TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ

KHOA CÔNG NGHỆ THÔNG TIN

THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH

HỌC KỲ I, NĂM HỌC 2025-2026

TÌM HIỂU VỀ SUPPORT VECTOR MACHINE (SVM) TRONG PHÂN LOẠI VĂN BẢN VÀ XÂY DỰNG CÔNG CỤ LỌC EMAIL SPAM

Giảng viên hướng dẫn:

ThS. Đoàn Phước miền

Sinh viên thực hiện:

Họ tên: Lê Quốc Tuấn

MSSV: 110123059

Lớp DA23TTB

Vĩnh Long, tháng 12 năm 2025

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

Vĩnh Long, ngày ……. tháng ……. năm ……….

Giáo viên hướng dẫn

(Ký tên và ghi rõ họ tên)

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

Vĩnh Long, ngày ……. tháng ……. năm ……….

Thành viên Hội đồng

(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Tôi xin chân thành cảm ơn quý Thầy Cô Khoa Công Nghệ Thông tin, Trường Kỹ Thuật và Công nghệ Trường Đại hoc Trà Vinh đã tận tình hướng dẫn, truyền đạt kiến thức, kỹ năng chuyên môn và tạo điều kiện thuận lợi để em hoàn thành đồ án này.

Tôi xin gửi lời cảm ơn đến thầy Đoàn Phước Miền người đã hướng dẫn, giúp đỡ và động viên tôi trong suốt quá trình thực hiện đồ án.

Cuối cùng tôi xin bày tỏ lòng biết ơn sâu sắc đến gia đình, bạn bè và những người đã luôn ủng hộ, động viên tôi trong suốt quá trình học tập và thực hiện đồ án này.

Do thời gian thực hiện có hạn, kiến thức còn nhiều hạn chế nên đồ án thực hiện không tránh khỏi những sai sót. Tôi rất mong nhận được những ý kiên góp ý từ thầy cô để tôi có thêm kinh nghiệm và kĩ năng để tiếp tục hoàn thiện đồ án hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

DANH SÁCH MỤC HÌNH ẢNH – BẢNG BIỂU

TÓM TẮT ĐỒ ÁN CƠ SƠ NGÀNH

Đề tài "Tìm hiểu về công nghệ Support Vector Machine (SVM) trong phân loại văn bản và xây dựng công cụ lọc Email spam" được thực hiện nhằm nghiên cứu và ứng dụng một thuật toán học máy có giám sát vào bài toán thực tiễn trong lĩnh vực công nghệ thông tin. Nội dung đồ án tập trung vào việc tìm hiểu cơ sở lý thuyết của thuật toán SVM, phân tích khả năng áp dụng của SVM trong bài toán phân loại văn bản, từ đó triển khai thử nghiệm mô hình để lọc email spam. Thông qua đề tài, sinh viên có cơ hội tiếp cận quy trình xây dựng một hệ thống học máy hoàn chỉnh, bao gồm xử lý dữ liệu, huấn luyện mô hình, đánh giá kết quả và phát triển công cụ minh họa.

Support Vector Machine (SVM) là một thuật toán học máy thuộc nhóm học có giám sát, được sử dụng rộng rãi trong các bài toán phân loại và hồi quy. Nguyên lý cốt lõi của SVM là tìm ra một siêu phẳng phân chia dữ liệu sao cho khoảng cách giữa siêu phẳng đó và các điểm dữ liệu gần nhất của mỗi lớp là lớn nhất. Các điểm dữ liệu nằm sát ranh giới phân chia được gọi là các vector hỗ trợ và đóng vai trò quyết định trong việc hình thành mô hình. Nhờ cơ chế tối ưu hóa biên phân cách, SVM có khả năng tổng quát hóa tốt và hoạt động hiệu quả trong các bài toán có số lượng đặc trưng lớn, đặc biệt là dữ liệu dạng văn bản.

Lọc email spam là bài toán phân loại tự động các thư điện tử thành hai nhóm chính: "Spam" (thư rác) và "Không Spam" (thư hợp lệ). Email spam thường chứa nội dung quảng cáo, lừa đảo hoặc liên kết độc hại, gây ảnh hưởng đến hiệu quả làm việc và tiềm ẩn nguy cơ mất an toàn thông tin. Việc áp dụng các phương pháp học máy, cụ thể là SVM, cho phép hệ thống tự động học từ các email đã được gán nhãn trước đó để phân loại các email mới một cách chính xác. Công cụ lọc email spam được xây dựng trong đề tài góp phần hỗ trợ người dùng giảm thiểu thư rác, nâng cao hiệu quả sử dụng email và đảm bảo an toàn thông tin trong môi trường số.

MỞ ĐẦU

1. Lý do chọn đề tài

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, dữ liệu số ngày càng gia tăng với tốc độ nhanh, đặc biệt là dữ liệu văn bản trong các hệ thống thư điện tử. Email đã trở thành phương tiện trao đổi thông tin quan trọng trong học tập, công việc và kinh doanh. Tuy nhiên, đi kèm với sự phát triển đó là tình trạng email spam ngày càng phổ biến, gây ảnh hưởng đến hiệu quả làm việc, làm lãng phí thời gian và tiềm ẩn nhiều nguy cơ về an toàn thông tin. Việc xử lý và phân loại email thủ công không còn phù hợp khi số lượng email ngày càng lớn.

Trước thực trạng đó, các phương pháp học máy đã và đang được ứng dụng rộng rãi để giải quyết bài toán phân loại dữ liệu tự động. Trong số các thuật toán học máy có giám sát, Support Vector Machine (SVM) là một thuật toán tiêu biểu, có khả năng phân loại hiệu quả và hoạt động tốt với dữ liệu có số chiều lớn như văn bản. Vì vậy, việc lựa chọn đề tài tìm hiểu công nghệ SVM và ứng dụng vào bài toán lọc email spam không chỉ mang ý nghĩa thực tiễn mà còn giúp sinh viên tiếp cận và vận dụng kiến thức học máy vào một bài toán cụ thể.

2. Mục tiêu nghiên cứu

Mục tiêu chính của đề tài là nghiên cứu và tìm hiểu cơ sở lý thuyết của thuật toán Support Vector Machine (SVM) trong học máy có giám sát, đồng thời phân tích khả năng ứng dụng của thuật toán này trong bài toán phân loại văn bản. Trên cơ sở đó, đề tài hướng đến việc xây dựng một mô hình phân loại email có khả năng nhận diện và phân biệt giữa email spam và email không spam.

Bên cạnh đó, đề tài còn nhằm giúp sinh viên nắm vững quy trình xây dựng một hệ thống học máy cơ bản, bao gồm thu thập dữ liệu, tiền xử lý dữ liệu văn bản, trích xuất đặc trưng, huấn luyện mô hình, đánh giá kết quả và triển khai công cụ minh họa. Qua quá trình thực hiện, sinh viên được củng cố kiến thức chuyên ngành và nâng cao kỹ năng nghiên cứu, thực hành trong lĩnh vực công nghệ thông tin.

3. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu của đề tài là thuật toán Support Vector Machine (SVM) thuộc nhóm học máy có giám sát và bài toán phân loại email spam trong lĩnh vực xử lý văn bản. Cụ thể, đề tài tập trung nghiên cứu nguyên lý hoạt động, đặc điểm và khả năng ứng dụng của SVM trong việc phân loại dữ liệu văn bản nhị phân.

Phạm vi nghiên cứu của đề tài giới hạn ở việc áp dụng SVM để phân loại email thành hai lớp "Spam" và "Không Spam" dựa trên nội dung văn bản email. Dữ liệu sử dụng là các bộ dữ liệu email đã được gán nhãn sẵn, phục vụ mục đích học tập và thực nghiệm. Đề tài không đi sâu vào các hệ thống lọc spam thương mại quy mô lớn hay các mô hình học sâu phức tạp, mà tập trung đánh giá hiệu quả của SVM trong điều kiện bài toán cơ bản.

4. Phương pháp nghiên cứu

Đề tài được thực hiện dựa trên sự kết hợp giữa phương pháp nghiên cứu lý thuyết và phương pháp nghiên cứu thực nghiệm:

- Phương pháp nghiên cứu lý thuyết: Tiến hành tìm hiểu, tổng hợp và phân tích các tài liệu, giáo trình và bài báo khoa học liên quan đến học máy có giám sát, thuật toán Support Vector Machine (SVM) và bài toán phân loại văn bản. Thông qua đó, đề tài làm rõ các khái niệm, nguyên lý hoạt động, ưu điểm và hạn chế của SVM, làm cơ sở cho việc áp dụng vào bài toán lọc email spam.

- Phương pháp nghiên cứu thực nghiệm: Trên cơ sở lý thuyết đã nghiên cứu, tiến hành xây dựng mô hình phân loại email spam bằng thuật toán SVM. Quá trình thực nghiệm bao gồm các bước: thu thập và tiền xử lý dữ liệu email, trích xuất đặc trưng từ văn bản, huấn luyện mô hình SVM, đánh giá kết quả phân loại thông qua các chỉ số đo lường và xây dựng công cụ demo lọc email spam. Kết quả thực nghiệm được phân tích nhằm đánh giá tính hiệu quả và khả năng ứng dụng của mô hình.

CHƯƠNG 1: TỔNG QUAN

1. Đặt vấn đề

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, dữ liệu số ngày càng gia tăng với tốc độ nhanh, đặc biệt là dữ liệu văn bản trong các hệ thống thư điện tử. Email hiện nay được sử dụng rộng rãi trong học tập, công việc và trao đổi thông tin hằng ngày. Tuy nhiên, song song với sự tiện lợi đó, tình trạng email spam ngày càng phổ biến và gây ra nhiều phiền toái cho người sử dụng. Email spam không chỉ làm quá tải hộp thư mà còn tiềm ẩn nguy cơ mất an toàn thông tin, lừa đảo và phát tán mã độc.

Việc phân loại email thủ công trở nên không khả thi khi số lượng email ngày càng lớn. Do đó, cần có các phương pháp tự động để nhận diện và phân loại email spam một cách hiệu quả. Học máy, đặc biệt là các thuật toán học có giám sát, đã chứng minh được khả năng giải quyết tốt các bài toán phân loại dữ liệu. Trong số đó, Support Vector Machine (SVM) là một thuật toán tiêu biểu, có hiệu quả cao trong phân loại văn bản nhờ khả năng xử lý dữ liệu có số chiều lớn và khả năng tổng quát hóa tốt. Vì vậy, việc nghiên cứu và ứng dụng SVM vào bài toán lọc email spam là một hướng tiếp cận phù hợp và có ý nghĩa thực tiễn

2. Mục đích nghiên cứu

Mục đích của đề tài là nghiên cứu tổng quan về thuật toán Support Vector Machine (SVM) trong lĩnh vực học máy có giám sát và phân tích khả năng ứng dụng của thuật toán này trong bài toán phân loại email spam. Thông qua việc tìm hiểu cơ sở lý thuyết của SVM, đề tài hướng đến việc làm rõ nguyên lý hoạt động, ưu điểm và tính hiệu quả của SVM khi áp dụng cho dữ liệu văn bản.

Bên cạnh đó, đề tài còn nhằm xây dựng một mô hình phân loại email cơ bản dựa trên SVM, giúp phân biệt giữa email spam và email không spam. Qua quá trình nghiên cứu và thực nghiệm, sinh viên có cơ hội củng cố kiến thức chuyên ngành, tiếp cận quy trình xây dựng một hệ thống học máy đơn giản và nâng cao kỹ năng vận dụng lý thuyết vào bài toán thực tế.

CHƯƠNG 2 NGHIÊN CỨU LÝ THUYẾT

2.1. Tìm hiểu mô hình học máy (Machine Learning Model)

2.1.1. Khái niệm mô hình học máy

- Trong lĩnh vực khoa học máy tính, học máy (Machine Learning – ML) là một nhánh của trí tuệ nhân tạo (AI) cho phép máy tính tự học từ dữ liệu để dự đoán, phân loại hoặc đưa ra quyết định mà không cần lập trình cụ thể từng quy tắc. Trọng tâm của học máy chính là mô hình học máy là thành phần thực hiện quá trình học và suy luận dựa trên dữ liệu.

- Về bản chất, mô hình học máy có thể được hiểu như một hàm toán học hoặc chương trình được hình thành thông qua quá trình "học" của thuật toán. Mục tiêu của mô hình là khám phá mối quan hệ tiềm ẩn giữa dữ liệu đầu vào và đầu ra, từ đó tổng quát hóa để có thể dự đoán chính xác kết quả cho những dữ liệu mới chưa từng gặp trước đó.

- Chẳng hạn, trong bài toán phân loại email spam, mô hình học máy được huấn luyện từ một tập dữ liệu gồm hàng nghìn email đã được gán nhãn "spam" và "không spam". Thông qua quá trình học, mô hình sẽ phát hiện ra các đặc trưng ngôn ngữ đặc trưng của thư rác, chẳng hạn như sự xuất hiện của các từ khóa như "free", "click here" hay "giảm giá". Khi gặp một email mới, mô hình có thể dựa vào những đặc trưng đã học để dự đoán xem đó có phải là thư rác hay không.

- Tóm lại, mô hình học máy chính là sản phẩm đầu ra của quá trình huấn luyện – nơi thuật toán học máy phân tích dữ liệu, điều chỉnh các tham số, và xây dựng nên một hệ thống có khả năng tự động nhận biết, phân loại hoặc dự đoán dựa trên kinh nghiệm rút ra từ dữ liệu quá khứ.

2.1.2. Cấu trúc và nguyên lý hoạt động của mô hình học máy

- Một mô hình học máy thường bao gồm ba thành phần cơ bản: dữ liệu đầu vào, hàm mô hình, và đầu ra dự đoán. Các thành phần này phối hợp với nhau để biến dữ liệu thô thành thông tin hữu ích, từ đó hỗ trợ các quyết định hoặc dự đoán.

+ Dữ liệu đầu vào (Input Data):

- Dữ liệu đầu vào là các thông tin mà mô hình sử dụng để học và đưa ra dự đoán. Dữ liệu có thể đa dạng, bao gồm văn bản, hình ảnh, âm thanh hoặc dữ liệu số. Trong ví dụ về lọc email spam, dữ liệu đầu vào là nội dung email, tiêu đề, và các đặc trưng như tần suất xuất hiện của từ khóa hoặc độ dài câu. Trước khi đưa vào mô hình, dữ liệu thường được xử lý qua các bước tiền xử lý, như chuẩn hóa, loại bỏ dữ liệu nhiễu, trích xuất đặc trưng (feature extraction), và mã hóa thành dạng số học.

+ Hàm mô hình (Model Fuction):

- Hàm mô hình là phần trung tâm, thể hiện mối quan hệ toán học giữa dữ liệu đầu vào và kết quả dự đoán. Trong quá trình huấn luyện, các tham số của hàm mô hình được tối ưu để giảm thiểu sai số giữa dự đoán và giá trị thực tế. Ví dụ, trong mô hình hồi quy tuyến tính, hàm mô hình được biểu diễn dưới dạng:

$$\boldsymbol{y = w_1 + x_1 + w_2 + x_2 + \ldots + w_n + x_n + b}$$

Trong đó:

- $\boldsymbol{w_i}$: là các trọng số được tối ưu hóa trong quá trình huấn luyện

- **b**: là hệ số chệch (bias)

- $\boldsymbol{x_i}$: là các đặc trưng của dữ liệu đầu vào

+ Đầu ra (output):

- Đầu ra của mô hình là kết quả dự đoán hoặc phân loại dựa trên dữ liệu đầu vào. Trong bài toán phân loại email spam, đầu ra là nhãn “spam” hoặc “not spam”. Đối với bài toán dự đoán giá nhà, đầu ra là giá trị liên tục dự đoán dựa trên đặc trưng của căn nhà.

2.1.3. Mối quan hệ giữa mô hình học máy

- Trong học máy, hai khái niệm thuật toán học máy (Machine Learning Algorithm) và mô hình học máy (Machine Learning Model) thường xuất hiện cùng nhau, nhưng bản chất và vai trò của chúng khác nhau rõ rệt.

- Thuật toán học máy là phương pháp hoặc quy trình toán học mà máy tính sử dụng để học từ dữ liệu. Nó định nghĩa cách tìm kiếm, tối ưu các tham số, và rút trích các mẫu từ dữ liệu đầu vào. Thuật toán học máy bao gồm các công thức, công cụ tính toán, và quy tắc điều chỉnh tham số sao cho mô hình học máy có thể hoạt động hiệu quả. Ví dụ, các thuật toán phổ biến gồm: Linear Regression, Logistic Regression, Decision Tree, Random Forest, Naive Bayes, Support Vector Machine (SVM), và các thuật toán Boosting như XGBoost hay LightGBM.

- Ngược lại, mô hình học máy là sản phẩm đầu ra của thuật toán học máy sau khi được huấn luyện trên dữ liệu. Mô hình chứa các tham số đã được tối ưu, cấu trúc hàm số, và các quy tắc học được từ dữ liệu huấn luyện. Mô hình chính là thực thể có thể được triển khai, lưu trữ, và sử dụng để dự đoán hoặc phân loại trên dữ liệu mới chưa từng gặp.

- Ví dụ minh họa: khi sử dụng thuật toán SVM để phân loại email spam, thuật toán SVM sẽ tìm siêu phẳng (hyperplane) tối ưu phân tách giữa thư rác và thư hợp lệ dựa trên tập dữ liệu huấn luyện. Quá trình này tạo ra một mô hình SVM với các tham số đã được học (ví dụ, trọng số vector hỗ trợ), có thể lưu lại và sử dụng để dự đoán email mới. Trong trường hợp này:

+ Thuật toán SVM = phương pháp học, cách máy tính học từ dữ liệu

+ Mô hình SVM = kết quả đã học được, sẵn sàng để phân loại email

- Thuật toán là cách học, mô hình là kết quả học. Mối quan hệ này rất quan trọng vì việc lựa chọn thuật toán phù hợp và huấn luyện hiệu quả sẽ quyết định chất lượng của mô hình. Một mô hình tốt là kết quả của thuật toán thích hợp, dữ liệu chất lượng, và quá trình huấn luyện tối ưu. Nhờ hiểu rõ mối quan hệ này, người phát triển có thể điều chỉnh thuật toán, tối ưu tham số và nâng cao hiệu suất mô hình, từ đó đảm bảo ứng dụng thực tế (như lọc email spam, dự đoán, phân loại hình ảnh) đạt kết quả chính xác và ổn định.

2.1.4. Quá trình huấn luyện mô hình (Model Training)

- Huấn luyện mô hình (Model Training) là giai đoạn trọng tâm trong học máy, nơi thuật toán được áp dụng vào dữ liệu để tìm ra các quy luật, mối quan hệ giữa đầu vào và đầu ra, từ đó xây dựng mô hình có khả năng dự đoán hoặc phân loại dữ liệu mới. Quá trình này quyết định trực tiếp chất lượng và hiệu quả của mô hình học máy.

- Các bước huấn luyện mô hình:

+ Chuẩn bị dữ liệu huấn luyện

- Dữ liệu huấn luyện là tập hợp các mẫu có đặc trưng (features) và kết quả mong muốn (labels). Trước khi huấn luyện, dữ liệu thường được tiền xử lý để loại bỏ nhiễu, xử lý giá trị thiếu, chuẩn hóa hoặc mã hóa các biến không phải số. Trong bài toán phân loại email spam, ví dụ, nội dung email được biến đổi thành vector số, các từ khóa quan trọng được trích xuất, và nhãn “spam” hay “not spam” được gán cho mỗi email.

+ Chọn thuật toán phù hợp:

- Việc chọn thuật toán phụ thuộc vào loại bài toán (phân loại, hồi quy, clustering, v.v.) và đặc trưng của dữ liệu. Ví dụ, SVM hoặc Naive Bayes thường được dùng cho phân loại văn bản, trong khi Linear Regression phù hợp với dự đoán số liên tục.

+ Huấn luyện mô hình:

- Thuật toán được áp dụng lên dữ liệu huấn luyện để tối ưu hóa các tham số của mô hình. Trong quá trình này, mô hình sẽ điều chỉnh các trọng số hoặc cấu trúc nội bộ sao cho sai số dự đoán trên dữ liệu huấn luyện được giảm thiểu. Các phương pháp tối ưu hóa phổ biến gồm Gradient Descent, Stochastic Gradient Descent, hay thuật toán tối ưu hóa cho từng loại mô hình cụ thể.

+ Đánh giá mô hình (Model Evaluation)

- Sau khi huấn luyện, mô hình được kiểm thử trên dữ liệu chưa thấy trước đó (tập test) để đánh giá khả năng tổng quát hóa. Các chỉ số đánh giá phổ biến bao gồm Accuracy, Precision, Recall, F1-score, hoặc Mean Squared Error cho bài toán hồi quy. Việc đánh giá giúp xác định mô hình có phù hợp để triển khai hay cần điều chỉnh thêm.
- Nếu hiệu suất chưa đạt yêu cầu, các siêu tham số (hyperparameters) như số lượng lớp trong mạng neural, giá trị C trong SVM, hay độ sâu cây quyết định có thể được điều chỉnh.

- Việc tối ưu hóa này giúp mô hình đạt hiệu quả cao hơn và giảm thiểu hiện tượng overfitting hoặc underfitting.

- Quá trình huấn luyện mô hình là giai đoạn học thực tế, biến dữ liệu thô thành một mô hình có khả năng dự đoán chính xác. Một mô hình được huấn luyện tốt không chỉ ghi nhớ dữ liệu đã học mà còn tổng quát hóa tốt trên dữ liệu mới, từ đó đáp ứng yêu cầu thực tế trong các ứng dụng như lọc email spam, phân loại hình ảnh, dự đoán hành vi người dùng, hay các bài toán dự đoán khác.

2.1.5. Quá trình triển khai mô hình (Model Deloyment)

- Sau khi mô hình học máy đã được huấn luyện và đánh giá, bước tiếp theo là triển khai mô hình (Model Deployment) để sử dụng trong môi trường thực tế. Triển khai mô hình là giai đoạn quan trọng giúp chuyển mô hình từ trạng thái lý thuyết sang ứng dụng thực tế, từ đó tạo ra giá trị cho các tổ chức hoặc sản phẩm.

- Các bước triển khai mô hình:

+ Tích hợp vào hệ thống

- Mô hình được đóng gói và tích hợp vào ứng dụng hoặc dịch vụ thông qua API, microservice, hoặc module phần mềm. Trong ví dụ về lọc email spam, mô hình SVM sau khi huấn luyện có thể được triển khai trên máy chủ xử lý email, nhận dữ liệu đầu vào là các email mới, và trả về kết quả phân loại "spam" hoặc "not spam".

+ Xử lý dữ liệu đầu vào theo thời gian thực

- Trong môi trường triển khai, dữ liệu mới liên tục được sinh ra. Mô hình phải có khả năng nhận dữ liệu, tiền xử lý, trích xuất đặc trưng và đưa ra dự đoán nhanh chóng. Đối với hệ thống email, việc này đảm bảo các thư rác được phân loại tự động và tức thì.

+ Theo dõi hiệu suất mô hình

- Triển khai không chỉ là "bật chạy" mô hình mà còn cần giám sát hiệu quả dự đoán. Các chỉ số như Accuracy, Precision, Recall, hoặc thời gian phản hồi được theo dõi liên tục để đảm bảo mô hình hoạt động ổn định và chính xác.

+ Cập nhật và bảo trì mô hình

- Dữ liệu trong thực tế thay đổi theo thời gian (ví dụ, email spam liên tục xuất hiện với nội dung mới). Do đó, mô hình cần được huấn luyện lại hoặc tinh chỉnh định kỳ để duy trì hiệu suất. Việc quản lý phiên bản mô hình (Model Versioning) giúp theo dõi các bản cập nhật, phục hồi mô hình cũ nếu cần, và giảm rủi ro khi triển khai.

+ Bảo mật và quản lý dữ liệu

- Khi triển khai, mô hình thường xử lý dữ liệu nhạy cảm, như nội dung email hoặc thông tin cá nhân. Do đó, việc bảo mật dữ liệu và tuân thủ các quy định về quyền riêng tư là bắt buộc, đồng thời đảm bảo mô hình không bị lạm dụng hoặc tác động xấu bởi dữ liệu đầu vào.

- Quá trình triển khai mô hình là bước chuyển đổi từ mô hình huấn luyện sang ứng dụng thực tế, cho phép tổ chức khai thác giá trị từ các dự đoán và phân loại mà mô hình cung cấp. Một triển khai thành công bao gồm tích hợp hệ thống, xử lý dữ liệu, giám sát hiệu suất, cập nhật mô hình định kỳ và bảo mật dữ liệu, đảm bảo mô hình hoạt động hiệu quả, ổn định và lâu dài.

2.2. Phân loại các phương pháp học máy

- Trong lĩnh vực trí tuệ nhân tạo, học máy (Machine Learning) được chia thành nhiều phương pháp khác nhau dựa trên cách mà mô hình học từ dữ liệu. Mỗi phương pháp được thiết kế để giải quyết những loại bài toán và mục tiêu riêng biệt. Ba nhóm chính được công nhận rộng rãi nhất là học có giám sát (Supervised Learning), học không giám sát (Unsupervised Learning) và học tăng cường (Reinforcement Learning). Các phương pháp này khác nhau ở mức độ can thiệp của con người, dạng dữ liệu sử dụng, và mục tiêu học tập mà mô hình hướng đến.

2.2.1. Học máy có giám sát (Supervised Learning)

- Học có giám sát là hình thức học mà dữ liệu đầu vào (input) đi kèm với nhãn đầu ra (output) tương ứng. Mục tiêu của mô hình là học được mối quan hệ giữa đầu vào và đầu ra, từ đó dự đoán chính xác đầu ra cho dữ liệu mới chưa từng thấy.

- Trong quá trình huấn luyện, mô hình được "giám sát" bằng cách so sánh dự đoán của nó với nhãn thực tế, sau đó điều chỉnh tham số để giảm sai số (error). Quá trình này lặp lại cho đến khi mô hình đạt được độ chính xác mong muốn.

- Học máy có giám sát thường được áp dụng trong hai nhóm bài toán chính:

+ Bài toán phân loại (Classification)

- Mục tiêu: Dự đoán nhãn rời rạc cho một đối tượng.

- Ví dụ: Phân loại Email thành "Spam" hoặc "không Spam", nhận dạng khuôn mặt, phân loại văn bản cảm xúc (tích cực/tiêu cưc).

- Các thuật toán phổ biến: Support Vector Machine (SVM), Naive Bayes, Decision Tree, Random Forest, Logistic Regression, k-Nearest Neighbors (kNN).

+ Bài toán hồi quy (Regression)

- Mục tiêu: Dự đoán giá trị liên tục của biến đầu ra.

- Ví dụ Dự đoán giá nhà, dự đoán nhiệt độ, dự báo doanh thu.

- Thuật toán phổ biến: Linear Regression, Polynomial Regression, Support Vector Regression (SVR).

- Điểm mạnh của học có giám sát là độ chính xác cao khi có dữ liệu gán nhãn đầy đủ. Tuy nhiên, việc chuẩn bị dữ liêu huấn luyện là tốn ém và đòi hỏi chuyên môn, vì cần lượng lớn mẫu có nhãn đúng.

2.2.2. Học không giám sát (Unsupervised Learning)

- Khác với học có giám sát, học không giám sát sử dụng dữ liệu đầu vào không có nhãn. Mục tiêu của mô hình là tự phát hiện ra cấu trúc tiềm ẩn hoặc quy luật trong dữ liệu, chẳng hạ như phân nhóm, giảm chiều dữ liệu hoặc sự tương đồng giữa các phần tử trong tập dữ liệu. Thay vì được "chỉ dẫn" phải học gì, mô hình tự tìm kiếm mối liên hệ hoặc sự tương đồng giữa các phần tử trong tập dữ liệu.

- Các nhóm bài toán phổ biến trong học không giám sát gồm:

+ Phân cụm (Clustering):

- Mục tiêu: Nhóm các đối tượng có đặc điểm tương đồng vào cùng một cụm.

- Ví dụ: Phân nhóm khách hàng có hành vi mua sắm tương tự, nhóm tin theo chủ dề, phát hiện cộng đồng trong mạng xã hội.

- Thuật toán tiêu biểu: K-Means, Hierarchical Clustering, DBSCAN.

+ Giảm chiều dữ liệu (Dimensionnality Reduction)

- Mục tiêu: Giảm số lượng đặc trưng trong dữ liệu mà vẫn giữ lại thông tin quan trọng nhất.

- Ví dụ: Nén hình ảnh, tiền xử lý dữ liệu cho học sâu (Deep Learning), trực quan hóa dữ liệu nhiều chiều.

- Thuật toán phổ biến: Principal Component Analysis (PCA), t-SNE, Autoencoder

+ Phát hiện bất thường (Anomaly Detection)

- Mục tiêu: Xác định các phần tử hiếm hoặc khác biệt so với phần lớn dữ liệu.

- Ứng dụng: Phát hiện gian lận thẻ tín dụng, phát hiện lỗi trong hệ thống hoặc tín hiệu cảm biến bất thường.

- Điểm mạng của học không giám sát là khả năng khai phá tri thức từ dữ liệu thô, đặc biệt hữu ích khi không có hoặc rất ít dữ liệu gán nhãn. Tuy nhiên, kết quả thường khó đánh giá khách quan, do không có "đáp án đúng" để so sánh.

2.2.3. Học tăng cường (Reinforcement Learning)

- Học tăng cường là một hướng học máy trong đó mô hình (gọi là tác nhân – agent) học cách hành động thông qua tương tác liên tục với môi trường (environment). Mục tiêu của tác nhân là tối đa hóa phần thưởng (reward) mà nó nhận được thông qua quá trình thử và sai (trial and error).

- Ví dụ điển hình về học tăng cường bao gồm:

+ Xe tự lái học cách điều khiển an toàn.

+ Robot học cách di chuyển hoặc gấp đồ vật.

+ Hệ thống đề xuất học cách gợi ý sản phầm phù hợp nhất cho người dùng.

+ Các chương trình chơi cờ (như AlphaGo của Googlr DeepMind) tự học cách chơi tối ưu thông qua hàng triệu ván đấu.

- Ưu điểm của học tăng cường là khả năng học linh hoạt và thích nghi với môi trường thay đổi, không phụ thuộc vào dữ liệu gán nhãn sẵn. Tuy nhiên, nhượt điểm là thười gian huấn luyện lâu và chi phí tính toán cao, vì mô hình phải thực hiện rất nhiều lần thử nghiệm.

***Tổng kết:**

- Ba hướng tiếp cận chính trong học máy là học có giám sát, học không giám sát và học tăng cường đại diện cho ba cơ chế khác nhau trong cách máy tính hoc từ dữ liệu.

- Sự kết hợp linh hoạt giữa các phương pháp này đã tạo nên nền tảng cho nhiều ứng dụng trong thực tiễn, từ nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên, hệ thống gợi ý, dự báo tài chính, cho đến phát hiện gian lận và tự động hóa thông minh. Nhờ đó, học máy trở thành một trong những công nghệ cốt lõi thúc đẩy cuộc cách mạng công nghiệp 4.0, góp phần định hình nên các hệ thống thông minh trong đời sống hiện đại.

2.3. Một số mô hình học máy phổ biến

- Trong lĩnh vực học máy, có rất nhiều mô hình được phát triển nhằm giải quyết các bài toán khác nhau từ phân loại, hồi quy, phân cụm cho đến dự báo và tối ưu hóa. Mỗi mô hình mang trong mình một giả định học tập khác nhau về dữ liệu, và việc lựa chọn mô hình phù hợp là yếu tố quyết định đến hiệu quả của hệ thống.

- Những mô hình tiêu biểu, được sử dụng rộng rãi nhất trong thực tiễn và nghiên cứu.

2.3.1. Hồi quy tuyến tính (Linear Regression)

- Hồi quy tuyến tính là mô hình học có giám sát cơ bản nhất, được sử dụng để dự đoán giá trị liên tục. Mục tiêu của mô hình là tìm mối quan hệ tuyến tính giữa biến đầu vào (đặc trưng – features) và biến đầu ra (mục tiêu – target). Hồi quy tuyến tính tối ưu các trọng số sao cho tổng sai số bình phương nhỏ nhất (Least Squares Error).

- Công thức:

$$y = w_0 + w_1 x_1 + w_2 + \cdots + w_n x_n$$

- Trong đó

y: *Giá trị dự đoán*

x_i: *Biến đầu vào*

w_i: *Trọng số mà mô hình học được*

- Ưu điểm: Dễ hiểu, dễ triển khai, kết quả có thể giải thích rõ ràng, thời gian huấn luyện nhanh, phù hợp với dữ liệu có quan hệ tuyến tính.

- Nhược điểm: Hiệu quả kém khi dữ liệu có quan hệ phi tuyến tính, dễ bị ảnh hưởng bởi nhiễu và điểm ngoại lai.

2.3.2. Hồi quy logistic (Logistic Regression)

- Logistic Regression thực chất là mô hình phân loại nhị phân, được sử dụng rộng rãi trong học có giám sát. Thay vì dự đoán giá trị liên tục, mô hình này dự đoán xác suất để một mẫu thuộc về một lớp nhất định (ví dụ: “spam” hoặc “không spam”).

- Công thức xác suất:

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 + \cdots + w_n x_n)}}$$

Nếu xác suất lớn hơn 0.5 → mô hình gán nhãn 1, ngược lại gán nhãn 0.

- Ưu điểm: dễ triển khai, kết quả dễ diễn giải, hiệu quả trong các bài toán phân loại tuyến tính.

- Hạn chế: Không phù hợp cho dữ liệu phi tuyến phức tạp, phụ thuộc mạnh vào chất lượng và phân phối dữ liệu đầu vào.

2.3.3. Cây quyết định (Decision Tree)

- Decision Tree là mô hình trực quan mô phỏng cách con người ra quyết định, thông qua việc chia nhỏ dữ liệu thành các nhánh dựa trên giá trị của đặc trưng. Mỗi nút trong cây đại diện cho một điều kiện, còn nhánh lá biểu diễn kết quả dự đoán. Cây được xây dựng bằng cách chọn đặc trưng chia dữ liệu sao cho độ thuần khiết của tập dữ liệu tăng cao nhất, thường đo bằng chỉ số Gini hoặc Entropy.

- Ưu điểm: Dễ hiểu, dễ diễn giải, biểu diễn được mối quan hệ phi tuyến, có thể xử lý cả dữ liệu định tính lẫn định lượng.

- Hạn chế: Dễ bị overfitting (quá khớp) nếu cây quá sâu, nhạy cảm với biến nhiễu hoặc thay đổi dữ liệu nhỏ.

2.3.4. Rừng ngẫu nhiên (Random Forest)

- Random Forest là một tập hợp (ensemble) của nhiều cây quyết định. Thay vì dựa vào một cây duy nhất, mô hình này kết hợp kết quả của hàng trăm cây để đưa ra quyết định cuối cùng bằng cách bỏ phiếu (voting) hoặc trung bình (averaging). Ý tưởng chính là mỗi cây có thể sai khác nhau, nhưng khi kết hợp lại, lỗi trung bình sẽ giảm đáng kể. Kỹ thuật này giúp mô hình ổn định hơn, giảm hiện tượng quá khớp và hoạt động tốt trên nhiều loại dữ liệu.

- Ưu điểm: Độ chính xác cao, ít overfitting., có thể đo được mức độ quan trọng của từng đặc trưng.

- Hạn chế: Kích thước mô hình lớn, khó diễn giải, thời gian huấn luyện lâu hơn so với cây đơn.

2.3.5. Máy vector hỗ trợ (Support Vector Machine – SVM)

- SVM là một trong những mô hình mạnh mẽ nhất trong học có giám sát, đặc biệt là phân loại nhị phân. Ý tưởng chính của SVM là tìm một siêu phẳng (hyperplane) phân tách các lớp dữ liệu sao cho khoảng cách (margin) giữa siêu phẳng và các điểm dữ liệu gần nhất là lớn nhất. Trong trường hợp dữ liệu không thể phân tách tuyến tính, SVM sử dụng hàm nhân (kernel trick) để chiếu dữ liệu lên không gian có chiều cao hơn, giúp tách các lớp một cách rõ ràng hơn.

- Ưu điểm: Hiệu quả cao trong không gian đặc trưng lớn (nhiều chiều), hoạt động tốt với dữ liệu phức tạp, phi tuyến.

- Hạn chế: Thời gian huấn luyện chậm khi tập dữ liệu lớn, khó điều chỉnh tham số kernel và margin tối ưu.

2.3.6. Mạng nơ-ron nhân tạo (Artificial Neural Network – ANN)

- Mạng nơ-ron nhân tạo được lấy cảm hứng từ cấu trúc và cơ chế hoạt động của não người. Mạng bao gồm nhiều tầng (layer) chứa các nơ-ron (neuron) – đơn vị xử lý cơ bản. Mỗi nơ-ron nhận đầu vào, tính tổng có trọng số, sau đó áp dụng một hàm kích hoạt (activation function) để tạo đầu ra. Khi các tầng được kết nối với nhau, mô hình có khả năng học các đặc trưng phức tạp và biểu diễn mối quan hệ phi tuyến sâu sắc trong dữ liệu.

- Công thức cơ bản của một nơ-ron:

$$y = f(w_0 + w_1x_1 + w_2 + \cdots + w_nx_n + b)$$

- Trong đó:

f: *Hàm kích hoạt*

b: *bias(hệ số điều chỉnh)*

- Ưu điểm: Có khả năng học biểu diễn phức tạp, phù hợp với dữ liệu lớn, là nền tảng cho các mô hình học sâu (Deep Learning).

- Hạn chế: Cần nhiều dữ liệu và tài nguyên tính toán, Khó giải thích "vì sao" mô hình đưa ra kết quả (black-box model).

2.4. Thuật toán SVM (Support Vector Machine)

- Support Vector Machine (SVM) là một trong những thuật toán học có giám sát mạnh mẽ và phổ biến nhất trong lĩnh vực học máy, đặc biệt hiệu quả trong các bài toán phân loại (classification) và hồi quy (regression). Ý tưởng cốt lõi của SVM là tìm ra một siêu phẳng (hyperplane) tối ưu để phân tách dữ liệu của các lớp khác nhau với khoảng cách (margin) lớn nhất.

Giả sử ta có một tập dữ liệu huấn luyện gồm các cặp (x_i, y_i), trong đó:

- $x_i$là vector đặc trưng đầu vào,
- $y_i \in \{-1, +1\}$là nhãn lớp.

SVM cố gắng tìm siêu phẳng $w^T x + b = 0$sao cho:

$$y_i = (w^T x + b) \geq 1, \forall i$$

Khoảng cách giữa siêu phẳng và các điểm gần nhất của mỗi lớp được gọi là **biên (margin)**, và SVM chọn siêu phẳng sao cho biên này **lớn nhất có thể**.

Bài toán tối ưu được viết lại như sau:

$$\min_{w,b} \frac{1}{2} |w|^2, \text{với điều kiện } y_i = (w^T x + b) \geq 1$$

Trong thực tế, dữ liệu thường không hoàn toàn tách biệt, nên ta thêm các biến lỏng (slack variables) và tham số điều chỉnh Cđể cân bằng giữa độ chính xác và khả năng tổng quát hóa (generalization).

SVM được xem là một phương pháp "đặt ranh giới tốt nhất" giữa các lớp, thay vì chỉ tìm ranh giới đơn thuần.

2.4.1. Khái niệm siêu phẳng (Hyperplane) và biên (Margin)

- Siêu phẳng (Hyperplane) là một mặt phẳng trong không gian nhiều chiều có nhiệm vụ chia dữ liệu thành hai lớp.

- Ví dụ, trong không gian hai chiều, siêu phẳng là một đường thẳng; trong không gian ba chiều, là một mặt phẳng.

- Biên (Margin) là khoảng cách từ siêu phẳng phân tách tới các điểm dữ liệu gần nhất thuộc hai lớp khác nhau.

- Mục tiêu của SVM là tìm siêu phẳng có biên lớn nhất tức là ranh giới an toàn nhất giữa hai lớp. Một biên lớn thường giúp mô hình tổng quát hóa tốt hơn cho dữ liệu mới, tránh overfitting.

- Nếu dữ liệu không thể tách tuyến tính, SVM vẫn có thể xử lý bằng cách chiếu dữ liệu lên không gian đặc trưng cao hơn nhờ hàm kernel, cho phép siêu phẳng trong không gian mới trở nên tuyến tính.

2.4.2. Vector hỗ trợ (Support Vectors)

- Các vector hỗ trợ (support vectors) là những điểm dữ liệu nằm sát biên phân tách nhất, tức là các điểm "khó phân loại" nhất.

- Chúng là các phần tử duy nhất ảnh hưởng trực tiếp đến vị trí của siêu phẳng. Điểm nào nằm xa biên thì không ảnh hưởng đến kết quả huấn luyện. Điều này giúp SVM có hiệu suất tổng quát hóa cao và hiệu quả tính toán, vì mô hình chỉ cần quan tâm đến một phần nhỏ dữ liệu.

- Một cách hình tượng, support vectors chính là "điểm neo" giúp xác định ranh giới giữa các lớp. Nếu ta bỏ đi bất kỳ điểm nào trong số này, ranh giới có thể thay đổi đáng kể.

2.4.3. Hàm Kernel và mở rộng phi tuyến tính

- Trong nhiều trường hợp, dữ liệu không thể phân tách tuyến tính trong không gian ban đầu. Để giải quyết điều đó, SVM sử dụng hàm kernel (kernel trick) — một kỹ thuật toán học giúp chiếu dữ liệu sang không gian đặc trưng cao hơn mà không cần tính toán trực tiếp các tọa độ trong không gian đó.

- Hàm kernel cho phép SVM phân tách dữ liệu phi tuyến bằng một siêu phẳng tuyến tính trong không gian mở rộng. Một số hàm kernel thông dụng gồm:

- **Linear kernel:** $K(x_i, x_j) = x_i^T x_j$

→ Dùng khi dữ liệu tách tuyến tính.

- **Polynomial kernel:** $K(x_i, x_j) = (x_i^T x_j + c)^d$

→ Cho phép ranh giới cong với bậc d.

- **Radial Basis Function (RBF kernel):** $K(x_i, x_j) = \exp\,(-\gamma \parallel x_i - x_j \parallel^2)$

→ Rất phổ biến vì mô tả được ranh giới phức tạp.

- **Sigmoid kernel:** $K(x_i, x_j) = \tanh\,(\alpha x_i^T x_j + c)$

→ Liên hệ với mạng nơ-ron nhân tạo.

- Nhờ kernel, SVM có thể giải quyết các bài toán khó như phân loại hình ảnh, nhận dạng chữ viết tay, lọc email spam, hay phân tích cảm xúc, nơi mà ranh giới giữa các lớp rất phức tạp.

2.4.4. Ưu điểm và hạn chế của SVM

Ưu điểm:

Hiệu quả cao với dữ liệu có số chiều lớn (đặc biệt trong xử lý văn bản, ảnh).

Hoạt động tốt ngay cả khi số lượng mẫu nhỏ.

Dựa trên vector hỗ trợ → mô hình gọn nhẹ, tổng quát hóa tốt.

Có thể mô hình hóa ranh giới phi tuyến bằng kernel.

Hạn chế:

Thời gian huấn luyện lâu khi dữ liệu quá lớn.

Việc chọn tham số C, γvà loại kernel ảnh hưởng lớn đến hiệu suất.

Khó diễn giải kết quả hơn so với các mô hình tuyến tính như Logistic Regression.

Không hiệu quả khi dữ liệu có quá nhiều nhiễu hoặc các lớp chồng lấn mạnh.

2.5. Ứng dụng của SVM trong lọc email spam

2.5.1. Tổng quan bài toán lọc email spam

- Lọc email spam là một trong những ứng dụng thực tiễn và quan trọng nhất của học máy. Mục tiêu của bài toán là phân loại một email bất kỳ vào hai nhóm: **ham (email hợp lệ, nhãn 0)** và **spam (email rác, nhãn 1)**. Trong thực tế, lượng email spam ngày càng tăng với nhiều hình thức như quảng cáo, lừa đảo tài chính, phát tán mã độc hoặc liên kết độc hại. Điều này không chỉ ảnh hưởng đến năng suất làm việc mà còn tiềm ẩn rủi ro bảo mật. Vì vậy, việc xây dựng một mô hình tự động có khả năng nhận diện email spam nhanh chóng và chính xác là vô cùng cần thiết.

2.5.2. Quy trình xử lý và biểu diễn dữ liệu văn bản (BoW, TF–IDF, stopwords, stemming)

Dữ liệu email thô không thể đưa trực tiếp vào mô hình, mà cần trải qua quá trình tiền xử lý và chuẩn hóa. Các bước chính gồm:

Làm sạch dữ liệu (Text Cleaning): chuyển toàn bộ văn bản về chữ thường, loại bỏ ký tự đặc biệt, URL, số, emoji, dấu câu và khoảng trắng thừa.

Loại bỏ stopwords: những từ không mang nhiều ý nghĩa phân loại như “và”, “thì”, “là”, “the”, “is” …

Stemming và Lemmatization: đưa từ về dạng gốc, ví dụ “running” → “run”, “studies” → “study”.

Vector hóa văn bản: sử dụng Bag of Words (BoW) hoặc TF–IDF. Trong đó, TF–IDF đặc biệt hiệu quả vì đánh giá được mức độ quan trọng của từ trong toàn bộ tập dữ liệu. Ví dụ, từ “free” thường xuất hiện nhiều trong email spam nên có trọng số TF–IDF cao.

2. 5.3. Ứng dụng mô hình SVM trong phân loại email spam

- Sau khi dữ liệu đã được vector hóa, mô hình SVM sẽ được huấn luyện để phân loại email. Mỗi email được biểu diễn thành một điểm trong không gian nhiều chiều. SVM tìm siêu phẳng (hyperplane) phân tách hai lớp: ham (0) và spam (1). Siêu phẳng được chọn sao cho khoảng cách (margin) giữa hai lớp là lớn nhất, giúp mô hình tổng quát hóa tốt hơn trên dữ liệu mới.

- Trong bài toán văn bản, **Linear Kernel** thường được sử dụng vì dữ liệu TF–IDF có tính tuyến tính cao, vừa nhanh vừa chính xác. RBF Kernel có thể dùng cho dữ liệu phi tuyến, nhưng với email spam thường không cần thiết.

- Quy trình huấn luyện gồm: chuẩn bị dữ liệu (cleaning, stopwords, stemming), chuyển đổi văn bản sang BoW hoặc TF–IDF, chia dữ liệu thành tập huấn luyện và kiểm thử, huấn luyện mô hình SVM, dự đoán nhãn, và cuối cùng đánh giá bằng các chỉ số như Accuracy, Precision, Recall, F1-Score và Confusion Matrix. Trong bài toán spam, Precision và Recall đặc biệt quan trọng: nếu nhầm email hợp lệ thành spam (false positive) sẽ gây mất thông tin quan trọng, còn nếu nhầm spam thành ham (false negative) thì nguy hiểm hơn vì có thể chứa lừa đảo hoặc mã độc.

2.5.4. Đánh giá và so sánh SVM với các mô hình khác

- Để đánh giá hiệu quả, SVM thường được so sánh với các mô hình khác:

Naive Bayes: rất nhanh, phù hợp với dữ liệu văn bản, nhưng giả định đặc trưng độc lập nên kém ổn định hơn SVM.

Logistic Regression: học nhanh hơn SVM, hoạt động tốt với dữ liệu TF–IDF, nhưng độ chính xác thường thấp hơn khi dữ liệu phức tạp.

SVM: mạnh mẽ với dữ liệu văn bản nhiều chiều, tối ưu hóa ranh giới phân tách tốt, giảm overfitting, đặc biệt hiệu quả khi dùng Linear Kernel, TF–IDF.

- Kết luận: Naive Bayes nhanh nhất, Logistic Regression cân bằng nhưng đôi khi kém ổn định, còn SVM thường cho độ chính xác cao và ổn định nhất trong đa số bài toán email spam. Vì vậy, SVM trở thành một trong những mô hình tiêu chuẩn cho các hệ thống lọc spam hiện đại.

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1. Mô tả bài toán

Email spam (thư rác) là các thư điện tử không mong muốn, thường chứa nội dung quảng cáo, lừa đảo hoặc mã độc, gây ảnh hưởng đến người dùng và hệ thống thư điện tử. Ngược lại, email hợp lệ (ham) là những thư phục vụ cho mục đích trao đổi thông tin chính đáng.

Bài toán đặt ra trong đề tài là xây dựng một mô hình phân loại email, tự động xác định một email đầu vào thuộc một trong hai lớp: Spam; Không spam (Ham)

Đây là một bài toán phân loại nhị phân (Binary Classification) trong học máy có giám sát, trong đó: Dữ liệu đầu vào: nội dung văn bản của email, Nhãn đầu ra: spam hoặc ham

Mục tiêu của bài toán là xây dựng mô hình có khả năng: Phân loại email với độ chính xác cao, Hoạt động hiệu quả trên dữ liệu văn bản có số chiều lớn, Có khả năng tổng quát hóa tốt với email mới chưa xuất hiện trong tập huấn luyện

3.2. Thu nhập dữ liệu và cài đặt môi trường

3.2.1. Nguồn dữ liệu

Dữ liệu sử dụng trong đề tài được thu thập từ **kho dữ liệu Kaggle**, một nền tảng chia sẻ dữ liệu phục vụ nghiên cứu và học máy. Bộ dữ liệu được lựa chọn có nguồn gốc từ tập **SpamAssassin**, là bộ dữ liệu phổ biến trong nghiên cứu bài toán lọc email spam.

Việc lựa chọn dữ liệu từ Kaggle đảm bảo: dữ liệu có độ tin cậy cao, có nhẫn sẵn, phù hợp cho học máy có giám sát, được sử dụng rộng rãi trong các nghiên cứu liên quan.

3.2.2. Cấu trúc bộ dữ liệu

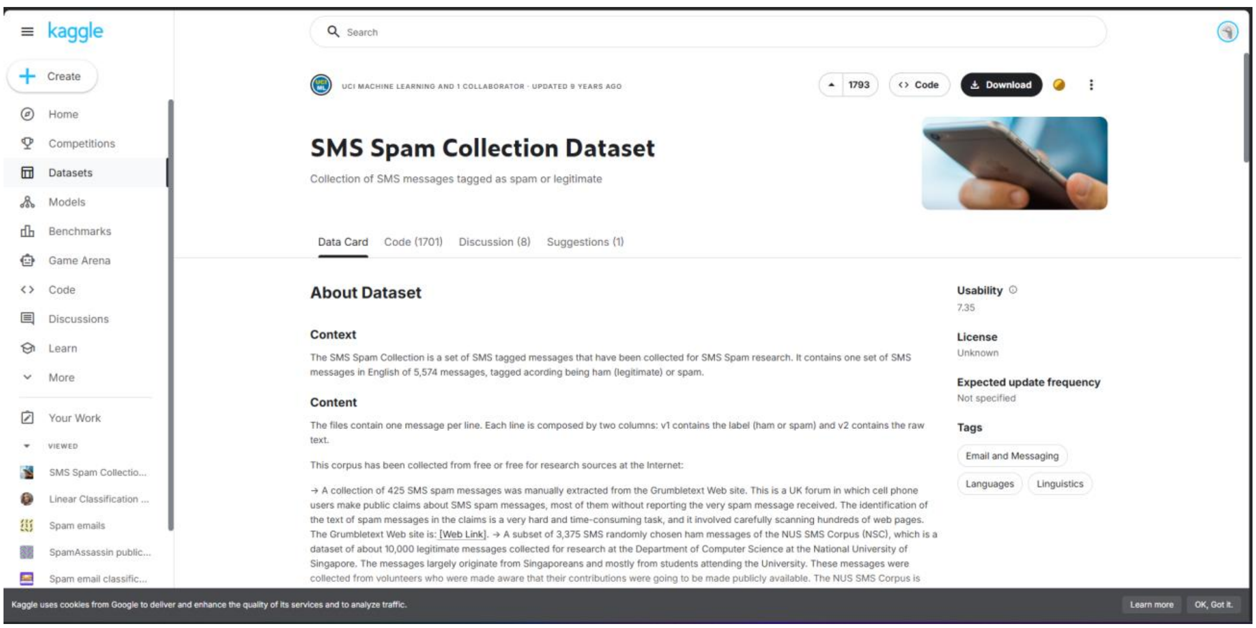
Sau khi tải về và giải nén, dữ liệu được tổ chức thành các thư mục:

- **easy_ham**: email hợp lệ, dễ phân biệt
- **hard_ham**: email hợp lệ nhưng có nội dung dễ nhầm lẫn với spam
- **spam**: email rác

Mỗi email được lưu dưới dạng tệp văn bản, chứa toàn bộ nội dung email gốc.

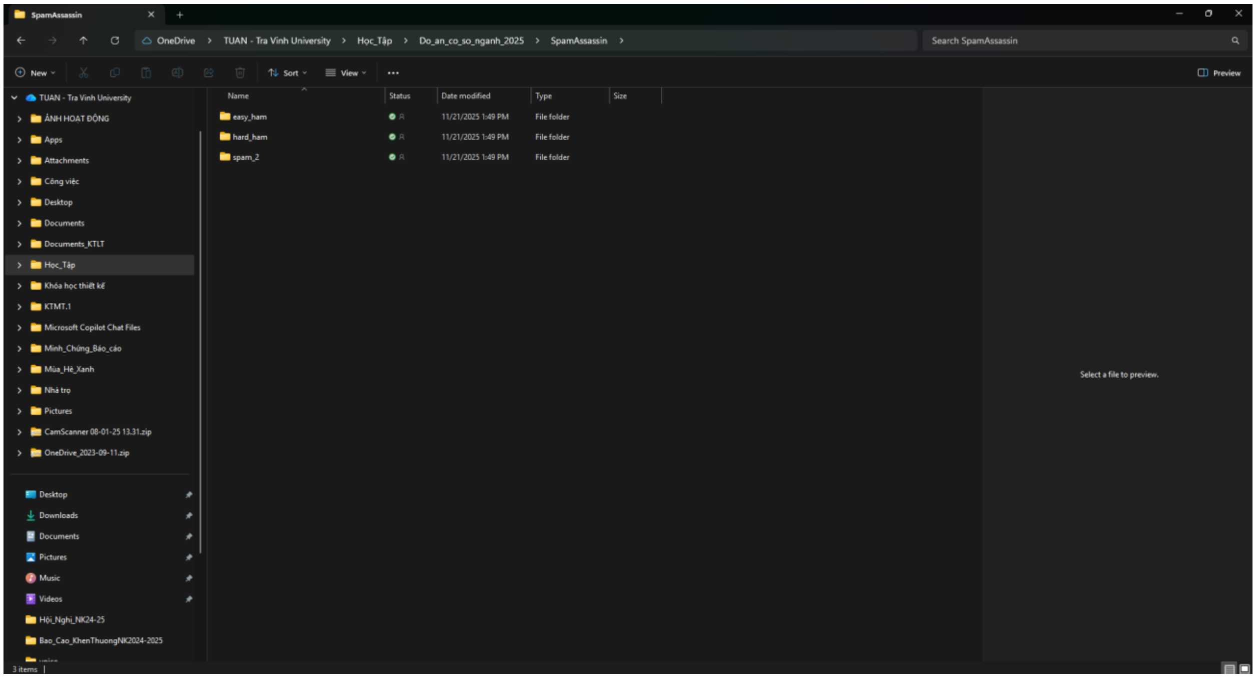
Bước 1: Tìm kiếm bộ dữ liệu trên trang kaggle.com

- Tìm kiếm trên trình duyệt trang kaggle.com sau đó đăng nhập, nhấp vào ô tìm kím và tìm dữ liệu phù hợp.



Bước 2: Tải bộ dữ liệu và giải nén

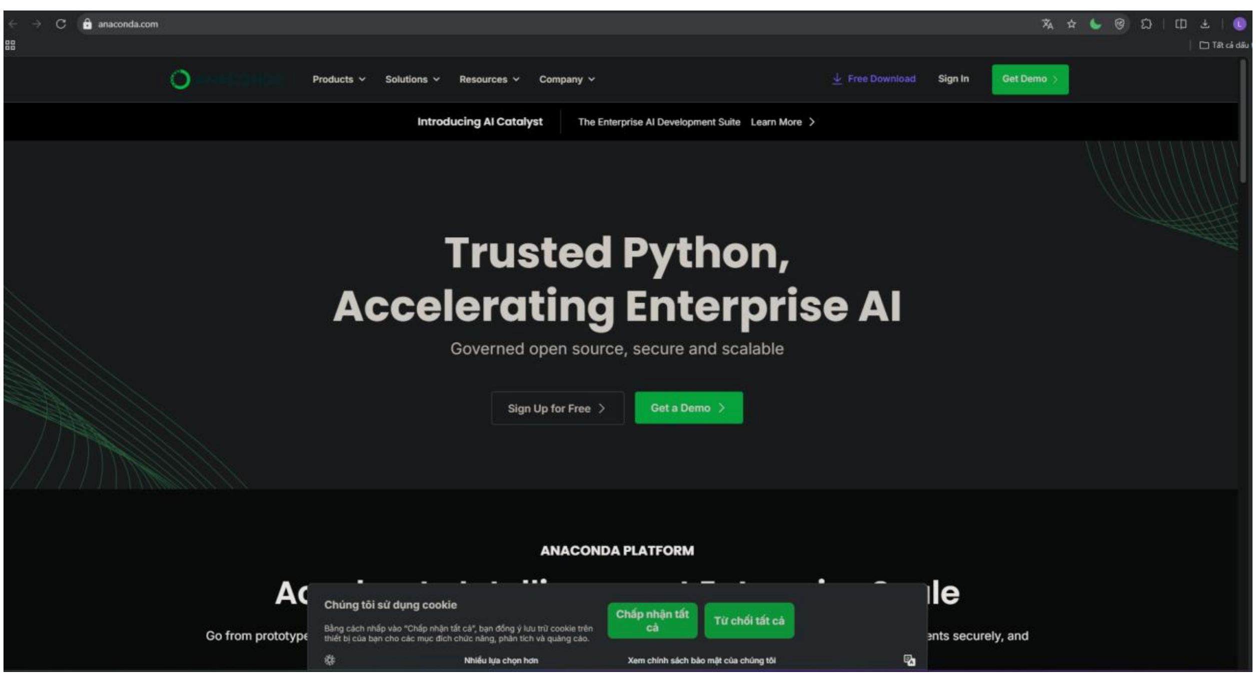
- Sau khi giải nén có 3 thư mục: "easy_ham": email hợp lệ, dễ phân biệt; "hard_ham": email hợp lệ nhưng có nội dung dễ nhầm lẫn với spam; "spam_2": email rác.



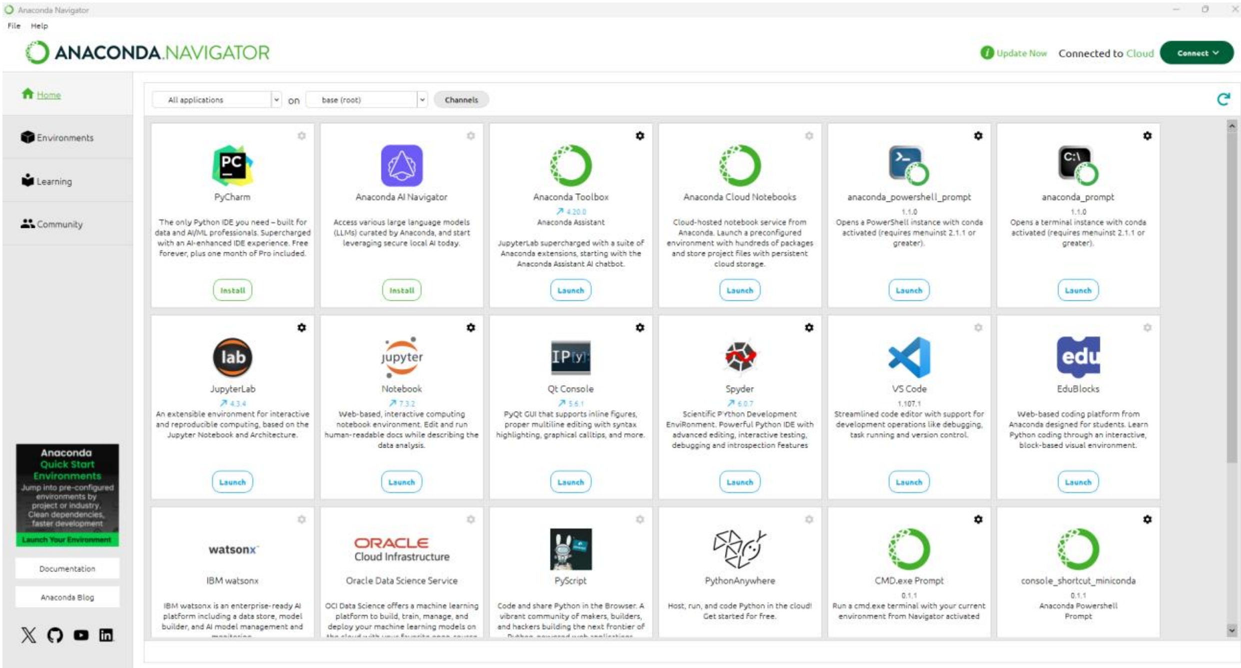
3.2.3. Cài đặt môi trường

Bước 1: Cài đặt Anaconda Navigator

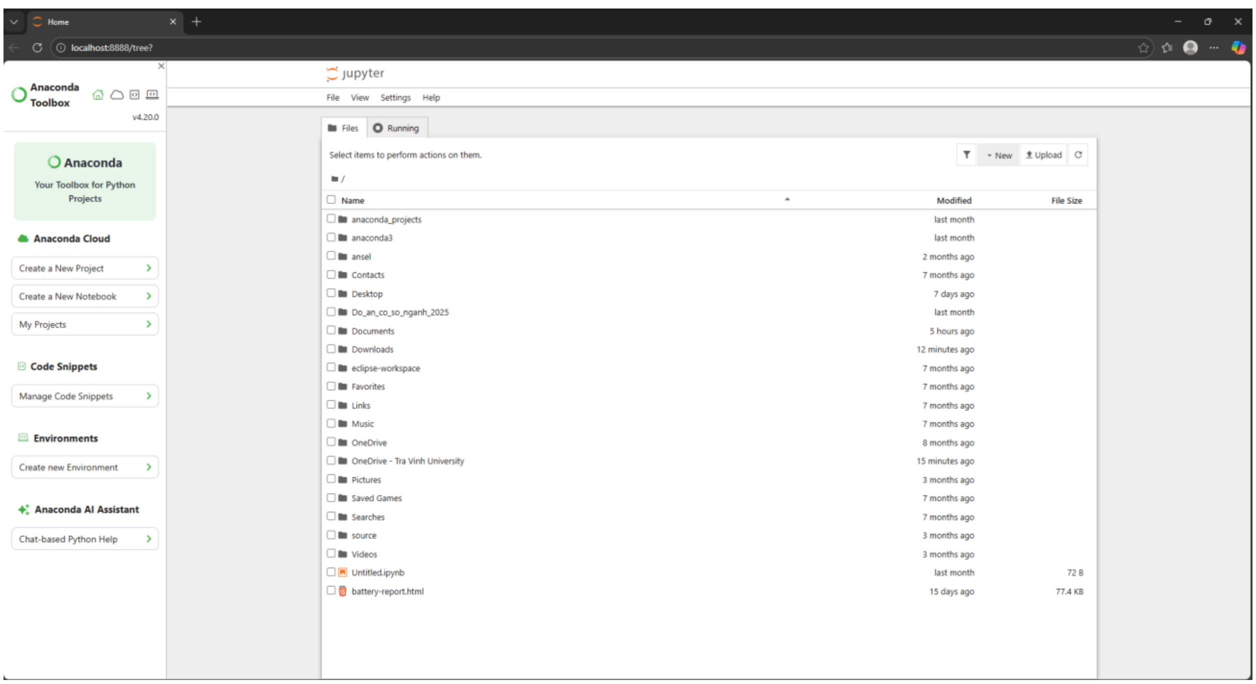
- Truy cập vào trình duyệt bất kì, vào thanh tìm kiếm gõ và tìm kiếm trang anacoda.com sau đó đăng nhập và tải xuống phiên bản Anaconda3 2025.06 (Python 3.13.5 64- bit) dành cho máy tính Window.



Bước 2: Khởi tạo Jupyter Notebook từ Anaconda Navigator



Giao diện Jupyter Notebook sau khi khởi động, hiển thị danh sách các thư mục và tệp tin trên máy tính



3.3. Xử dụng và tiền xử lý dữ liệu

3.3.1. Khai báo thư viện

- Để hiện thực hóa mô hình phân loại spam, không spam, tôi sử dụng ngôn ngữ Python với thư viện scikit-learn cho các bước tiền xử lý văn bản (TF-IDF), chia dữ liệu (train/test split), huấn luyện mô hình SVM (SVC) và đánh giá kết quả (classification_report). Ngoài ra, thư viện chuẩn os được dùng để thao tác với hệ thống file.

```
import os
from sklearn.feature_extraction.text import
TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report
```

3.3.2. Đọc dữ liệu từ thư mục

Để đọc dữ liệu email từ các thư mục (bao gồm cả thư mục con), đồ án xây dựng hàm **load_emails_from_folder**. Hàm này có nhiệm vụ duyệt qua toàn bộ thư mục, đọc nội dung từng email và lưu vào danh sách.

```
# Hàm đọc email từ thư mục (bao gồm cả thư mục con)
def load_emails_from_folder(folder):
    emails = []
    for root, dirs, files in os.walk(folder):   # duyệt cả
thư mục con
        for filename in files:
            filepath = os.path.join(root, filename)
            try:
                with open(filepath, 'r', encoding='latin-
1', errors='ignore') as f:
                    emails.append(f.read())
            except Exception as e:
```

```
            print("Không đọc được file:", filepath, e)

    return emails
```

3.3.3. Tải dữ liệu từ các thư mục con

Trong bước này, chương trình tiến hành đọc dữ liệu email từ bộ SpamAssassin đã được giải nén và lưu trữ trong các thư mục con **easy_ham**, **hard_ham** và **spam_2**. Hàm **load_emails_from_folder** được sử dụng để duyệt toàn bộ cây thư mục, bao gồm cả các thư mục con, và thu thập nội dung của từng file email dưới dạng văn bản. Kết quả thu được là ba tập dữ liệu riêng biệt, tương ứng với các nhóm email ham dễ, ham khó và spam.

```
# Đường dẫn gốc tới SpamAssassin
base_path = r"C:\Users\as\OneDrive - Tra Vinh University\Học_Tập\Do_an_co_so_nganh_2025\SpamAssassin"


# Đọc dữ liệu từ các thư mục con
easy_ham = load_emails_from_folder(os.path.join(base_path, "easy_ham"))
hard_ham = load_emails_from_folder(os.path.join(base_path, "hard_ham"))
spam = load_emails_from_folder(os.path.join(base_path, "spam_2"))


print("Số email easy_ham:", len(easy_ham))
print("Số email hard_ham:", len(hard_ham))
print("Số email spam:", len(spam))
```

Kết quả sau khi thực thi chương trình:

```
[4]: # Đường dẫn gốc tới SpamAssassin
base_path = r"C:\Users\as\OneDrive - Tra Vinh University\Học_Tập\Do_an_co_so_nganh_2025\SpamAssassin"

# Đọc dữ liệu từ các thư mục con
easy_ham = load_emails_from_folder(os.path.join(base_path, "easy_ham"))
hard_ham = load_emails_from_folder(os.path.join(base_path, "hard_ham"))
spam = load_emails_from_folder(os.path.join(base_path, "spam_2"))

print("Số email easy_ham:", len(easy_ham))
print("Số email hard_ham:", len(hard_ham))
print("Số email spam:", len(spam))

Số email easy_ham: 5103
Số email hard_ham: 501
Số email spam: 2794
```

3.2.4. Gộp dữ liệu và gán nhãn

Sau khi đọc dữ liệu, các email được gộp lại và gán nhãn tương ứng.

```
# Gộp dữ liệu và gán nhãn
texts = easy_ham + hard_ham + spam
labels = ["ham"] * (len(easy_ham) + len(hard_ham)) +
["spam"] * len(spam)


print("Tổng số email:", len(texts))
```

Kết quả thực thi chương trình

```
[15]: # Gộp dữ liệu và gán nhãn
texts = easy_ham + hard_ham + spam
labels = ["ham"] * (len(easy_ham) + len(hard_ham)) + ["spam"] * len(spam)

print("Tổng số email:", len(texts))

Tổng số email: 8398
```

3.2.5. Vector hóa bằng TF - IDF

Do dữ liệu email là dữ liệu văn bản, cần chuyển đổi sang dạng vector số để mô hình SVM có thể xử lý. kỹ thuật **TF-IDF** được sử dụng để biểu diễn nội dung email.

Trong bước này, tập dữ liệu sau khi đã được biểu diễn bằng **TF-IDF** sẽ được chia thành hai phần: 80% dành cho huấn luyện mô hình và 20% dành cho kiểm thử. Việc chia dữ liệu được thực hiện bằng hàm **train_test_split** trong thư viện **scikit-learn**. Kết quả thu được gồm bốn tập: **X_train** và **y_train** dùng cho quá trình huấn luyện, **X_test** và **y_test** dùng để đánh giá mô hình. Tham số **random_state** được thiết lập nhằm đảm bảo quá trình chia dữ liệu có thể tái lập, giúp kết quả thực nghiệm ổn định và có thể so sánh được.

```
# Vector hóa bằng TF-IDF
vectorizer = TfidfVectorizer(max_features=5000)
X = vectorizer.fit_transform(texts)

# Chia dữ liệu train/test
X_train, X_test, y_train, y_test = train_test_split(
    X, labels, test_size=0.2, random_state=42
)
```

3.2.6. Huấn luyện mô hình SVM

Trong bước này, mô hình Support Vector Machine (SVM) được sử dụng để phân loại email thành hai nhóm spam và ham. Với dữ liệu văn bản có số chiều lớn, kernel tuyến tính **(kernel='linear')** được lựa chọn nhằm tối ưu hiệu quả xử lý. Quá trình huấn luyện được thực hiện bằng lệnh **model.fit(X_train, y_train),** trong đó X_train và y_train là tập dữ liệu huấn luyện đã được chuẩn bị từ bước trước. Kết quả của bước này là một mô hình SVM đã học được đặc trưng từ dữ liệu, sẵn sàng cho quá trình kiểm thử và đánh giá.

Sau khi hoàn tất quá trình huấn luyện, mô hình SVM được kiểm thử trên tập dữ liệu kiểm thử để đánh giá hiệu quả phân loại. Việc đánh giá được thực hiện bằng hàm classification_report trong thư viện scikit-learn, cung cấp các chỉ số quan trọng như **Precision** (độ chính xác của từng lớp), **Recall** (khả năng phát hiện đúng email spam) và **F1-score** (chỉ số cân bằng giữa Precision và Recall). Các thước đo này giúp phản ánh toàn diện hiệu năng của mô hình, đồng thời là cơ sở để so sánh và cải thiện chất lượng phân loại trong nghiên cứu.

```
# Huấn luyện mô hình SVM
model = SVC(kernel='linear')
model.fit(X_train, y_train)
```

```
# Đánh giá mô hình
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

Kết quả sau khi thực thi chương trình

```
[17]: # Huấn luyện mô hình SVM
model = SVC(kernel='linear')
model.fit(X_train, y_train)

# Đánh giá mô hình
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

         ham       1.00      0.99      1.00      1134
        spam       0.99      0.99      0.99       546

    accuracy                           0.99      1680
   macro avg       0.99      0.99      0.99      1680
weighted avg       0.99      0.99      0.99      1680
```

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

4.1 Kết quả phân loại email

-

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Về kiến thức

Thông qua việc nghiên cứu và thực hiện đề tài *"Tìm hiểu về Support Vector Machine (SVM) trong phân loại văn bản và xây dựng công cụ lọc email spam"*, tôi đã nắm được các kiến thức cơ bản về học máy có giám sát, đặc biệt là thuật toán Support Vector Machine.

Đề tài giúp làm rõ nguyên lý hoạt động của SVM trong bài toán phân loại nhị phân, bao gồm khái niệm siêu phẳng phân chia, khoảng cách biên (margin) và các điểm hỗ trợ (support vectors). Bên cạnh đó, tôi hiểu được vai trò của việc biểu diễn dữ liệu văn bản bằng các đặc trưng số thông qua phương pháp TF-IDF trong các bài toán xử lý ngôn ngữ tự nhiên.

Qua quá trình tìm hiểu lý thuyết và áp dụng vào bài toán cụ thể, sinh viên đã có cái nhìn tổng quan hơn về cách một thuật toán học máy được xây dựng, huấn luyện và đánh giá trong thực tế.

5.2. Về thực hành

Về mặt thực hành, đề tài đã giúp tôi đã rèn luyện kỹ năng làm việc với dữ liệu thực tế, từ khâu thu thập, tiền xử lý dữ liệu email cho đến việc xây dựng và huấn luyện mô hình SVM bằng ngôn ngữ Python.

Tôi đã làm quen với môi trường Jupyter Notebook thông qua Anaconda, sử dụng các thư viện phổ biến như Pandas và Scikit-learn để triển khai mô hình phân loại email spam. Kết quả thực nghiệm cho thấy mô hình SVM có khả năng phân loại email spam và không spam với độ chính xác tương đối tốt, đáp ứng mục tiêu đề ra của đề tài.

Ngoài ra, quá trình thực hiện đề tài còn giúp sinh viên nâng cao khả năng đọc hiểu tài liệu kỹ thuật, phân tích kết quả và trình bày nội dung nghiên cứu một cách có hệ thống.