trọng, vì bạn có thể thay đổi kernel và điều chỉnh các siêu tham số khác để cải thiện hiệu suất của mô hình.

- Huấn luyện mô hình: Sử dụng X\_train và y\_train, mô hình SVM được huấn luyện trên tập dữ liệu huấn luyện.

- Dự đoán trên tập kiểm tra: Mô hình được sử dụng để dự đoán nhãn trên tập dữ liệu kiểm tra (X\_test).

- Đánh giá mô hình: Sử dụng accuracy\_score để tính toán độ chính xác của mô hình dựa trên dự đoán so với nhãn thực tế trên tập kiểm tra. Đồng thời, classification\_report cung cấp thông tin chi tiết về precision, recall, f1-score và support cho mỗi lớp.

* 1. Dữ liệu in ra



- Độ chính xác (Accuracy): Khoảng 85.80% là tỷ lệ các dự đoán đúng trên tập kiểm tra.

- Precision và Recall:

+ Precision cho lớp 0 (có label 0.0) là 88%, nghĩa là 88% các dự đoán là đúng cho lớp này.

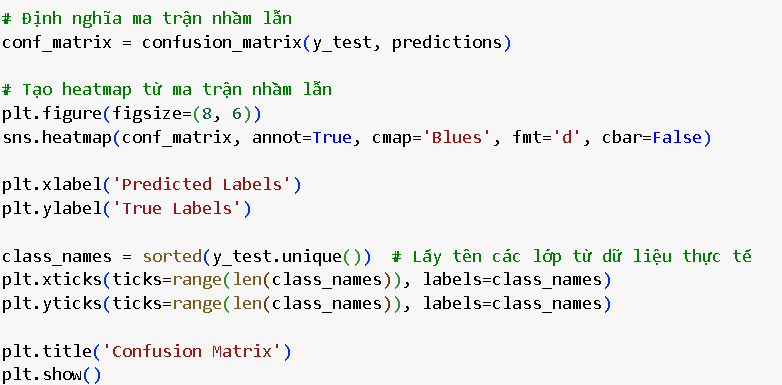
+ Precision cho lớp 1 (có label 1.0) chỉ khoảng 71%, có thể dự đoán không chính xác hơn cho lớp này.

+ Recall cho lớp 0 là 96%, còn lớp 1 chỉ là 43%. Điều này có thể chỉ ra mô hình dự đoán tốt hơn trong việc nhận dạng lớp 0 hơn là lớp 1.

- F1-score cho lớp 0 là 92%, còn lớp 1 chỉ là 53%. F1-score là sự kết hợp giữa precision và recall.

- Support: Số lượng thực tế của từng lớp trong tập kiểm tra.

1. Khởi tạo confusion matrix



- Mã trên đây đang tạo một biểu đồ heatmap từ confusion\_matrix để hình dung việc dự đoán của mô hình so với nhãn thực tế trên tập kiểm tra.

- Confusion matrix là một công cụ mạnh mẽ trong việc đánh giá hiệu suất của mô hình phân loại. Dưới đây là một số ưu và nhược điểm của confusion matrix:

* **Ưu điểm:**

- Cung cấp cái nhìn tổng quan về hiệu suất: Confusion matrix cung cấp thông tin chi tiết về các dự đoán đúng và sai của mô hình cho từng lớp, giúp cung cấp cái nhìn tổng quan về hiệu suất của mô hình.

- Đánh giá hiệu suất theo nhiều khía cạnh: Dựa trên các giá trị trong ma trận, bạn có thể tính toán precision, recall, F1-score và các thông số đánh giá khác cho từng lớp, giúp bạn hiểu rõ hơn về việc mô hình của bạn hoạt động như thế nào đối với từng lớp dữ liệu.

- Hỗ trợ tinh chỉnh mô hình: Khi bạn biết được lớp nào mà mô hình của bạn dự đoán tốt và lớp nào mà mô hình mắc nhiều sai sót, bạn có thể tinh chỉnh mô hình để cải thiện hiệu suất.

* **Nhược điểm:**

- Không cung cấp thông tin chi tiết về thực tế: Confusion matrix chỉ cung cấp thông tin về dự đoán của mô hình so với nhãn thực tế, không cung cấp thông tin về lý do tại sao mô hình sai lầm.

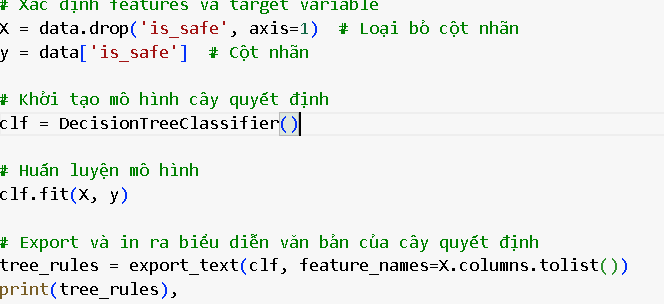
- Chưa hoàn hảo cho dữ liệu không cân bằng: Trong trường hợp dữ liệu mất cân bằng, khi có một lớp có số lượng quan sát nhiều hơn so với lớp khác, confusion matrix có thể trở nên không cân đối và không hiệu quả trong việc đánh giá hiệu suất của mô hình.

- Khó hiểu khi số lượng lớp lớn: Confusion matrix có thể trở nên khó hiểu khi có nhiều lớp, đặc biệt là khi số lượng lớp tăng lên.

* **Kết Luận:**

- Mặc dù có nhược điểm, confusion matrix vẫn là một công cụ quan trọng và hữu ích trong việc đánh giá hiệu suất của mô hình phân loại. Sự kết hợp giữa confusion matrix và các metrics khác như precision, recall, và F1-score sẽ cung cấp cái nhìn tổng quan và chi tiết hơn về hiệu suất của mô hình.

1. Biểu diễn cây quyết định dạng văn bản



- Đoạn mã trên (export\_text(clf, feature\_names=X.columns.tolist())) được sử dụng để xuất cấu trúc hoặc các quy tắc của một mô hình cây quyết định trong Python, thường là từ thư viện scikit-learn. Việc này có thể cung cấp một cái nhìn tổng quan về cách mô hình đưa ra các quyết định dựa trên các thuộc tính của dữ liệu.

* **Ưu điểm:**

- Hiểu được quyết định của mô hình: Xuất ra các quy tắc hoặc cấu trúc của mô hình cây quyết định giúp bạn hiểu cách mô hình đưa ra dự đoán. Điều này có thể hữu ích để giải thích mô hình cho người khác hoặc để kiểm tra xem các quy tắc quyết định có hợp lý không.

- Dễ đọc và diễn giải: Văn bản mô tả quy tắc dễ hiểu hơn so với việc xem trực tiếp cấu trúc cây, giúp dễ dàng diễn giải và trình bày thông tin.

* **Nhược điểm:**

- Giới hạn cho các cây quyết định: Phương pháp này chỉ áp dụng cho mô hình cây quyết định hoặc các mô hình tương tự. Các mô hình phức tạp hơn như mạng nơ-ron sâu không thể trực tiếp áp dụng cách này.

- Khả năng tái tạo mô hình hạn chế: Mô tả văn bản có thể không cung cấp đủ thông tin để tái tạo lại chính xác mô hình ban đầu nếu cần thiết.