TRƯỜNG ĐẠI HỌC HỌC VĂN LANG

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN MÔN HỌC HK242

SỐ HOÁ VÀ QUẢN TRỊ THÔNG TIN SỐ (71ITDS40403)

QUẢN LÝ DỮ LIỆU VÀ BẢO MẬT THÔNG TIN

Nhóm sinh viên thực hiện (Họ tên - Mã SV):

1.Trần Tấn Phát_2274802010644 (Trưởng nhóm)

2. Đăng Võ Quang Huy_2274802010301

3. Huỳnh Gia Huy_2274802010303

4. Lê Minh Tâm_2274802010781

5. Nguyễn Thái Nguyên_2274802010587

TP. Hồ Chí Minh – năm 2025

LỜI CẢM ƠN

Lời nói đầu tiên, chúng em xin chân thành cảm ơn sự hướng dẫn của thầy Hoàng Lê Minh và thầy Nguyễn Thái Anh, người đã luôn dành thời gian và tâm huyết để hỗ trợ chúng em trong suốt quá trình học tập và nghiên cứu. Trong quá trình thực hiện nghiên cứu đề tài, chúng em đã gặp không ít khó khăn nhưng nhờ có sự hướng dẫn tận tình của thầy nên nhóm em đã có thể hoàn thành tốt bài tiểu luận. Tiếp theo, chúng em xin gửi lời chân thành cảm ơn đến Khoa Công nghệ thông tin- Đại Học Văn Lang thành phố Hồ Chí Minh đã tạo điều kiện thuận lợi cho chúng em học tập và hoàn thành đề tài tiểu luận này. Mặc dù nhóm đã rất cố gắng vận dụng những kiến thức đã học được trong thời gian qua để hoàn thành bài tiểu luận nhưng do không có nhiều kinh nghiệm thực tiễn nên khó tránh khỏi những thiếu sót trong quá trình nghiên cứu và làm bài. Nhưng với tinh thần cầu tiến và mong muốn tiến bộ, chúng em tin rằng những ý kiến và đóng góp của quý thầy cô và các bạn đọc giả sẽ góp phần giúp chúng em hoàn thiện bản thân hơn.

Table of Contents

LỜI MỞ ĐẦU

A. Lý do chọn đề tài

- Chủ đề "Quản lý dữ liệu và bảo mật thông tin" được lựa chọn dựa trên tầm quan trọng ngày càng gia tăng của việc số hóa thông tin trong mọi lĩnh vực của đời sống hiện đại, đặc biệt là trong bối cảnh cuộc cách mạng công nghiệp 4.0 đang diễn ra mạnh mẽ. Trong ngành công nghệ thông tin, quản lý dữ liệu không chỉ là một yêu cầu kỹ thuật mà còn là một yếu tố sống còn đối với sự phát triển bền vững của các tổ chức, doanh nghiệp và cả hệ thống xã hội.
- Dữ liệu ngày nay không chỉ đơn thuần là các con số hay ký tự, mà đã trở thành tài sản quý giá, được ví như "dầu mỏ" của thời đại số. Tuy nhiên, cùng với sự gia tăng về khối lượng và giá trị của dữ liệu là những thách thức lớn liên quan đến việc bảo vệ chúng trước các nguy cơ như tấn công mạng, rò rỉ thông tin, hoặc sử dụng sai mục đích.
- Ở góc độ thực tế, nhu cầu đảm bảo an toàn thông tin đang trở nên cấp thiết hơn bao giờ hết khi các vụ vi phạm dữ liệu liên tục được ghi nhận trên toàn cầu. Từ các doanh nghiệp nhỏ đến các tập đoàn lớn, từ cơ quan nhà nước đến cá nhân, tất cả đều cần một hệ thống quản lý dữ liệu hiệu quả và các biện pháp bảo mật thông tin tối ưu.

- Việc lựa chọn chủ đề này không chỉ xuất phát từ xu hướng phát triển của ngành công nghệ mà còn từ mong muốn cá nhân trong việc tìm hiểu sâu hơn về cách tổ chức, lưu trữ và bảo vệ dữ liệu một cách khoa học, từ đó đóng góp vào việc giải quyết các vấn đề thực tiễn trong lĩnh vực số hóa và quản trị thông tin số.

B. Phạm vi tìm hiểu

- Đối tượng nghiên cứu của chủ đề này tập trung vào các hệ thống quản lý dữ liệu và các phương pháp bảo mật thông tin trong bối cảnh số hóa. Phạm vi tìm hiểu bao gồm cả lịch sử phát triển của lĩnh vực này và quá trình ứng dụng thực tế qua các thời kỳ. Vào những năm đầu của kỷ nguyên số, dữ liệu chủ yếu được lưu trữ dưới dạng vật lý như giấy tờ, hồ sơ, sau đó chuyển sang các hệ thống máy tính đơn giản với các cơ sở dữ liệu cơ bản.
- Đến nay, với sự ra đời của điện toán đám mây, trí tuệ nhân tạo và phân tích dữ liệu lớn (Big Data), cách thức quản lý và bảo mật thông tin đã thay đổi hoàn toàn, đòi hỏi các công cụ và chiến lược mới để thích nghi.
- Trong phạm vi nghiên cứu, nhóm sẽ xem xét các khía cạnh từ lý thuyết đến thực hành, từ các mô hình quản lý dữ liệu truyền thống như hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) đến các giải pháp hiện đại như blockchain hay hệ thống mã hóa tiên tiến.
- Đồng thời, quá trình phát triển thực tế của chủ đề cũng sẽ được phân tích qua các ví dụ cụ thể, chẳng hạn như cách các tổ chức lớn như Google, Amazon hay các cơ quan chính phủ đã triển khai hệ thống quản lý dữ liệu và bảo mật thông tin để đối phó với những thách thức trong thời đại số.

C. Phương pháp thực hiện

- Để thực hiện chủ đề này, nhóm
- sẽ áp dụng một quy trình nghiên cứu có hệ thống bao gồm nhiều bước khác nhau. Trước tiên, việc thu thập dữ liệu sẽ được tiến hành thông qua việc tham khảo các tài liệu học thuật, sách chuyên ngành, bài báo khoa học và các nguồn thông tin đáng tin cậy trên Internet.
- Ngoài ra, nhóm cũng sẽ tìm hiểu các báo cáo thực tế từ các doanh nghiệp hoặc tổ chức đã triển khai hệ thống quản lý dữ liệu và bảo mật thông tin để có cái nhìn toàn diện hơn.
- Sau khi thu thập dữ liệu, nhóm sẽ tiến hành thống kê và phân tích chúng bằng cách sử dụng các công cụ như bảng biểu, sơ đồ luồng dữ liệu (DFD) và các phần mềm hỗ trợ phân tích dữ liệu. Việc so sánh các mô hình quản lý dữ liệu khác nhau (ví dụ: mô hình tập trung so với mô hình phân tán) cũng sẽ được thực hiện để đánh giá ưu, nhược điểm của từng phương pháp.
- Cuối cùng, dựa trên các kết quả phân tích, nhóm sẽ đề xuất một số giải pháp hoặc cải tiến phù hợp với nhu cầu thực tế, đồng thời kiểm chứng tính khả thi thông qua việc xây dựng một mô hình thử nghiệm đơn giản.

CHƯƠNG 1: GIỚI THIỆU KHÁI QUÁT

1. LÝ THUYẾT CƠ BẢN

- Số hóa và quản trị thông tin số là hai khái niệm nền tảng trong thời đại công nghệ hiện nay, đóng vai trò quan trọng trong việc tối ưu hóa cách thông tin được xử lý và sử dụng. Số hóa là quá trình chuyển đổi thông tin từ dạng vật lý (như thư tay, giấy tờ) sang dạng số (dữ liệu điện tử), tạo điều kiện thuận lợi cho việc lưu trữ, xử lý và chia sẻ thông tin một cách nhanh chóng.
- Trong khi đó, quản trị thông tin số bao gồm các hoạt động tổ chức, kiểm soát và bảo vệ dữ liệu để đảm bảo hiệu quả sử dụng và an toàn tối đa. Hai khái niệm này gắn bó mật thiết với chủ đề "Quản lý dữ liệu và bảo mật thông tin thông qua việc phân loại email người dùng", bởi lẽ quản lý dữ liệu là bước cơ bản để sắp xếp và phân loại email một cách khoa học, còn bảo mật thông tin đảm bảo rằng các email chứa dữ liệu nhạy cảm được bảo vệ khỏi các nguy cơ trong môi trường số.
- Về mặt lý thuyết, quản lý dữ liệu trong bối cảnh này liên quan đến việc thu thập email, phân loại chúng dựa trên nội dung (ví dụ: email cá nhân, công việc, quảng cáo), lưu trữ và xử lý dữ liệu để phục vụ các mục đích khác nhau. Các công cụ phổ biến hỗ trợ quá trình này bao gồm hệ quản trị cơ sở dữ liệu (DBMS), các thuật toán học máy để phân loại tự động, và hệ thống lưu trữ đám mây để đảm bảo khả năng truy cập linh hoạt.
- Đồng thời, bảo mật thông tin tập trung vào việc bảo vệ nội dung email trước các mối đe dọa như truy cập trái phép, phishing (lừa đảo qua email), hoặc rò rỉ dữ liệu. Các phương pháp bảo mật thường được áp dụng bao gồm mã hóa (encryption) để bảo vệ nội dung email, xác thực đa yếu tố (MFA) để kiểm soát quyền truy cập, và tường lửa (firewall) để ngăn chặn các cuộc tấn công mạng.

2. PHƯƠNG PHÁP, QUY TRÌNH THỰC HIỆN

- Để hiện thực hóa chủ đề "Quản lý dữ liệu và bảo mật thông tin thông qua việc phân loại email người dùng",bằng cách áp dụng một quy trình gồm ba giai đoạn chính:

    - Nghiên cứu và thiết kế hệ thống.
    - Triển khai thử nghiệm.
    - Đánh giá và cải tiến.

- Ở giai đoạn nghiên cứu, nhóm sẽ tập trung vào việc phân tích các yêu cầu cơ bản của một hệ thống phân loại email, bao gồm khả năng nhận diện nội dung email, phân loại chính xác theo các danh mục, và đảm bảo bảo mật cho dữ liệu nhạy cảm. Sau đó, nhóm sẽ thiết kế một mô hình hệ thống đơn giản dựa trên các công cụ như MySQL (để lưu trữ email), Python với thư viện học máy như scikit-learn (để phân loại email), và thuật toán mã hóa AES (để bảo mật thông tin).

- Phạm vi thực hiện sẽ được giới hạn trong việc xử lý một tập hợp email mẫu của một nhóm người dùng nhỏ, chẳng hạn như email của một cá nhân hoặc một nhóm nhân viên trong công ty, thay vì toàn bộ hệ thống email của một tổ chức lớn. Điều này giúp đảm bảo tính khả thi trong thời gian thực hiện đồ án. Quy trình triển khai bao gồm các bước: thu thập email mẫu, áp dụng thuật toán phân loại để sắp xếp chúng vào các danh mục (ví dụ: công việc, cá nhân, spam), mã hóa các email nhạy cảm, lưu trữ vào cơ sở dữ liệu, và kiểm tra hiệu quả hoạt động của hệ thống thông qua việc truy xuất và giải mã email.

3. NỘI DUNG LIÊN QUAN NHÓM TỪNG THỰC HIỆN

- Trong đồ án này, nhóm chúng tôi gồm 5 thành viên, mỗi người đảm nhận một vai trò cụ thể để hoàn thành các hạng mục chính của dự án. Các nội dung bao gồm việc xây dựng cơ sở dữ liệu mẫu để lưu trữ email (chứa các trường như người gửi, tiêu đề, nội dung), áp dụng thuật toán học máy để phân loại email dựa trên nội dung (ví dụ: email công việc, email quảng cáo), mã hóa các email nhạy cảm (như email chứa thông tin tài chính hoặc cá nhân), và thiết lập quy trình truy cập dữ liệu an toàn bằng mật khẩu để đảm bảo chỉ người dùng được ủy quyền mới có thể xem nội dung. Dưới đây là chức năng của từng thành viên trong nhóm:

    - **Trần Tấn Phát- Trưởng nhóm:** Chịu trách nhiệm nghiên cứu lý thuyết về quản lý dữ liệu và bảo mật thông tin trong lĩnh vực phân loại email, đồng thời điều phối công việc giữa các thành viên. Tổng hợp tài liệu, xác định các khái niệm cốt lõi và định hướng chung cho dự án.
    - **Đặng Võ Quang Huy:** Đảm nhận việc thiết kế mô hình hệ thống phân loại và bảo mật email. Công việc cụ thể bao gồm xây dựng cấu trúc cơ sở dữ liệu bằng MySQL, xác định các trường dữ liệu (người gửi, tiêu đề, nội dung), và phối hợp tích hợp các chức năng phân loại và mã hóa.
    - **Nguyễn Thái Nguyên:** Phụ trách triển khai thử nghiệm trên máy tính, bao gồm việc lập trình bằng Python để thu thập email mẫu, áp dụng thư viện scikit-learn để phân loại email tự động theo các danh mục (công việc, cá nhân, spam), và kiểm tra hiệu quả của thuật toán phân loại.
    - **Huỳnh Gia Huy:** Tập trung vào bảo mật thông tin, bao gồm việc tích hợp thuật toán mã hóa AES để bảo vệ các email nhạy cảm, thiết lập quy trình xác thực bằng mật khẩu, và kiểm tra tính an toàn của hệ thống khi truy xuất dữ liệu.
    - **Lê Minh Tâm:** Tập trung đánh giá thành viên thông qua báo cáo hằng ngày và kiểm tra tiến trình dự án.

4. PHẦN BÁO CÁO ĐỒ ÁN MÔN HỌC PHÂN LOẠI EMAIL SPAM VÀ HAM - MỐI LIÊN HỆ VỚI SỐ HÓA VÀ QUẢN TRỊ THÔNG TIN SỐ

1. Giới thiệu

Trong bối cảnh công nghệ hiện đại, email trở thành công cụ giao tiếp phổ biến, nhưng cũng đi kèm với vấn đề về lượng thư không mong muốn, hay còn gọi là email spam. Phân loại

email spam và ham là một bài toán quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) và machine learning. Mục tiêu của đồ án này là phân loại email thành hai nhóm: spam và ham, đồng thời tìm hiểu mối liên hệ giữa quá trình này và các lĩnh vực trong số hóa và quản trị thông tin số.

2. Số hóa Dữ Liệu và Quá Trình Phân Loại Email

2.1. Số hóa Dữ Liệu Email

Email là một dạng dữ liệu văn bản phi cấu trúc (unstructured data), tức là không có một cấu trúc cụ thể mà máy tính có thể dễ dàng xử lý. Để phân loại email, cần chuyển đổi nội dung của email từ dạng văn bản thành dạng số hóa, giúp máy tính có thể hiểu và xử lý được. Quá trình này bao gồm việc sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) để trích xuất các đặc trưng quan trọng từ nội dung email.

2.2. Natural Language Processing (NLP) trong Phân Loại Email

NLP là công nghệ chủ yếu giúp phân tích nội dung văn bản của email. Một số bước cơ bản trong NLP khi phân loại email gồm:

_ **Tokenization**: Chia nhỏ văn bản thành các từ hoặc cụm từ.

_ **Stopwords Removal**: Loại bỏ các từ không mang nhiều ý nghĩa (như “the”, “is”).

_ **Stemming/Lemmatization**: Chuyển đổi các từ về dạng gốc để giảm độ phức tạp.

_ **Vectorization**: Chuyển đổi các từ thành các vector số để mô hình học máy có thể sử dụng.

3. Quản Trị Thông Tin Số và Phân Loại Email

3.1. Bảo Mật và Lọc Nội Dung

Một trong những mục tiêu chính của quản trị thông tin số là bảo vệ người dùng khỏi các nguy cơ từ thư rác, lừa đảo qua email (phishing), và các mối đe dọa bảo mật khác. Hệ thống phân loại email tự động giúp lọc bỏ những email spam không mong muốn, bảo vệ người dùng khỏi các tác động tiêu cực.

_**Email Spam**: Là các thư không mong muốn, thường chứa quảng cáo, hoặc cố gắng lừa đảo người nhận.

_**Email Ham**: Là thư chính thống, hữu ích và có liên quan đến công việc hoặc giao tiếp cá nhân.

3.2. Quản Lý và Tổ Chức Dữ Liệu

Quản lý và tổ chức dữ liệu email đóng vai trò quan trọng trong việc đảm bảo hiệu quả giao tiếp. Các hệ thống email hiện đại như Gmail, Outlook sử dụng các mô hình học máy (machine

learning) để phân loại email tự động, giúp người dùng dễ dàng quản lý hộp thư của mình, và tránh tình trạng bị làm phiền bởi thư rác.

3.3. Hỗ Trợ Ra Quyết Định

Thông qua phân loại email, các dữ liệu có thể được tổng hợp và phân tích để rút ra những thông tin hữu ích. Ví dụ, việc phân tích các email spam có thể giúp xác định các mối đe dọa bảo mật, trong khi phân tích các email ham giúp cải thiện hiệu quả giao tiếp và tổ chức công việc.

4. Ứng Dụng Thực Tế

4.1. Spam Detection (Phát Hiện Thư Rác)

Spam detection là một trong những ứng dụng quan trọng nhất của phân loại email. Các mô hình học máy như Naive Bayes, SVM, hoặc các mạng nơ-ron nhân tạo có thể được sử dụng để phân loại email dựa trên các đặc trưng văn bản, chẳng hạn như tần suất xuất hiện của từ, các cụm từ đặc trưng, hay các mẫu hành vi trong nội dung email.

4.2. Phân Tích Nội Dung Email

Ngoài việc phân loại spam và ham, việc phân tích nội dung email cũng rất quan trọng trong việc tìm kiếm thông tin quan trọng từ hàng ngàn email. Các công cụ phân tích email sử dụng NLP để trích xuất thông tin, phân tích tần suất giao tiếp, và theo dõi các xu hướng hoặc chủ đề nổi bật trong email.

4.3. An Ninh Mạng và Bảo Mật

Các hệ thống phân loại email đóng vai trò quan trọng trong việc bảo vệ dữ liệu cá nhân và doanh nghiệp khỏi các mối đe dọa như phishing, malware, và các cuộc tấn công qua email. Việc phát hiện và chặn các email độc hại giúp bảo vệ không chỉ hệ thống cá nhân mà còn cả cơ sở hạ tầng mạng của doanh nghiệp.

5.Ứng dung chức năng

5.1 Mô hình huấn luyện

```
import numpy as np  # Thư viện tính toán số học
import pandas as pd  # Thư viện xử lý dữ liệu dạng bảng
import pickle  # Thư viện để lưu và tải mô hình
import re  # Thư viện xử lý biểu thức chính quy cho tiền xử lý văn bản
from sklearn.model_selection import train_test_split  # Hàm chia dữ liệu thành tập huấn luyện và kiểm tra
from sklearn.feature_extraction.text import TfidfVectorizer  # Chuyển văn bản thành vector TF-IDF
from sklearn.linear_model import LogisticRegression  # Mô hình hồi quy logistic
from sklearn.metrics import accuracy_score  # Đánh giá độ chính xác của mô hình
import random  # Thư viện tạo số ngẫu nhiên
```

```
from nltk.corpus import wordnet  # Từ điển WordNet để tìm từ đồng nghĩa
import nltk  # Thư viện xử lý ngôn ngữ tự nhiên
nltk.download('wordnet')  # Tải dữ liệu WordNet
nltk.download('omw-1.4')  # Tải dữ liệu mở rộng cho WordNet

# Load dataset
print(" Đang tải dữ liệu...")  # Thông báo trạng thái
df = pd.read_csv('DoAn/mail_data.csv', encoding='utf-8')  # Đọc file CSV chứa dữ liệu email
df.fillna('', inplace=True)  # Thay các giá trị NaN bằng chuỗi rỗng
print(f" Số dòng dữ liệu: {len(df)}")  # In số lượng dòng dữ liệu

df['Category'] = df['Category'].map({'spam': 0, 'ham': 1})  # Chuyển nhãn 'spam' thành 0, 'ham' thành 1

# Hàm tiền xử lý văn bản
def preprocess_text(text):
    text = text.lower()  # Chuyển văn bản thành chữ thường
    text = re.sub(r'\d+', '', text)  # Xóa tất cả các số
    text = re.sub(r'[^\w\s@.]', '', text)  # Xóa ký tự đặc biệt, giữ lại chữ, khoảng trắng, @ và .
    text = text.strip()  # Xóa khoảng trắng thừa ở đầu và cuối
    return text  # Trả về văn bản đã xử lý

df['Message'] = df['Message'].apply(preprocess_text)  # Áp dụng tiền xử lý cho cột 'Message'

# Synonym Replacement
def synonym_replacement(text):  # Hàm thay thế từ bằng từ đồng nghĩa
    words = text.split()  # Tách văn bản thành danh sách các từ
    new_words = []  # Danh sách lưu các từ mới
    for word in words:  # Duyệt qua từng từ
        synonyms = wordnet.synsets(word)  # Lấy danh sách từ đồng nghĩa từ WordNet
        if synonyms:  # Nếu có từ đồng nghĩa
            new_word = synonyms[0].lemmas()[0].name()  # Lấy từ đồng nghĩa đầu tiên
            new_words.append(new_word)  # Thêm vào danh sách
        else:
            new_words.append(word)  # Nếu không có, giữ nguyên từ gốc
    return ' '.join(new_words)  # Ghép các từ thành chuỗi và trả về

# Data Augmentation for spam
df_spam = df[df['Category'] == 0]  # Lọc dữ liệu spam (nhãn 0)
df_spam_aug = df_spam.copy()  # Tạo bản sao của dữ liệu spam
df_spam_aug['Message'] = df_spam_aug['Message'].apply(synonym_replacement)  # Tăng cường dữ liệu bằng từ đồng nghĩa
df_spam = pd.concat([df_spam, df_spam_aug])  # Gộp dữ liệu spam gốc và tăng cường

# Under-Sampling for ham
count_spam = len(df_spam)  # Đếm số dòng dữ liệu spam
df_ham = df[df['Category'] == 1].sample(n=count_spam, random_state=42)  # Lấy mẫu ngẫu nhiên số dòng ham bằng số spam

df = pd.concat([df_spam, df_ham]).sample(frac=1, random_state=42)  # Gộp spam và ham, xáo trộn ngẫu nhiên

# Split data
X = df['Message']  # Lấy cột văn bản làm đặc trưng đầu vào
Y = df['Category']  # Lấy cột nhãn làm đầu ra
```

```
X_train, X_test, Y_train, Y_test = train_test_split(
    X, Y, test_size=0.2, random_state=42, stratify=Y)  # Chia dữ liệu: 80% huấn luyện, 20% kiểm tra, giữ tỷ lệ nhãn

vectorizer = TfidfVectorizer(min_df=1, lowercase=True, stop_words='english')  # Khởi tạo vectorizer TF-IDF
X_train_features = vectorizer.fit_transform(X_train)  # Chuyển tập huấn luyện thành vector TF-IDF
X_test_features = vectorizer.transform(X_test)  # Chuyển tập kiểm tra thành vector TF-IDF

Y_train = Y_train.astype(int)  # Chuyển nhãn huấn luyện thành kiểu int
Y_test = Y_test.astype(int)  # Chuyển nhãn kiểm tra thành kiểu int

print(" Đang huấn luyện mô hình…")  # Thông báo trạng thái
model = LogisticRegression(C=1.0, max_iter=500, solver='liblinear')  # Khởi tạo mô hình Logistic Regression
model.fit(X_train_features, Y_train)  # Huấn luyện mô hình với dữ liệu huấn luyện

train_accuracy = accuracy_score(Y_train, model.predict(X_train_features))  # Tính độ chính xác trên tập huấn luyện
test_accuracy = accuracy_score(Y_test, model.predict(X_test_features))  # Tính độ chính xác trên tập kiểm tra

print(f' Training Accuracy: {train_accuracy:.4f}')  # In độ chính xác tập huấn luyện
print(f' Test Accuracy: {test_accuracy:.4f}')  # In độ chính xác tập kiểm tra

with open('spam_classifier.pkl', 'wb') as f:  # Mở file để lưu mô hình
    pickle.dump((vectorizer, model), f)  # Lưu vectorizer và mô hình vào file .pkl

print(" Mô hình đã được lưu thành công: spam_classifier.pkl")  # Thông báo lưu thành công
```

Huấn luyện mô hình phân loại

5.2 Khởi chạy mô hình thực thi

```
import pickle  # Thư viện để tải mô hình và vectorizer từ file
import re  # Thư viện xử lý biểu thức chính quy cho tiền xử lý văn bản
from flask import Flask, request, jsonify, render_template  # Flask framework và các công cụ liên quan
from flask_cors import CORS  # Hỗ trợ Cross-Origin Resource Sharing

app = Flask(__name__)  # Khởi tạo ứng dụng Flask, __name__ xác định vị trí ứng dụng
CORS(app)  # Kích hoạt CORS để hỗ trợ yêu cầu từ các nguồn khác

# Hàm tiền xử lý văn bản (Giữ giống `train_model.py`)
def preprocess_text(text):
    text = text.lower()  # Chuyển văn bản thành chữ thường để đồng nhất
    text = re.sub(r'\d+', '', text)  # Xóa tất cả các số trong văn bản
    text = re.sub(r'[^\w\s@.]', '', text)  # Xóa ký tự đặc biệt, giữ lại chữ, khoảng trắng, @ và .
    text = text.strip()  # Xóa khoảng trắng thừa ở đầu và cuối
    return text  # Trả về văn bản đã xử lý

# Load mô hình và vectorizer **một lần duy nhất**
with open('spam_classifier.pkl', 'rb') as f:  # Mở file mô hình đã lưu
```

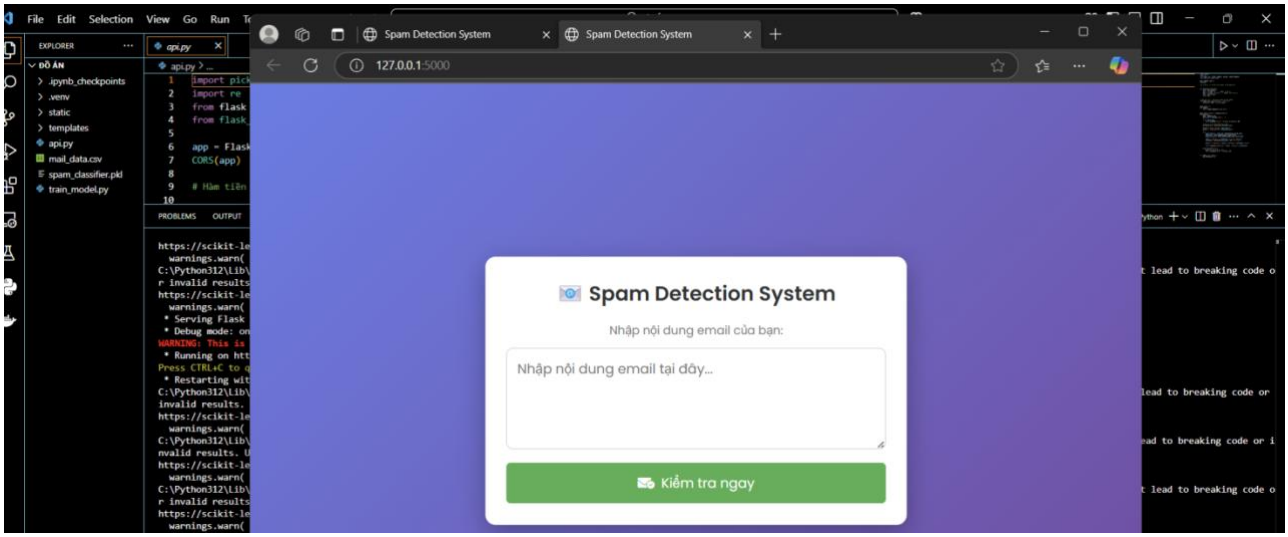
```
    vectorizer, model = pickle.load(f)  # Tải vectorizer và mô hình từ file .pkl

@app.route('/')  # Định nghĩa route chính (trang mặc định)
def index():
    return render_template('index.html')  # Trả về file HTML để hiển thị giao diện

@app.route('/predict', methods=['POST'])  # Định nghĩa route /predict, chỉ chấp nhận POST
def predict():
    data = request.json  # Lấy dữ liệu JSON từ yêu cầu POST
    email_text = data.get('email', '')  # Lấy giá trị 'email' từ JSON, mặc định là chuỗi rỗng nếu không có

    if not email_text:  # Kiểm tra nếu không có văn bản email
        return jsonify({'error': 'No email provided'}), 400  # Trả về lỗi JSON, mã 400 (Bad Request)

    # Tiền xử lý văn bản trước khi dự đoán
    processed_text = preprocess_text(email_text)  # Gọi hàm tiền xử lý để làm sạch văn bản

    print(f" Input received: {email_text}")  # In văn bản gốc để debug
    print(f" Processed text: {processed_text}")  # In văn bản đã xử lý để debug

    try:  # Bắt đầu khối xử lý lỗi
        input_features = vectorizer.transform([processed_text])  # Chuyển văn bản thành vector
        prediction = model.predict(input_features)[0]  # Dự đoán nhãn (0 = Spam, 1 = Ham)
        prediction_prob = model.predict_proba(input_features)[0]  # Lấy xác suất của các nhãn

        result = 'Ham mail' if prediction == 1 else 'Spam mail'  # Quyết định nhãn dựa trên dự đoán
        confidence = max(prediction_prob) * 100  # Tính độ tin cậy (phần trăm) từ xác suất cao nhất

        print(f" Prediction: {result} (Confidence: {confidence:.2f}%)")  # In kết quả dự đoán để debug

        return jsonify({'prediction': result, 'accuracy': confidence})  # Trả về kết quả JSON với nhãn và độ tin cậy

    except Exception as e:  # Xử lý lỗi nếu dự đoán thất bại
        print(f"Lỗi dự đoán: {e}")  # In lỗi để debug
        return jsonify({'error': str(e)}), 500  # Trả về lỗi JSON, mã 500 (Internal Server Error)

if __name__ == '__main__':  # Kiểm tra nếu file được chạy trực tiếp
    app.run(debug=True)  # Chạy ứng dụng Flask ở chế độ debug (hiển thị lỗi và tự động reload)
```

Kết quả thu được:

6. Kết quả thực hiện

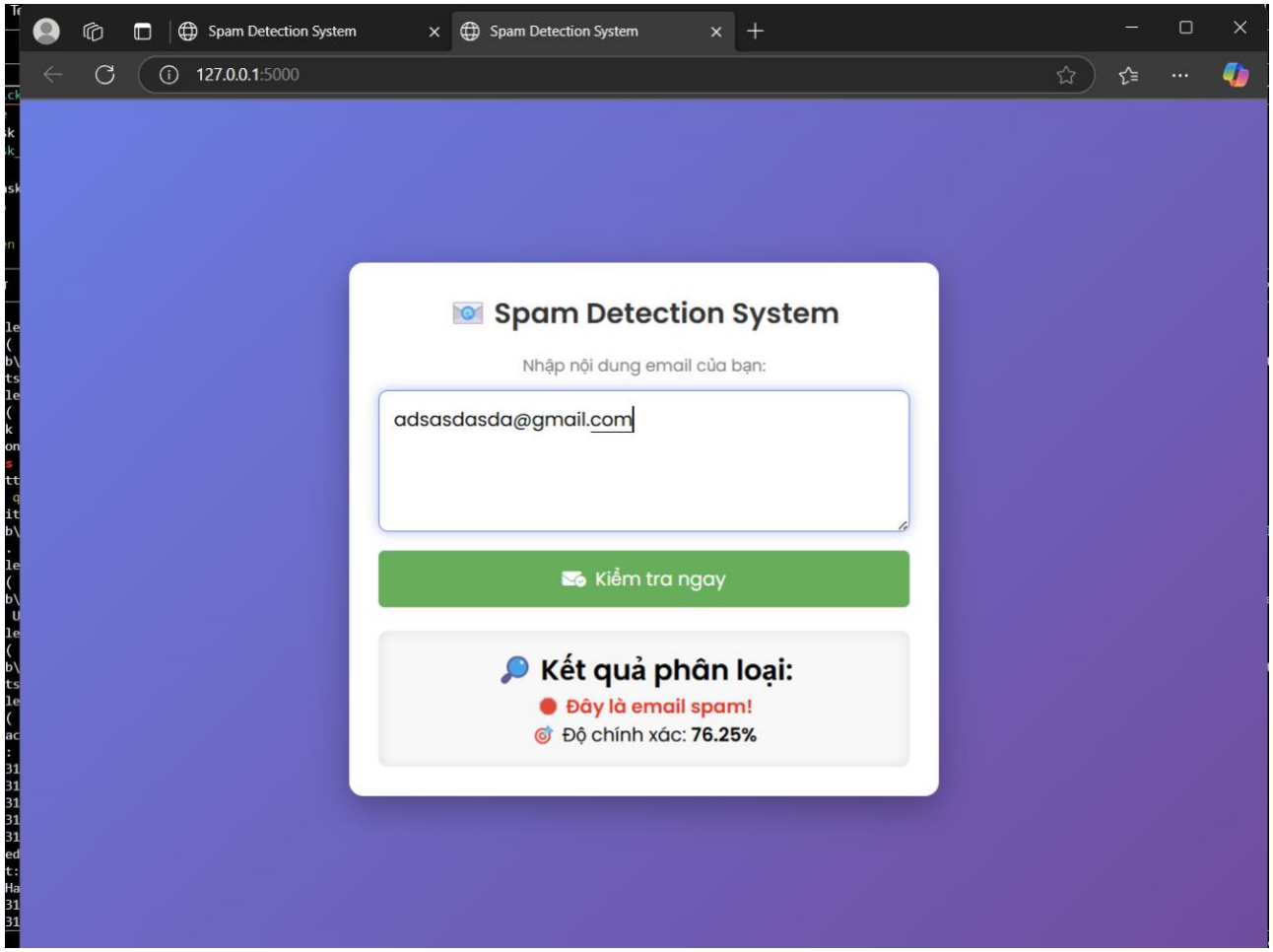
Giải thích:

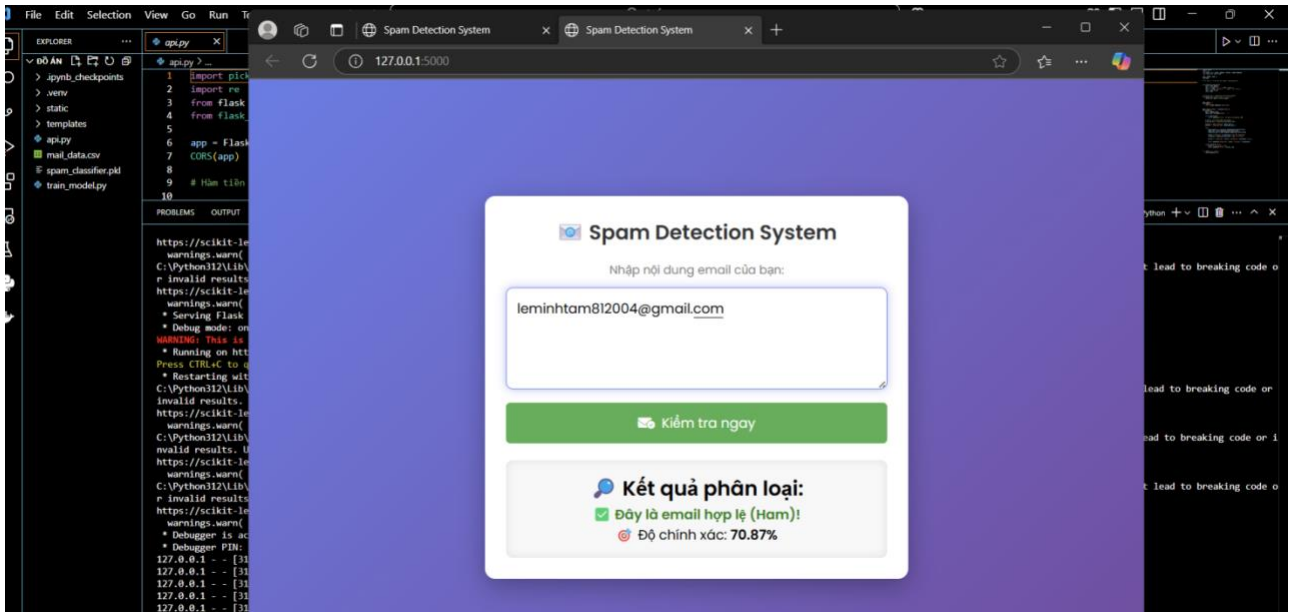
Đoạn code trên là một ứng dụng web đơn giản sử dụng Flask để triển khai mô hình học máy. Đầu tiên, nó import các thư viện cần thiết, bao gồm pickle để load mô hình đã được huấn luyện, Flask để tạo API web, và CORS để cho phép truy cập từ các nguồn khác. Sau đó, ứng dụng được khởi tạo với Flask(__name__) và kích hoạt CORS. Khi chạy, Flask server sẽ được khởi động ở chế độ debug, giúp phát hiện lỗi trong quá trình phát triển. Ngoài ra, log hiển thị các request gửi đến API, chẳng hạn như truy cập trang chính (GET /) hoặc gửi dữ liệu dự đoán (POST /predict). Một số cảnh báo liên quan đến phiên bản sklearn xuất hiện, nhưng chúng không ảnh hưởng nếu mô hình vẫn hoạt động bình thường.



Kết quả phân loại Gmail có phải gmail rác hay không.

Hệ thống **Spam Detection System** hiển thị trong hình ảnh là một ứng dụng web giúp người dùng kiểm tra xem email có phải là spam hay không. Giao diện đơn giản và trực quan với nền gradient, ô nhập nội dung email và nút **"Kiểm tra ngay"** để gửi dữ liệu đến máy chủ. Khi người dùng nhập email và nhấn nút kiểm tra, nội dung sẽ được gửi đến backend Flask, nơi mô hình học máy đã được huấn luyện sẽ phân tích và đưa ra dự đoán. Nếu hệ thống hoạt động đúng, nó sẽ phản hồi kết quả và hiển thị trên giao diện. Log của Flask trong ảnh cho thấy server đang chạy, tiếp nhận yêu cầu và xử lý dữ liệu.





Kết quả sau khi phân loại

6. KẾT LUẬN

Phân loại email spam là một ví dụ điển hình về ứng dụng của số hóa và quản trị thông tin số trong môi trường hiện đại. Quá trình phân loại này không chỉ giúp cải thiện bảo mật mà còn giúp nâng cao hiệu quả quản lý thông tin, tối ưu hóa giao tiếp và hỗ trợ quyết định trong tổ chức. Các công nghệ như NLP và machine learning đóng vai trò quan trọng trong việc tạo ra các hệ thống email thông minh, tự động phân loại và bảo vệ người dùng khỏi các mối nguy hiểm tiềm ẩn.

Bảng Đánh Giá Sinh Viên

STT	Tên sinh viên - Mã số sinh viên	Công việc	Phần trăm hoàn thành
1	Trần Tấn Phát - 2274802010644	Nghiêm cứu mô hình về Gmail	100%
2	Đặng Võ Quang Huy- 2274802010301	Thiết kế mô hình hệ thống	100%
3	Nguyễn Thái Nguyên- 2274802010587	Triển khai thử nghiệm sửa lỗi	100%
4	Huỳnh Gia Huy - 2274802010303	Kiểm tra độ bảo mật	100%
5	Lê Minh Tâm- 2274802010781	Đánh giá và kiểm tra dự án	100%